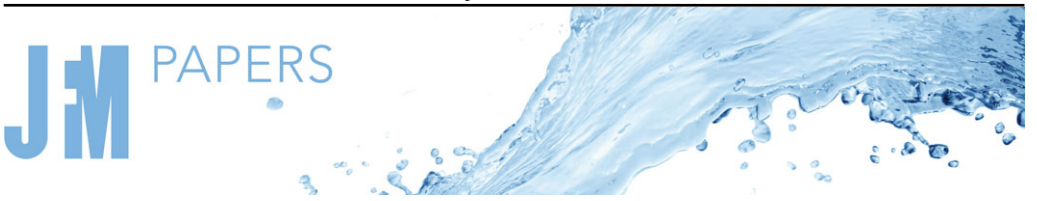
J. Fluid Mech. (2025), *vol.* 1008, A21, doi:10.1017/jfm.2025.91



Machine-learning-based pressure reconstruction with moving boundaries

Hongping Wang[1,2], **Fan Wu**[1,2], **Yi Liu**[1,2], **Xinyi He**[1,2], **Shuyi Feng**[3,4] and **Shizhao Wang**[1,2]

[1]State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, PR China

[2]School of Engineering Science, University of Chinese Academy of Sciences, Beijing 100049, PR China

[3]Department of Structural Heart Disease, National Center for Cardiovascular Disease, PR China

[4]Fuwai Hospital, Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing, PR China

Corresponding author: Shizhao Wang, wangsz@lnm.imech.ac.cn

(Received 21 June 2024; revised 29 December 2024; accepted 13 January 2025)

The greatest challenge in pressure reconstruction from the measured velocity fields is that the error of material acceleration is significantly contaminated due to error propagation. Particularly for flows with moving boundaries, accurate boundary velocities are difficult to obtain due to error propagation, and a complex boundary processing technique is needed to treat the moving boundaries. The present work proposes a machine-learning-based method to determine the pressure for incompressible flows with moving boundaries. The proposed network consists of two neural networks: one network, named the boundary network, is used to track the Lagrangian boundary points; the other physics-informed neural network, named the flow network, is adopted to approximate the flow fields. These two networks are coupled by imposing boundary conditions. We further propose a new dynamic weight strategy for the loss terms to guarantee convergence and stability. The performance of the proposed method is validated by two examples: the flow over an oscillating cylinder and the flow around a swimming fish. The proposed method can accurately determine the pressure fields and boundary motion from synthetic particle image velocimetry (PIV) flow fields. Moreover, this method can also predict the boundary and pressure at a given instant without supervised data. Finally, this method was applied to reconstruct the pressure from the two-dimensional and three-dimensional PIV velocities of the left ventricle. All of the results indicate that the proposed method can accurately reconstruct the pressure fields for flows with moving boundaries and is a novel method for surface pressure estimation.

Key words: machine learning, general fluid mechanics, flow–structure interactions

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

1. Introduction

Laser- and imaging-based techniques, including particle image velocimetry (PIV) and particle tracking velocimetry (PTV), have been widely used as reliable tools to visualise, measure and study the intrinsic characteristics of complex flows (Adrian 2005; Westerweel *et al.* 2013; Raffel *et al.* 2018). The velocity fields of unsteady flows can be accurately measured using planar techniques (Keane & Adrian 1992; Scarano 2002) or three-dimensional (3-D) measurement methods (Elsinga *et al.* 2006; Gao *et al.* 2013; Scarano 2013) with high temporal and spatial resolution. However, the measurement of pressure fields is challenging because common tracer particles only follow the motion of flows and are insensitive to dynamic variations in the pressure (Abe *et al.* 2004). The fluctuating pressure plays a prominent role in the evolution of flow structures and flow-induced noise (Lorenzoni *et al.* 2012; Zhou *et al.* 2021*b*). The surface pressure enables the estimation of object loads, which are highly important for engineering applications. Moreover, the intraventricular pressure gradient field has been demonstrated to be directly correlated with the function of the left ventricle (LV), and may represent a natural early indicator and predictor of cardiac function (Pedrizzetti & Domenichini 2015). Therefore, it is of great interest to determine pressure from velocity measurements by using the incompressible Navier–Stokes (N–S) equations (Liu & Katz 2006; van Oudheusden 2013).

However, pressure determination in scenarios with a moving boundary remains challenging. Generally, there are two methods for deriving pressure from velocity (van Oudheusden 2013; Cai *et al.* 2020; Fan *et al.* 2023). The first method involves integrating the pressure gradient along a specified straight path as proposed by Liu & Katz (2006), and the second method involves solving the pressure Poisson equation which is obtained by taking the divergence of the pressure gradient (de Kat & van Oudheusden 2012; van Oudheusden 2013; Pirnia *et al.* 2020). Both methods require accurate estimation of the material acceleration. Based on the N–S equations, the pressure gradient can be expressed as the sum of the material acceleration $-D\mathbf{U}/Dt$ and the viscous term $\nu\nabla^2\mathbf{U}$, where $\mathbf{U}$ denotes the velocity and ν is the kinematic viscosity. Much effort has been devoted to improving the accuracy of material acceleration in the Eulerian or Lagrangian framework. For the Eulerian framework, material acceleration involves both local and convective acceleration, which can be directly computed according to the structured velocity using finite difference or polynomial regression approaches (Kat & Ganapathisubramani 2012; van Oudheusden 2013; Wang *et al.* 2017*a*; Cai *et al.* 2020). For the Lagrangian framework, the acceleration is evaluated from the trajectory of a physical particle from PTV (Ferrari & Rossi 2008; Schanz *et al.* 2016; Shao *et al.* 2020; Schröder & Schanz 2023) or an imaginary particle from the Eulerian velocity fields (Liu & Katz 2006; de Kat & van Oudheusden 2012; Wang *et al.* 2017*b*). van Gent *et al.* (2017) assessed the performance of pressure field reconstructions from PIV and PTV measurements using zonal detached eddy simulations. The results indicated that a more accurate pressure field could be reconstructed via Lagrangian approaches in combination with other advanced techniques. The most widely used path integration-based and Poisson equation-based methods are feasible for the full fluid domain. However, when a stationary or moving object is immersed in the fluid domain, the pressure estimation around the immersed object becomes more complicated due to the fluid–solid interaction (Pirnia *et al.* 2020). The pressure determination depends strongly on accurate Neumann boundary conditions, which are difficult to obtain experimentally because of the high velocity gradient, coarse spatial resolution and laser light reflection near the surfaces of the immersed objects. Few studies have investigated the pressure under conditions with moving boundaries; the work reported by Pirnia *et al.* (2020) is an exception, where a finite element approach was used to estimate pressure fields from the planar velocity data around immersed bodies.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

In their work, by reducing the order of the velocity derivatives and incorporating the surface dynamics in the Poisson equation, the error of the pressure estimation was decreased to a low level compared with that obtained by integrating the pressure gradient and solving the pressure Poisson equation.

Machine-learning-based approaches have shown promise for improving the spatial/temporal resolution of turbulent flows (Fukami *et al.* 2019,2021; Kim *et al.* 2021; Wang *et al.* 2020*b*), developing turbulence models for computational fluid mechanics (CFD) (Ling *et al.* 2016; Duraisamy *et al.* 2019; Zhou *et al.* 2021*a*; Zhang *et al.* 2022), discovering governing equations and identifying unique dimensionless quantities (Brunton *et al.* 2016; Rabault *et al.* 2019; Zhao *et al.* 2023; Xu *et al.* 2022; Churchill *et al.* 2023), reducing the order of complex systems (Yu *et al.* 2019; Lui & Wolf 2019; Murata *et al.* 2020; Callaham *et al.* 2023; Liao *et al.* 2023), and enhancing experimental data (Cai *et al.* 2019; Shao *et al.* 2020; Gao *et al.* 2021; Cai *et al.* 2021*b*; Wang *et al.* 2022*a*). A comprehensive review of machine learning for fluid mechanics was given by Brunton *et al.* (2020). Recently, the physics-informed neural network (PINN) proposed by Raissi *et al.* (2020) has received much attention because of its excellent performance in solving inverse problems involving nonlinear partial differential equations (Raissi *et al.* 2019*a*). By embedding physical laws into the loss function of a data-driven neural network, the PINN can obtain optimal variables under constraints with known data and laws. Raissi *et al.* (2020) proposed hidden fluid mechanics based on a PINN to learn velocity and pressure fields from images of flow visualisations. Cai *et al.* (2021*b*) used a PINN to estimate velocity and pressure fields from the temperature data of tomographic background oriented Schlieren experiments. These inverse problems are extremely ill-posed because only a scalar quantity is given. However, the results suggest that the PINN is a powerful optimisation tool for addressing these inverse problems. In addition, Xu *et al.* (2021) employed a PINN and treated the governing equations as parametrised constraints to recover missing flow field information. Moreover, Arzani *et al.* (2021) used a PINN to obtain the wall shear stress for near-wall haemodynamics from sparse velocity measurements. Wang *et al.* (2022*a*) applied a PINN to reconstruct dense velocity fields from sparse experimental data under the constraint of N–S equations. One of the outstanding advantages of a PINN is that it can automatically solve for pressure fields without any pressure information. In their experimental assessment of the 3-D wake flow of a hemisphere, the low-pressure region estimated by the PINN was found to be consistent with the vortex core. Furthermore, a quantitative assessment of the pressure reconstruction methods, including the spectral decomposition-based fast pressure integration algorithm (Wang *et al.* 2017*a*), irrotation correction on pressure gradient and orthogonal-path integration method (Wang *et al.* 2016*c*) and a PINN, was reported by Fan *et al.* (2023). The results indicate that PINN-based pressure reconstruction is superior to other methods even if the velocity fields contain substantial measurement errors. Jiang *et al.* (2024) also demonstrated that a PINN can predict the latent pressure field in density-stratified flows. Notably, a PINN has been used to reconstruct two-dimensional (2-D) pressure fields around a swimming fish from PIV measurements (Calicchia *et al.* 2023). Due to the motion of the swimming fish, the kinematics data of the zebrafish must be considered during pressure reconstruction. The results indicate that the PINN predictions are more accurate than those of other methods because more physical information is used. Generally, moving boundaries can be incorporated into the loss function of a PINN, as proposed by Zhu *et al.* (2024). However, the loss function in their work considers only rigid motion with translation and rotation and cannot address the motion of non-rigid bodies. Moving boundary conditions, especially the boundary velocity, are difficult to measure accurately because of measurement noise. It is also needed to optimise the boundary conditions based on physical laws. There are

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

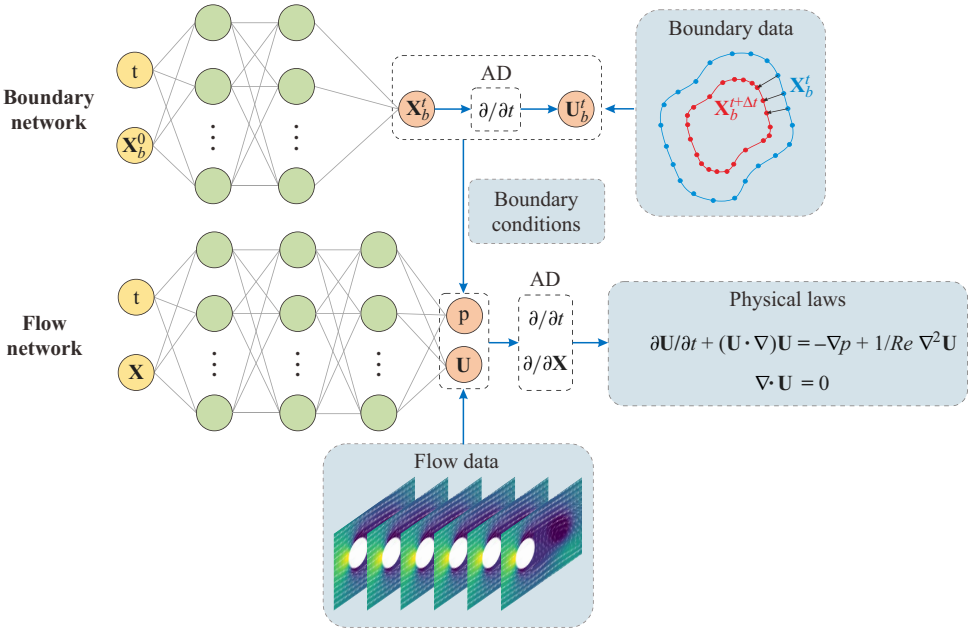


Figure 1. Architecture of the dual-network model for pressure reconstruction with moving boundaries. We name this network PINN-MB. The boundary network is used to track the motion of the boundaries, and the PINN-based flow network is used to approximate the flow fields. The parameters $\mathbf{X}$ and t denote space and time, respectively. $\mathbf{X}_b^t$ represents the Lagrangian points on the moving boundaries at time t.

still few examples for solving the pressure in the scenarios with moving boundaries due to the fluid–solid interactions.

In the present study, we propose a machine-learning-based method to reconstruct pressure fields from velocity fields with moving boundaries. This novel framework contains two neural networks: the PINN-based network is used to estimate the flow fields, and the other network is used to track the motion of moving objects. These two networks are coupled by boundary conditions. The rest of the paper is organised as follows. We first introduce our machine-learning-based pressure reconstruction approach in § 2. In § 3, the proposed method is comprehensively validated by considering the flows over an oscillating cylinder and around swimming fish, and then is applied to determine the pressure from experimental intraventricular flow data. Finally, concluding remarks are provided in § 4.

2. Methodology

2.1. *Dual-network model for pressure reconstruction with moving boundaries*

The PINN has been reviewed in many papers (Karniadakis *et al.* 2021; Cai *et al.* 2021*a*; Wang *et al.* 2022*a*,*b*); thus, in the present work, we do not provide a comprehensive introduction to the framework of the PINN. In brief, a PINN seamlessly integrates known data (measured data, boundary conditions and initial conditions) and physical laws by embedding partial differential equations into the loss function of a data-driven neural network via automatic differentiation. In this paper, we focus on the solution of the pressure fields with moving boundaries using a PINN.

In the present work, a dual-network model, named PINN-MB (MB is an abbreviation for moving boundary), is adopted to reconstruct the pressure with a moving boundary,

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

as shown in figure 1. PINN-MB consists of two neural networks: the boundary network and the flow network. The boundary network with input $(t, \mathbf{X}_b^0)$ and output $\mathbf{X}_b^t$ is used to capture the motion of the boundaries, where $\mathbf{X}_b^0$ denotes the initial boundary position and $\mathbf{X}_b^t$ represents the current boundary position at time t. The relationship between the inputs and outputs can be expressed as

$$\mathbf{X}_b^t = \mathscr{F}_{b.c.}\left(t, \mathbf{X}_b^0; \boldsymbol{\Theta}_{b.c.}\right). \tag{2.1}$$

Here, $\mathscr{F}_{b.c.}$ and $\boldsymbol{\Theta}_{b.c.}$ represent the boundary neural network and the corresponding trainable variables, respectively. Note that the boundary points should be preprocessed to ensure that the Lagrangian points at different times are one-to-one matches, as shown in the representation of the boundary data in figure 1. After ensuring the point correspondence, the boundaries at different times are described by a series of points with the same number and connectivity. The boundary network needs only the supervised boundary points for training, and the boundary velocity $\mathbf{U}_b^t$ is estimated from $\partial \mathbf{X}_b^t / \partial \mathbf{t}$ via automatic differentiation. Therefore, the boundary condition at time t is given as $(\mathbf{X}_b^t, \mathbf{U}_b^t)$. The flow network used to approximate the flow fields is expressed as follows:

$$(\mathbf{U}, p) = \mathscr{F}_{flow}\left(t, \mathbf{X}; \boldsymbol{\Theta}_{flow}\right), \tag{2.2}$$

where the inputs are the spatial coordinates $\mathbf{X}$ and time t, and the outputs are the velocity field $\mathbf{U}$ and pressure p. Similar to the PINN, the N–S equations are encoded into the loss function via automatic differentiation. The boundary network and flow network are coupled by minimising the loss function of the boundary condition as described by

$$\mathscr{L}_{b.c.} = \frac{\|\mathscr{F}_{flow}\left(t, \mathbf{X}_b^t; \boldsymbol{\Theta}_{flow}\right) - \mathbf{U}_b^t\|_2^2}{N_{b.c.}}. \tag{2.3}$$

The parameter $N_{b.c.}$ denotes the number of boundary points and $\|\cdot\|_2$ represents the ℓ_2 norm. Note that $(\mathbf{X}_b^t, \mathbf{U}_b^t)$ is estimated from the boundary network. Minimising $\mathscr{L}_{b.c.}$ implies that the boundary velocity computed by the boundary network must be equal to the flow velocity computed by the flow network.

The total loss is given by

$$\mathscr{L}_{total} = \mathscr{L}_{eqns} + \alpha \mathscr{L}_{b.c.} + \beta \mathscr{L}_{fdat} + \gamma \mathscr{L}_{bdat}. \tag{2.4}$$

In (2.4), $\mathscr{L}_{eqns}$ represents the residual of the N–S equations (Jin *et al.* 2021; Wang *et al.* 2022*a*), $\mathscr{L}_{fdat}$ represents the loss between the measured velocity $\mathbf{U}_{meas}$ and the velocity $\mathbf{U}_{pred}$ predicted by the flow network, and the sparse velocity measured by PIV or PTV uniformly distributes in the domain. The parameter $\mathscr{L}_{bdat}$ represents the loss between the measured boundary positions and the boundary positions predicted by the boundary network. The trainable variables $\boldsymbol{\Theta}_{b.c.}$ and $\boldsymbol{\Theta}_{flow}$ are simultaneously optimised by minimising the total loss. Due to the inherent multitasking property of the network, the weighting coefficients α, β and γ are used to balance the contributions of different loss terms in (2.4) (Wang *et al.* 2021). In the present work, we propose a novel approach to estimate these dynamic weights as follows:

$$\hat{\alpha}^{k+1} = \frac{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{eqns}|} / \sqrt{\mathscr{L}_{eqns}}}{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{b.c.}|} / \sqrt{\mathscr{L}_{b.c.}}}, \quad \hat{\beta}^{k+1} = \frac{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{eqns}|} / \sqrt{\mathscr{L}_{eqns}}}{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{fdat}|} / \sqrt{\mathscr{L}_{fdat}}},$$
$$\hat{\gamma}^{k+1} = \frac{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{eqns}|} / \sqrt{\mathscr{L}_{eqns}}}{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{bdat}|} / \sqrt{\mathscr{L}_{bdat}}}. \tag{2.5}$$

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

Here, $\nabla_{\boldsymbol{\Theta}}(\cdot)$ represents the gradient of the loss terms with respect to all of the trainable variables $\boldsymbol{\Theta}$ and can be computed via automatic differentiation with any deep learning framework. Additionally, $\overline{|\nabla_{\boldsymbol{\Theta}}\mathscr{L}_{eqns}|}$, $\overline{|\nabla_{\boldsymbol{\Theta}}\mathscr{L}_{b.c.}|}$, $\overline{|\nabla_{\boldsymbol{\Theta}}\mathscr{L}_{fdat}|}$ and $\overline{|\nabla_{\boldsymbol{\Theta}}\mathscr{L}_{bdat}|}$ denote the means of the absolute values of $\nabla_{\boldsymbol{\Theta}}\mathscr{L}_{eqns}$, $\nabla_{\boldsymbol{\Theta}}\mathscr{L}_{b.c.}$, $\nabla_{\boldsymbol{\Theta}}\mathscr{L}_{fdat}$ and $\nabla_{\boldsymbol{\Theta}}\mathscr{L}_{bdat}$, respectively. Equation (2.5) implies that the loss terms are balanced using both the gradient and loss statistics. In Appendix A, the present dynamic weight strategy is comprehensively compared with that proposed by Jin *et al.* (2021), where the dynamic weights are estimated using only the loss gradient statistics. The results indicate that our method is more robust and accurate. For more details on the comparison, please refer to the Appendix. Consequently, the weighting coefficients in the $k+1$th iteration are updated using a moving average strategy as $(\cdot)^{k+1} = (1-\eta)(\cdot)^{k} + \eta\hat{(\cdot)}^{k+1}$. The parameter η is a hyperparameter and is fixed at 0.1 (Jin *et al.* 2021).

An adaptive swish activation function is used in the present network. This function is expressed by

$$\sigma(x) = \frac{x}{1+e^{-ax}}, \tag{2.6}$$

where $a = n\zeta$, ζ is a trainable parameter and n is a scale factor. According to Jagtap *et al.* (2020), the scale factor n is set to 10, and the value of ζ is initialised to 0.1. For more details on the impact of the activation functions on the PINN performance, please refer to the paper by Wang *et al.* (2022*a*). Additionally, the inputs and outputs of the network are normalised. The input spatiotemporal coordinates are scaled by the minimum and maximum, and output velocities are scaled by the mean and standard deviation values. These parameters (the minimum, maximum, mean and standard deviation) are estimated according to the supervised data, whereas the mean and standard deviation of pressure are assumed to be zero and one. In Appendix B, we assess the influence of the spatial discretisation of the equation points on the pressure reconstruction performance. The residual of the equations is collected via two discretisation strategies: random uniform distribution and near-wall grid refinement. The error when the random uniform distribution is used is slightly smaller than that when near-wall grid refinement is used. Therefore, random uniform residual points for equations are adopted in this work. In Appendix C, the performance of the PINN-based pressure reconstruction method is assessed using the data for two axisymmetric corotating vortex rings, where the velocity is presented in cylindrical coordinates. This case proves that in addition to Cartesian coordinates, a PINN can also be used in cylindrical coordinates. We also investigate the effect of the amount of training data on the performance of pressure reconstruction based on the 2-D flow over an oscillating cylinder (§ 3.1) and fish swimming (§ 3.2) in Appendix D. The results indicate that the supervised data needed by the proposed PINN-MB are only 0.1 % of the CFD points. The neural networks are trained on a supercomputing platform with multiple NVIDIA GeForce RTX 4090 GPUs. The training variables are updated using the adaptive moment estimation (Adam) optimiser with an initial learning rate of 2×10^{-3} for both the flow network and the boundary network. The learning rate decays exponentially in a stepwise manner, and the learning rate of the last 2000 epochs is fixed at 1×10^{-5} to fine-tune the result.

2.2. *Advantages of PINN-MB by comparison with PINN and the numerical method*

This section compares PINN-MB with the original PINN and the numerical method to demonstrate its advantages. The proposed PINN-MB contains a boundary network $\mathscr{F}_{b.c.}$ and a flow network $\mathscr{F}_{flow}$; the former is adopted to track the motion of boundaries

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

	Inputs	Implementation	Optimisation	Prediction
PINN-MB	Flow fields boundary locations	Simple	optimise the flow fields and boundaries	Predict the flow fields and boundaries
PINN	Flow fields boundary locations boundary velocities[1]	Simple	optimise only the flow fields	Predict only the flow fields
Numerical method	Flow fields boundary locations boundary velocities[1]	Complex[2]	No optimisation[3]	No prediction[3]

Table 1. An overview of the differences among the PINN-MB, PINN and numerical method.
[1]The boundary velocity is deduced from the time derivative of the boundary location.
[2]The numerical method is difficult to construct the difference equation near the boundary.
[3]There is no capability for optimisation or prediction.

and the latter is adopted to interpolate the flow fields with the constraint of the N–S equation. The boundary velocity $\mathbf{U}_b^t$ is calculated from the optimal boundary $\mathbf{X}_b^t$ via automatic differentiation of the boundary network. The boundary condition $(\mathbf{X}_b^t, \mathbf{U}_b^t)$, including location and velocity, can be predicted at an arbitrary instant from the boundary network. PINN can also estimate the pressure field if the boundary condition is given as known data. However, for a real experiment, the boundary velocity should be estimated by finite difference or by fitting the noisy boundary positions. This may introduce a very high uncertainty arising from the time derivative of the inaccurate boundary position. Moreover, a PINN is incapable of boundary optimisation and prediction, as given in Appendix E. For the numerical method, the pressure can be easily determined from the given velocity field through the Poisson equation if the boundary is fixed. However, the local acceleration $(\partial/\partial t)$ near the boundary is difficult to estimate because of the boundary's deformation, except for adopting the immersed boundary method or the moving grid strategy. Meanwhile, the accuracy of the resulting pressure computation is sensitive to the propagation of the errors of velocity and boundary (van Oudheusden 2013). The advantages of PINN-MB over the PINN and the numerical method are summarised in table 1. In brief, PINN-MB can solve for the pressure from the optimised velocity and boundary.

3. Results

3.1. *Example 1: flow over an oscillating cylinder*

The performance of the proposed method is first validated using a simulated 2-D flow over an oscillating cylinder at a diameter-based Reynolds number of $Re_D = 185$, where D denotes the diameter of the cylinder. The computational domain spans $-15D \leqslant x \leqslant 45D$ and $-20D \leqslant y \leqslant 20D$, and the cylinder is initialised at the origin (0, 0) and moves in the cross-flow (y-axis) direction according to $y_c = 0.2sin(2\pi f_c t)$. The normalised parameter f_c is set to 0.156 according to Wang *et al.* (2019). The ground truth velocity $\mathbf{U}$ and pressure fields p are obtained by solving the non-dimensional incompressible N–S equations:

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U}\cdot\nabla)\mathbf{U} = -\nabla p + \frac{1}{Re_D}\nabla^2\mathbf{U},$$
$$\nabla\cdot\mathbf{U} = 0. \tag{3.1}$$

All variables are normalised by the fluid density ρ, the free streamwise velocity U_∞ and the diameter D. This simulation is conducted using OpenFoam with a moving grid strategy

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

over a body-fitted and wall-refined mesh. The non-dimensional time step in the simulation is set to 2.5×10^{-3}, with a maximum Courant–Friedrichs–Lewy number of 0.7. A no-slip condition is imposed at the cylinder surface. Note that the data are stored every twenty steps, which means that the stored time interval is 5.0×10^{-2}. After the CFD data are obtained, the scattered velocity data for training are collected by interpolating the flow fields on a uniform grid to mimic the typical PIV data. The inspected region is limited to the subdomain of $-1D \leqslant x \leqslant 3D$ and $-1D \leqslant y \leqslant 1D$. The vector space is denoted by $1/R$, where R is the number of points along the diameter. A small R indicates a low spatial resolution. There are 721 boundary points distributed uniformly on the surface of the cylinder, and the time sequence is the same as that of the velocity data. In addition, evaluation of the influence of measurement noise on the performance of the proposed method is vital. Therefore, Gaussian random noise is added to the training data (including the boundary data). The noise level σ_n is defined as the ratio of the mean noise magnitude to the velocity magnitude. The signal-to-noise ratio (SNR) can be simply estimated as $\mathrm{SNR} = 10\log_{10}1/\sigma_n^2$. We randomly sampled 1×10^6 data points and 0.5×10^6 boundary points from four hundred velocity fields without pressure information as training data, 1×10^6 points to estimate the residuals of the N–S equations and 0.5×10^6 points to estimate the boundary conditions.

The PINN-MB network is built for pressure reconstruction in scenarios with a moving boundary. The boundary network contains nine hidden layers and 32 neurons per hidden layer, and the flow network contains 13 hidden layers and 128 neurons per hidden layer, yielding a total of 207 707 trainable parameters. This means that we used 207 707 parameters to capture the spatiotemporal evolution of the flow over the oscillating cylinder, whereas the numerical simulation used approximately 1.4×10^7 grid points (400 velocity fields, each field containing 35 940 CFD points) in the subdomain to compute the flow fields. The network was trained for 4000 epochs with 30 iterations per epoch, and the batch size was set to 10 000. The initial position $\mathbf{X}_b^0$ was set to a standard circle with $D = 1$ without measurement noise. The boundary noise at the initial position can reasonably be ignored because the experimental model is accurately designed in advance. No pressure information is provided for training. An example snapshot of the velocity and pressure is shown in figure 2. The training data with $R = 20$ and $\sigma_n = 0\,\%$, the data learned over a high-resolution mesh with $R = 100$, and the absolute error between the exact and learned data are presented from top to bottom. From left to right, the contour maps represent u, v and p, respectively. At this instant, the cylinder moves forward along the y-axis, as indicated by the yellow arrows. The pressure is greater in the stagnation region and lower in the vortex region. Notably, the mean pressure at each time instant is subtracted to enable comparative assessments. A visual inspection of the results suggests that the predicted high-resolution data are approximately the same as the exacted data, and the maximum absolute error of the pressure coefficient is approximately 4 % of the maximum pressure. In figure 3, noise with $\sigma_n = 8\,\%$ was added to both the velocity data and boundary data. The velocity, pressure and boundary of the cylinder can still be recovered from the noisy data, although the absolute errors are larger than those in the case without noise. As shown in figures 2 and 3, a large error ($\sim$0.05) is always observed in the region near the boundary for high velocity gradients. This visual inspection shows that the proposed method can accurately determine the pressure fields in scenarios with a moving boundary.

To provide a quantitative comparison, we present the relative ℓ_2 norm error of the pressure as a function of R and σ_n in figure 4. The relative ℓ_2 norm error is defined as follows:

$$\delta_p = \frac{\|p_{ref} - p_{pred}\|_2}{\|p_{ref}\|_2}. \tag{3.2}$$

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

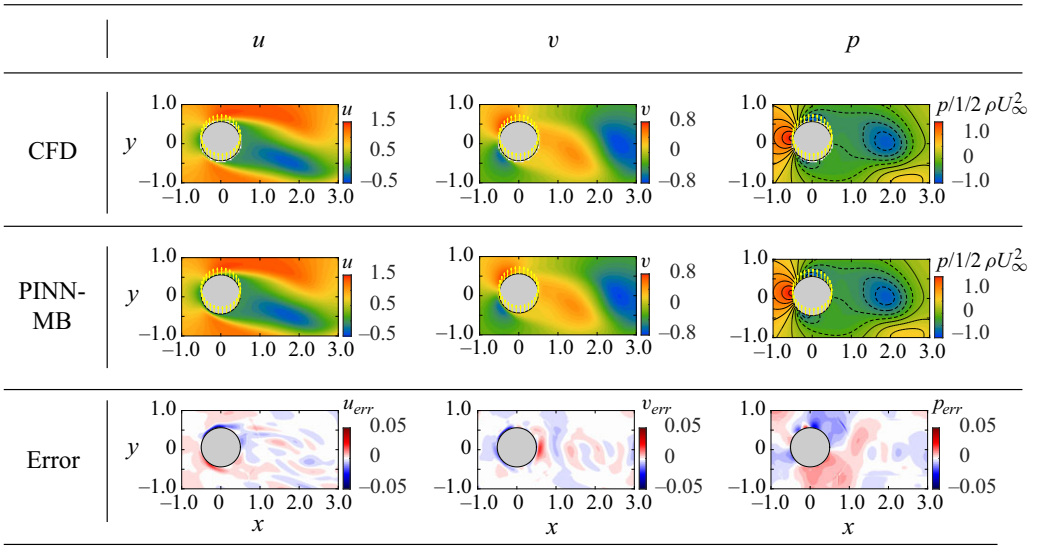


Figure 2. A representative snapshot of the velocity, pressure and corresponding absolute error. From left to right, the contour maps represent u, v and pressure p. The dashed black lines represent negative contour levels from -1.5 to -0.1 with an interval of 0.2, and the solid black lines represent positive contour levels from 0.1 to 1.5 with an interval of 0.2. Note that the spatially averaged pressure is subtracted from both the exacted data and the predicted data for comparative assessment. For the training data, the vector space $1/R$ is set to $1/20$, and the noise level is set to 0 %. The learned data were predicted over a mesh with an interval of $1/R = 1/100$. Therefore, the original CFD data were interpolated on the predicted mesh to compute the absolute error. The yellow arrows indicate the motion of the cylinder.

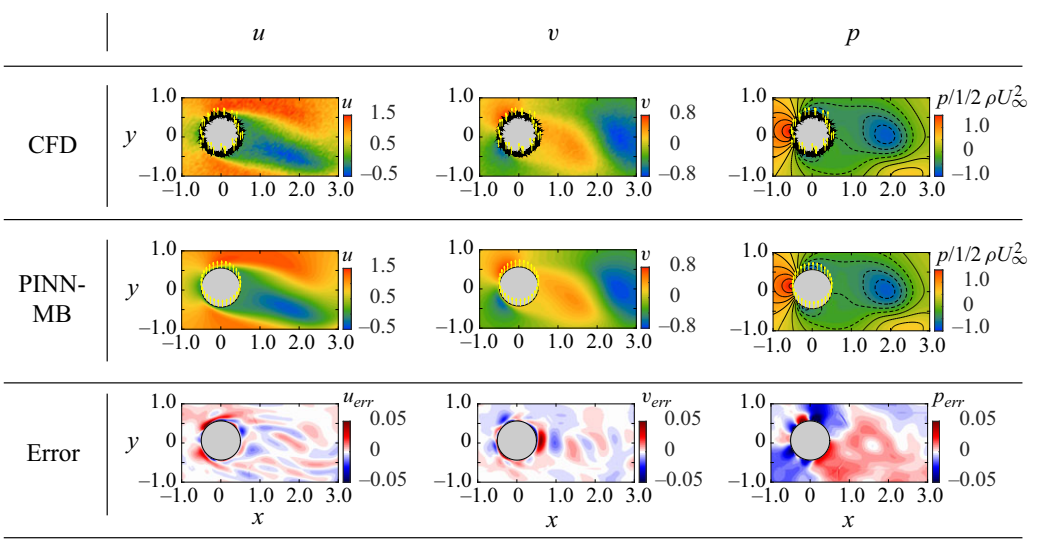


Figure 3. A representative snapshot of the velocity, pressure and corresponding absolute error. From left to right, the contour maps represent u, v and pressure p. The dashed black lines represent negative contour levels from -1.5 to -0.1 with an interval of 0.2, and the solid black lines represent positive contour levels from 0.1 to 1.5 with an interval of 0.2. Note that the spatially averaged pressure is subtracted from both the exacted data and the predicted data for comparative assessment. For the training data, the vector space R is set to $1/20$, and the noise level is set to 8 % (the corresponding SNR is 21.9). The learned data were predicted over a mesh with an interval of $R = 1/100$. Therefore, the original CFD data were interpolated over the predicted mesh to compute the absolute error. The yellow arrows indicate the motion of the cylinder.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

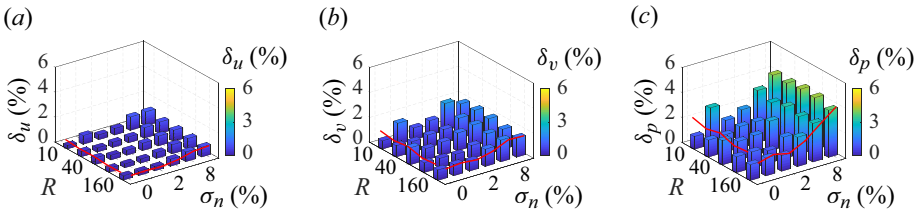


Figure 4. Relative ℓ_2 norm error as a function of noise level σ_n and vector space $1/R$: (*a*) u component, (*b*) v component and (*c*) pressure. In each case, the same PINN-MB model is used. The boundary network contains nine hidden layers and 32 neurons per hidden layer, and the flow network contains 13 hidden layers and 128 neurons per hidden layer. The red curves represent the mean values along the noise level or vector space.

The subscripts ref and $pred$ denote the reference value and the predicted value, respectively. The relative ℓ_2 norm error of the velocity components is also estimated using the same formula. All errors are averaged over space and time. Before presenting the results, we note that PINN-based pressure reconstruction has been compared with traditional approaches within pure flow fields (Fan *et al.* 2023). The previous work has shown that the PINN-based method is superior to the other two methods because the former can optimise the velocity fields. In the present work, a dual-network model is used to determine the pressure from velocity with a moving boundary. This method can address not only rigid bodies but also flexible boundaries, as shown in the next three examples. Returning to figure 4, it is evident that the pressure error (~ 4 %) is larger than the velocity error, which is consistent with the results of Wang *et al.* (2022*a*) and Fan *et al.* (2023). For the velocity, the relative error in the v component is slightly larger than that in the u component for the difference in magnitude. These figures show that the impact of the sparsity of the data on the pressure reconstruction performance is much smaller than that of the measurement noise level. The pressure error at $R = 10$, which corresponds to only ten points along the diameter, is nearly identical to that at $R = 160$. However, the pressure error increases from 1 % to 4 % as the noise level σ_n increases from 1 % (the corresponding SNR is approximately 40) to 16 % (the corresponding SNR is approximately 16) in this test case. These results demonstrate that reducing measurement noise and improving data accuracy are critical for pressure reconstruction from measured velocity data. Based on the quantitative analysis, we conclude that the proposed PINN-MB method is feasible and accurate for pressure reconstruction with a moving boundary.

Figure 5(*a*) presents the pressure coefficient around the cylinder at the time when the cylinder moves to the maximum vertical position of $y_c = 0.2$. The parameter θ denotes the angle of the surface points, as shown in figure 5(*a*). Note that the curves are obtained by averaging the data over all the instants when the cylinder reaches the maximum vertical position. Two prediction results with $R = 20$, $\sigma_n = 1$ % and $R = 160$, $\sigma_n = 8$ % are displayed in this figure. Both curves show good agreement with the CFD data, but the pressure coefficient predicted with the higher noise level has a larger deviation from the CFD data in the region of $80° \leqslant \theta \leqslant 120°$. Moreover, we present the power spectral density (PSD) of the pressure at (2, 0) in figure 5(*b*). The parameter f denotes the non-dimensional frequency. The distributions of the spectra obtained using the proposed neural network are consistent with the reference CFD data in the region of $0.2 \leqslant f \leqslant 3$. The deviation at a low frequency of $f = 0.1$ may be attributed to the larger error in the predicted pressure at the beginning and end of the corresponding time, as shown in the top plane of figure 6. Additionally, we present the cylinder motion, i.e. the y-axis centroid of the cylinder, in this figure. No evident difference among the three curves are observed. We next analyse the PSD with high frequencies $f > 3$, which shows that the predicted

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

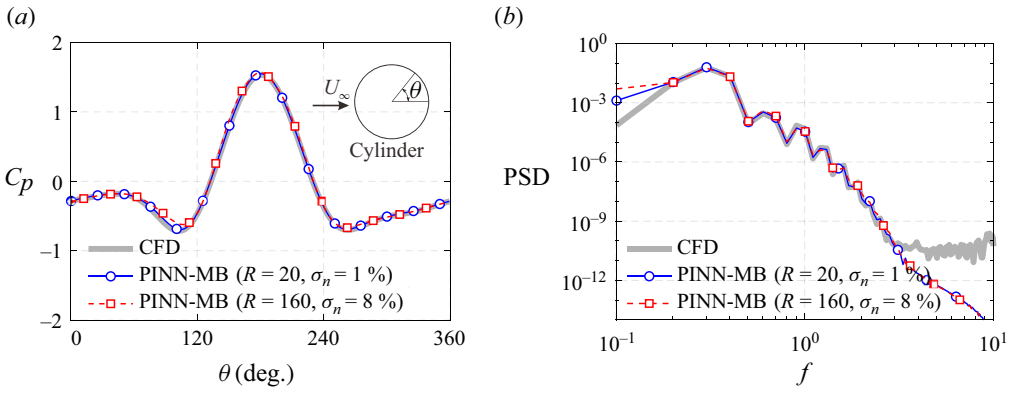


Figure 5. (*a*) Pressure coefficient around the cylinder at the instant when the vertical position of the cylinder reaches a maximum of $y_c = 0.2$. The parameter θ denotes the angle of the surface points. (*b*) PSD of the reference pressure and the predicted pressure at (2, 0), where f denotes the nondimensional frequency.

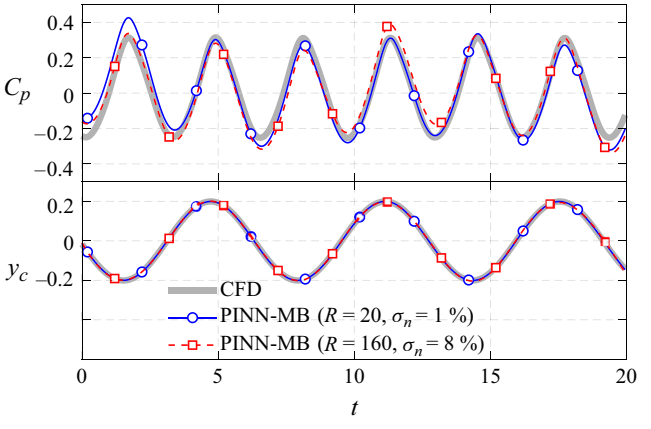


Figure 6. Time series of the pressure coefficients C_p at point (2,0) (the top panel) and the cylinder location y_c (the bottom panel). y_c denotes the y-axis centroid of the cylinder.

PSD is much lower than that of the reference curve in the high-frequency region. There are two possible reasons for this result. First, the interpolation from the CFD results to a regular mesh may cause errors at high wavenumbers. Second, the network may prefer to learn low-frequency modes rather than high-frequency modes, a phenomenon known as 'spectral bias' (Rahaman *et al.* 2019). There are several methods for overcoming this well-known issue. Eldan & Shamir (2016) proved that a simple function expressed by a small 3-layer feedforward neural network cannot be approximated by any 2-layer network unless its width is exponential in dimension. This implies that a deeper network is needed for learning at a high frequency. Tancik *et al.* (2020) showed that mapping the input variables via Fourier features enables a multi-layer perceptron to learn high-frequency functions in low-dimensional problem domains. This Fourier feature mapping significantly improves the high-frequency performance in the neural radiance field for view synthesis (Mildenhall *et al.* 2020). Moreover, for the super-resolution reconstruction of turbulence, generative adversarial networks have the superior capabilities for representing small-scale structures compared with convolution neural networks (Kim *et al.* 2021).

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

3.2. *Example 2: fish swimming*

A more complex case involves the reconstruction of the pressure around a swimming fish. Fish locomotion has attracted considerable attention from engineers and scientists because understanding the hydrodynamics of the fish is important for bionic underwater vehicle design. The fish uses the pressure signals detected along the lateral line to sense the surrounding environment (Calicchia *et al.* 2023), and generates propulsive forces by creating jets and vortices in the wake flow (Triantafyllou *et al.* 2000; Ting & Yang 2009; Tu *et al.* 2022). A two-dimensional direct numerical simulation (DNS) of a swimming fish is carried out using an in-house CFD solver (Liu *et al.* 2018, 2023) to validate the accuracy of the proposed method.

A two-dimensional NACA0012 model is used to mimic the fish body. The fish axis lies on the x-axis, and the head and tail are located at (0,0) and (1,0), respectively. The lateral displacement of the fish axis is defined as the carangiform swimming style, as measured by Videler & Hess (1984). This motion is prescribed as the product of an envelope $A_m(x)$ and a backward travelling wave as follows:

$$y/L = A_m(x)\sin\left(\frac{2\pi}{\lambda}(x - ct)\right). \tag{3.3}$$

Here, $A_m(x) = b_0 + b_1(x/L) + b_2(x/L)^2$, $b_0 = 0.02$, $b_1 = -0.08$ and $b_2 = 0.16$. The wavelength λ is equal to the fish length L, and the wave velocity c is set to the product of the tail beat frequency f and the wavelength λ. In this case, the parameters are set to the following values: $L = 0.1$ m, $f = 2$ Hz and the free stream velocity $U = 0.1\ \mathrm{m\,s^{-1}}$. Therefore, the Reynolds number, which is based on the body length and free stream velocity, $Re = LU/\nu$, is 1×10^4, and the Reynolds number based on the body length and tail beat frequency, $Re = L^2 f/\nu$, is 2×10^4. The non-dimensional time step normalised by the characteristic time L/U in the simulation is $\mathrm{d}t = 2 \times 10^{-3}$, whereas the time interval of the stored data is $\Delta t = 2\mathrm{d}t = 4 \times 10^{-3}$ for every two time steps to store the flow fields. Two hundred flow fields interpolated into a uniform grid in the domains $-0.3 \leqslant x/L \leqslant 1.3$ and $-0.3 \leqslant y/L \leqslant 0.3$ are postprocessed to train the network, and the vector space is set to $L/400$. Boundary data with 682 points were extracted exactly from the numerical simulation, were non-uniformly distributed on the surface of the fish, and the time sequence was the same as that of the velocity data. To test the effect of the temporal resolution on the performance of the proposed method, data for five cases with time steps $\Delta t = 2\mathrm{d}t$, $4\mathrm{d}t$, $6\mathrm{d}t$, $8\mathrm{d}t$ and $10\mathrm{d}t$ are generated to train the network. The numbers of corresponding velocity fields are 200, 100, 67, 50 and 40. For each case, we randomly sample 2×10^6 data points and 1×10^6 boundary points without pressure information as training data, 2×10^6 residual points to estimate the loss of N–S equations and 1×10^6 points are used for the boundary conditions. In this example, we do not consider the measurement noise.

The configuration of the PINN-MB model is set as follows: the boundary network contains nine hidden layers and 48 neurons per hidden layer, and the flow network contains 13 hidden layers and 160 neurons per hidden layer, yielding a total of 329 371 trainable parameters. The number of trainable parameters is much less than the CFD points in this subdomain (two hundred velocity fields with 76 982 CFD points per field). This network is trained for 18 000 epochs with 30 iterations per epoch and the batch size is set to 10 000. Note that all of the velocities, pressures and boundaries are still predicted in the vector space of $L/400$ with a time separation of $2\mathrm{d}t$. We first investigate whether PINN-MB can capture fish motion under these complex conditions. Figure 7 shows a comparison of the predicted fish bodies with those extracted from the CFD results. The red and blue

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

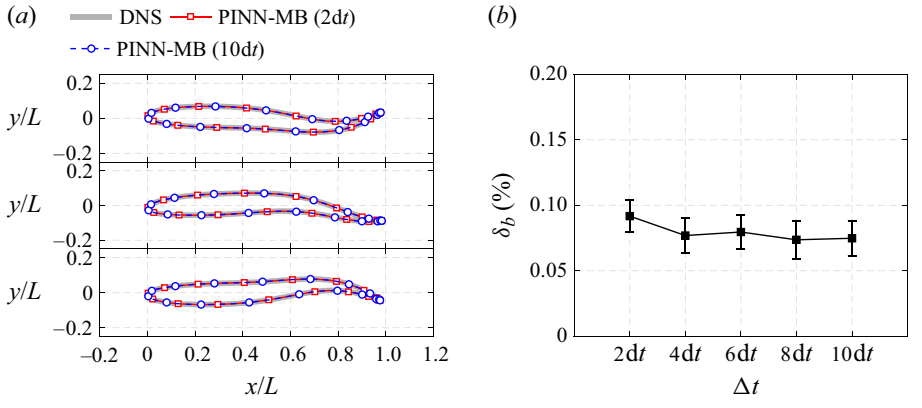


Figure 7. (*a*) Exact (grey thick curves) and predicted fish surfaces with $\Delta t = 2\mathrm{d}t$ (red curves) and $10\mathrm{d}t$ (blue curves) at different instants. From top to bottom, the non-dimensional time separation is 0.12. (*b*) Predicted error of the fish body as a function of the time steps of the training data Δt, the error bars represent the standard deviation. The error is normalised by the body length L. Note that, all of the data, including velocities and boundaries, are predicted at the time interval of $\Delta t = 2\mathrm{d}t$.

curves represent the predicted results when Δt is equal to $2\mathrm{d}t$ and $10\mathrm{d}t$, respectively. It is clear that the predicted curves are approximately the same as the original grey curves at different points in time, as the loss of boundary data for all the cases converges to less than 5.0×10^{-6}. We also plot the error of the predicted body surface as a function of the time steps of the training data in figure 7(b). The boundary error normalised by the body length is less than 0.1 % over the whole test region. This result indicates that PINN-MB can accurately predict the boundary for the coupling between the boundary and flow networks through the boundary conditions. Notably, the error of the predicted boundary is slightly larger at the beginning and end of the time (~ 0.2 %) because fewer constraints are imposed at these instants.

The instantaneous pressure around the fish at different time instants is shown in figure 8. The CFD results, PINN-MB results with $\Delta t = 10\mathrm{d}t$ and absolute error are displayed from left to right. The yellow arrows indicate the motion of the fish's body. The mean value is subtracted from the exacted and predicted data for comparative assessment. We do not show the predicted pressure with high temporal resolution $\Delta t = 2\mathrm{d}t$ because it is almost the same as that with low temporal resolution $\Delta t = 10\mathrm{d}t$. Generally, the PINN-MB model correctly predicts the pressure around the swimming fish. In particular, in the high-pressure region near the head, the pressure distribution is consistent with that of the DNS. At the first instant $t = 0.23$, as shown in figure 8, a strong suction zone with low pressure is created when the middle of the body bends to the right side (Thandiackal & Lauder 2020). A strong positive pressure is also consequently generated on the right side for pushing the water. PINN-MB can still correctly recover the pressure field at this instant. Visual inspection of the absolute error distribution in figure 8 shows that large errors are distributed mainly in the region of flow separation near the tail, where the flow is more complex than in the other regions. Figure 9 clearly shows the surface pressure coefficients extracted from the DNS and the predicted results. Note that PINN-MB can directly output the pressure at the boundaries without any interpolation. The predicted profiles with high temporal resolution $\Delta t = 2\mathrm{d}t$ and low temporal resolution $\Delta t = 10\mathrm{d}t$ show good agreement with the DNS results.

The results of the quantitative error analysis of the pressure reconstruction results for the swimming fish are presented in figure 10. We first present the relative ℓ_2 norm errors

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

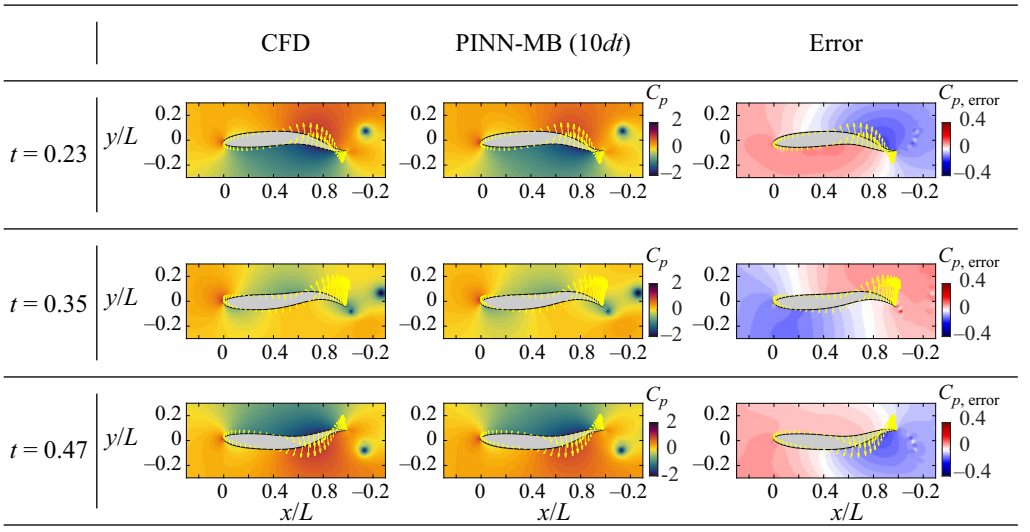


Figure 8. Representative snapshots of pressure for DNS, PINN-MB with $\Delta t = 10\mathrm{d}t$ and the absolute error. From top to bottom, the non-dimensional times are 0.23, 0.35 and 0.47, respectively. In this case, the vector intervals of both the training data and the predicted data are set to the same values of $L/400$. No noise is added to the training data. The yellow arrows indicate the motion of the fish's body. The mean value is subtracted from both the exacted data and the predicted data for comparative assessment.

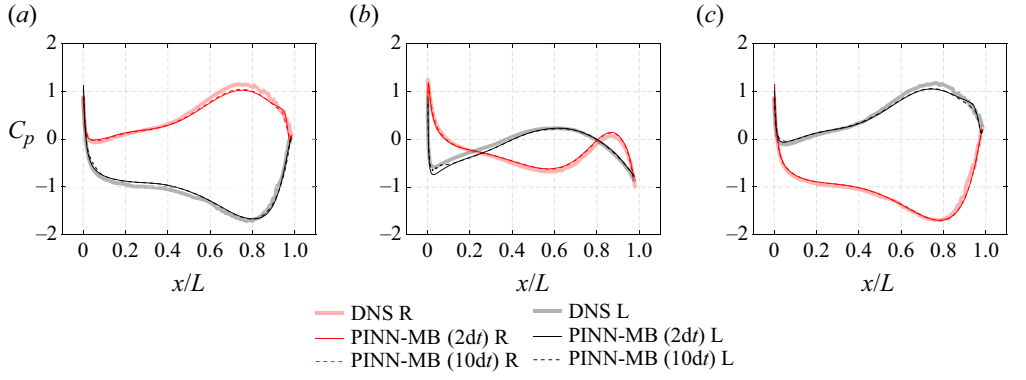


Figure 9. Comparison of the surface pressure coefficients between the CFD results (thick solid curves) and the results predicted by PINN-MB with $\Delta t = 2\mathrm{d}t$ (thin solid curves) and $\Delta t = 10\mathrm{d}t$ (thin dashed curves). The different colours denote the right (red, R) and left (black, L) sides of the fish body. From left to right, the non-dimensional times are 0.23, 0.35 and 0.47, respectively.

for the whole field and boundary as a function of time at a high temporal resolution of $\Delta t = 10\mathrm{d}t$. Large error fluctuations of 10 % to 30 % are observed in this figure. The error reaches a maximum at $t = 0.35$ and decreases to a minimum at $t = 0.23$ and 0.47. This is because the relative error is normalised by the magnitude of the reference pressure (CFD result), which varies over time. As shown in figures 8 and 9, the pressure magnitude at $t = 0.35$ is smaller than that at $t = 0.23$ and 0.47. Therefore, the relative error is larger at $t = 0.35$. We also present the mean pressure error as a function of the time steps of training data in figure 10(*b*). The mean pressure errors for the fields and boundaries are 14 % and 20 %, respectively. The error profiles remain relatively flat across all the temporal resolutions tested, which is consistent with the results reported by Calicchia *et al.* (2023).

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

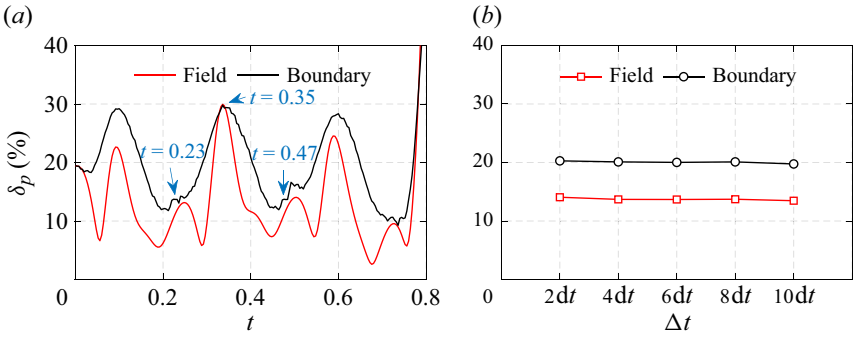


Figure 10. (*a*) Time series of the relative ℓ_2 norm error between the predicted pressure and the DNS pressure. (*b*) Error of the predicted pressure as a function of the time steps of the training data Δt. Note that all the data, including the velocities and boundaries, are predicted at a time interval of $\Delta t = 2\mathrm{d}t$. The red and black curves denote the errors of fields and boundaries, respectively.

This result indicates that PINN-MB is insensitive to the temporal resolution of the training data for the embedding of the N–S equations.

3.3. *Example 3: 2-D pressure in the left ventricle*

The primary function of the LV is to pump oxygenated blood into the aorta and then to every part of the body. Left ventricle dysfunction caused by structural injuries or pathological changes can induce pathological remodelling of the LV, which can lead to heart failure (HF) with significant morbidity and mortality (Stoll *et al.* 2019). Many studies have investigated cardiovascular imaging to quantitatively assess cardiac function in terms of morphology and structure. However, ventricular flow is altered in the early stages of remodelling (Stoll *et al.* 2019), which means that flow properties can be used to diagnose and prognose cardiac function (Pedrizzetti *et al.* 2014). The haemodynamic force, which corresponds to the global value of the intraventricular pressure gradient (IVPG), was suggested as a potential indicator for detecting early alterations in LV dysfunction and predicting disease outcome (Lapinskas *et al.* 2019; Mele *et al.* 2019; Stoll *et al.* 2019; Vallelonga *et al.* 2021). Therefore, *in vitro* experiments were conducted to measure the velocity distribution in the LV and reconstruct the intraventricular pressure using the proposed PINN-MB method for the motion of the ventricular wall.

Figure 11 shows the set-up of the time-resolved 2-D PIV measurements. The pulsatile flow was simulated using an in-house LV simulator driven by a piston pump. The displacement of the piston was controlled by a user-defined signal, which is a conversion of the time evolution of the left ventricular volume extracted from Pedrizzetti & Perktold (2003). The 3-D-printed silicone LV model (refractive index of 1.41, thickness of 2.0 mm, end-systolic volume of 80 ml, base-to-apex length of 82 mm) was immersed in an acrylic hydraulic chamber filled with the same flow media. The systole and diastole of the LV were imitated by increasing and decreasing the pressure in the hydraulic chamber by using a piston pump. This hydraulic chamber was designed to have nine sides, as proposed by Saaid *et al.* (2019), to facilitate 2-D or 3-D PIV measurements. During systole, blood flows from the LV to the aorta through the aortic valve, and then flows through the compliance chamber, resistance valve, fluid reservoir and left atrium before returning to the LV through the mitral valve. The compliance chamber downstream of the aortic valve is used to mimic arterial compliance. A resistance valve is used to adjust the systematic resistance. The fluid reservoir is open to the atmosphere to simulate the low pressure of the venous system.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

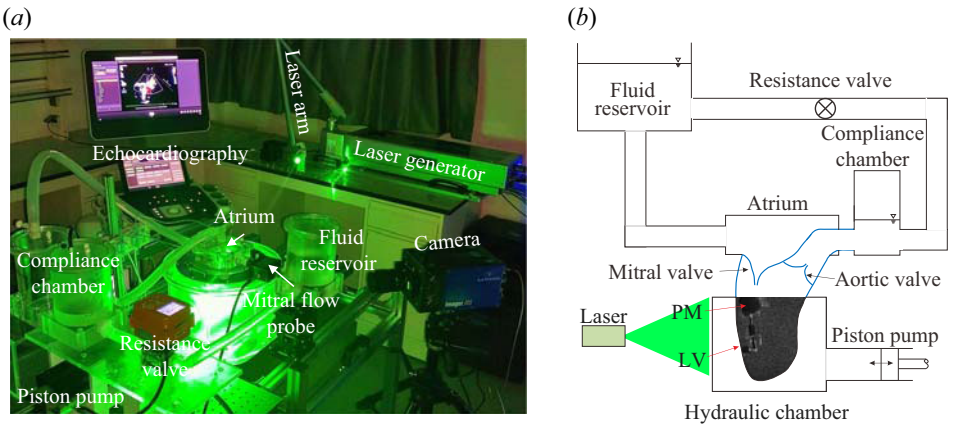


Figure 11. Set-up of the time-resolved 2-D-PIV measurements: (*a*) picture of the set-up, (*b*) schematic of the set-up. The native mitral and aortic valves, which were anatomised from a fresh porcine heart, were connected to a 3-D-printed silicone LV. As the piston pulls back, the aortic valve is closed due to flow deceleration, and the blood analogue flows through the opened mitral valve to fill the LV. As the piston moves forwards, the mitral valve is closed, and blood is pumped into the aorta, which is driven by rapidly increasing pressure. Note that a part of the flow field is sheltered by the papillary muscles.

Native mitral and aortic valves were anatomised from a fresh porcine heart weighing approximately 330 g, because the porcine heart has been used in preclinical trials and has similar haemodynamic performance to that of humans (Gooden *et al.* 2022). In this experiment, a complete mitral valve consisting of the mitral annulus, leaflets, chordae tendineae and papillary muscles was sutured to a specially designed plate, and this plate was connected with a 3-D-printed silicone LV. The papillary muscles are attached to the LV wall with connecting rods, ensuring synchronous movement with the LV. However, the papillary muscles block the laser light, producing a shadow without particles, as shown in figure 11(*b*). A mixed solution (62 % water, 38 % glycerol by volume and 60 g/L NaCl by weight at a working temperature of 26 °C) with a dynamic viscosity of 3.49 mPa·s, a density of 1173 kg/m^3 and a refractive index of 1.395 is used as blood analogue. The refractive index does not perfectly match that of silicone, and a better mixed solution was reported by Brindise *et al.* (2018) and Chen & Dasi (2023).

Because this work focused on pressure reconstruction from velocity fields, we provide only a brief introduction to the physiological conditions. The heart rate was 60 bpm and the stroke volume was 60 ml, this configuration yielded a cardiac output of 3.6 l/min. Figure 12 shows the transmitral pressure gradient (ventricular pressure–atrial pressure) and mitral flow rate during the cardiac cycle. In systole, the maximum pressure gradient is greater than 80 mmHg, and in diastole, the maximum instantaneous flow rate is approximately 8.3 l/min. The mean and maximum mitral inflow Reynolds numbers are 837 and 1930, respectively, computed as $Re = UD/\nu$, where U denotes the mean and maximum mitral velocities deduced from the cardiac output and the maximum instantaneous flow rate. The parameter D denotes the diameter of the mitral valve which is equal to 26 mm in this case. The associated Womersley number of the mitral inflow, defined as $Wo = D/2\sqrt{2\pi/T\nu}$, where T is the time of the cardiac cycle, is approximately 17. The systematic vascular resistance (SVR) is approximately 777 dyn·s·cm^{-5}, which is slightly less than that of a healthy human (800–1200 dyn·s·cm^{-5}).

Under these conditions, time-resolved 2-D PIV experiments were performed to measure the velocity field at the plane of the left ventricle outflow tract (LVOT). This plane

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

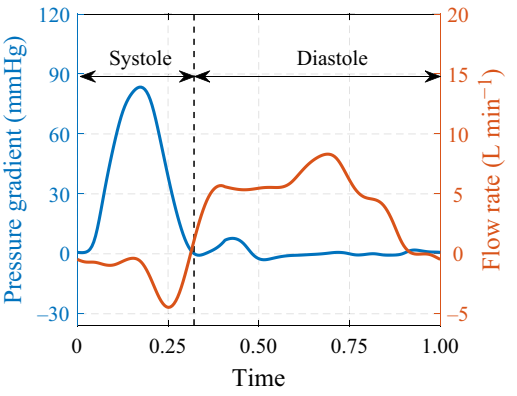


Figure 12. Flow rate (red curve) and transmitral pressure gradient (ventricular pressure–atrial pressure, blue curve) for the native mitral valve during the cardiac cycle. The time is normalised with respect to the period.

is commonly used in clinical practice to evaluate blood flow and cardiac function. Moreover, the flow over the LVOT is approximately two-dimensional because this plane is a geometrically symmetric plane of the LV. The flow was seeded with 7 μm fluorescent Rhodamine-B-coated particles at a density of 1500 kg/m^3. The response frequency was ∼945 kHz, which means that the fluorescent particles can fully follow the flow motion. These fluorescent particles were excited using a double-pulsed laser (Beamtech, Vlite-Hi-50) with a wavelength of 527 nm at a repetition rate of 0.2 kHz. The laser beam was guided to the hydraulic chamber by using a laser arm and the thickness of the laser sheet was approximately 1.0 mm. Orange fluorescence with a longer wavelength of 590 nm was emitted and recorded by a high-speed CMOS camera (Lavision Imager HS CMOS cameras) equipped with a 100 mm/f2.8 lens (Tokina atx-i) with a high-pass filter. The image pairs were recorded at 200 Hz with a full image resolution of 2016×2016 pixels, and the time separation was set to 400 μs to ensure that the maximum particle displacement was less than 8 pixels. This configuration yields a digital resolution of 20 pixels/mm. The velocity fields were calculated using in-house PIV software with a three-pass window deformation iterative multigrid scheme (Scarano 2002; Wang *et al.* 2020*a*). The interrogation window size was reduced to 32×32 pixels with 75 % overlap, and the velocity at the final pass was smoothed using Gaussian filtering with the same size as the interrogation window. This configuration yields a spatial resolution of 1.6 mm, and the vector step is 0.4 mm. The LV edge of each frame is detected according to the image intensity gradient and converted to physical coordinates based on the mapping function of the PIV data. Each edge contains 721 points with uniform distribution. In the present work, four cardiac cycles, consisting of 800 velocity fields with boundary data, were used to reconstruct the pressure fields.

To recover the pressure fields, the boundary network consists of nine hidden layers and 64 neurons per hidden layer, and the flow network contains 13 hidden layers and 200 neurons per hidden layer. This configuration yields 517 491 trainable parameters. The location of the LV wall at the first time step is chosen as the initial input X_b^0 for the boundary network. Approximately 1.4×10^6 velocity vectors are randomly selected from the PIV data and used as supervised data for the flow network, 1.0×10^6 boundary data are randomly selected as supervised data for the boundary network and 1.1×10^6 equation points with a uniform distribution are used to collect the residuals of the 2-D incompressible N–S equations. The dual-network model was trained for 10 000 epochs with 30 iterations per epoch. The corresponding loss functions are presented

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

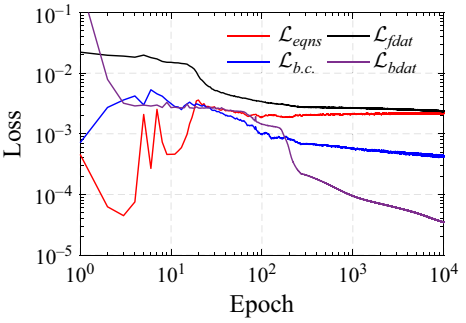


Figure 13. Loss functions (equation loss $\mathscr{L}_{eqns}$, boundary condition loss $\mathscr{L}_{b.c.}$, flow data loss $\mathscr{L}_{fdat}$ and boundary data loss $\mathscr{L}_{bdat}$) as a function of epochs for the 2-D experimental case of LV.

in figure 13. The training losses gradually converge to stable values after 1000 epochs. The final equation loss $\mathscr{L}_{eqns}$, boundary condition loss $\mathscr{L}_{b.c.}$, flow data loss $\mathscr{L}_{fdat}$ and boundary data loss $\mathscr{L}_{bdat}$ are approximately 2×10^{-3}, 4×10^{-4}, 2×10^{-3} and 3×10^{-5}, respectively. The equation loss is greater than that in the 3-D example as shown in figure 17(*a*), this is because of the two-dimensional assumption applied at the plane of LVOT. Figure 14 presents a series of instantaneous flow fields in three different phases (from left to right: early diastole, end diastole and systole) during the cardiac cycle. From top to bottom, these figures display the contour maps of the original PIV velocity magnitude and the velocity magnitude and pressure field predicted by PINN-MB. Note that the blank region within the original velocity is caused by the shadow of the papillary muscles as shown in figure 11(*b*). However, the proposed method has the capability to physically fill the gappy flow fields, the resulting fields are shown in figure 14. Because the ground truth pressure is unknown, we can only draw conclusions from a qualitative examination. In early diastole, when the mitral jet enters the LV, the inflow is decelerated by the increasing pressure gradient directed from the apex to the base. At the end of diastole, the pressure is high when the jet impinges onto the apex. During systole, the pressure is greater at the apex than at the base and the pressure accelerates the outgoing blood flow. However, the pressure gradient between the apex and base is much lower than that during diastole. Physically, the pressure presents a low level in the vortex ring centre induced by the mitral jet. This evolution of the IVPG is consistent with the clinical ultrasound results of a normal person (Pedrizzetti *et al.* 2014; Mele *et al.* 2019).

Once the surface pressure is obtained, the haemodynamic force imposed on the blood can be calculated as the integral of the surface pressure distribution:

$$\mathbf{F}(t) = -\int_{S(t)} p\mathbf{n}\mathrm{d}S. \tag{3.4}$$

Here, $S(t)$ is the surface of the LV and $\mathbf{n}$ is the outward unit normal vector. Intraventricular blood flow is driven by the pressure gradient, and the haemodynamic force $\mathbf{F}(t)$ represents a global force exchanged between the blood volume and the LV wall (Pedrizzetti 2019; Vallelonga *et al.* 2021). Compared with the conventional approach for the quantitative assessment of cardiac function, such as the ejection fraction, the use of haemodynamic force may provide insights into the interaction between tissue mechanics and fluid dynamics in the LV. Figure 15 displays the haemodynamic force in a period of the cardiac cycle. We note that this force is normalised with respect to the density, area (volume for the 3-D case) and gravitational acceleration for comparison (Pedrizzetti 2019; Vallelonga *et al.* 2021). The two components of the force, F_x and F_y, represent the force in the

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

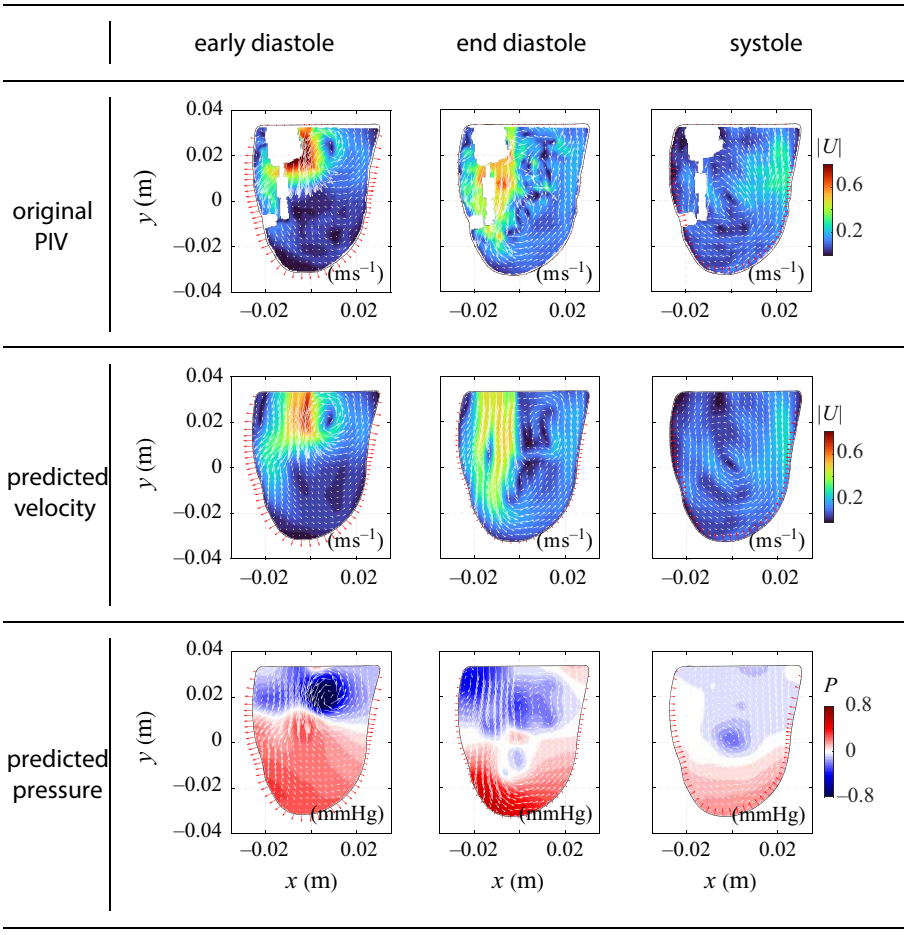


Figure 14. Instantaneous flow fields in the LV during different phases of the cardiac cycle: early diastole, end diastole and systole. From top to bottom, these figures present contour maps of the original PIV velocity magnitude, and the velocity magnitude and pressure fields predicted by the PINN-MB model. The predicted mesh is the same as the PIV mesh without increasing the spatial resolution. The unit of velocity is m s^{-1} and the pressure is converted to mmHg.

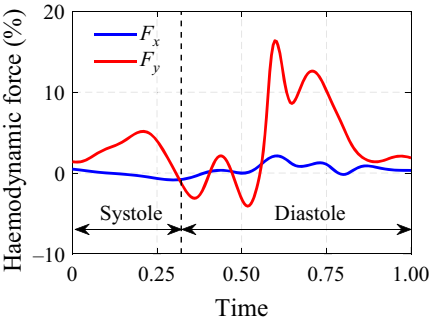


Figure 15. Haemodynamic force imposed on the blood during the cardiac cycle. The force is normalised by ρSg, where ρ is the blood density, S is the area enclosed by the LV boundary and g denotes the gravitational acceleration. The time is normalised with respect to the period. The two components of the force, F_x and F_y, represent the forces in the septal-lateral and apical-basal directions, respectively.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

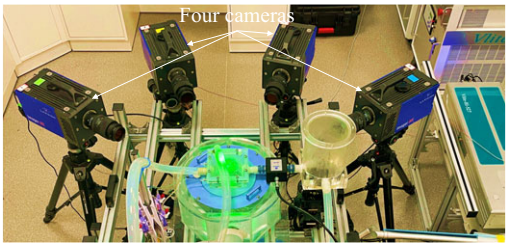


Figure 16. Set-up of the time-resolved tomographic PIV measurements.

septal-lateral and apical-basal directions, respectively. It is clear that the apical-basal force is much greater than the septal-lateral force. During systole, blood flow is accelerated towards the outflow direction because the haemodynamic force is directed from the apex to the base, as shown in figure 14. In early diastole, the force is negative and directed from the base to the apex, generating diastolic suction due to the relaxation of the LV. After this stage, the apical pressure gradually exceeds the basal pressure due to ventricular filling, which causes force to be directed from the apex to the base to decelerate the inflow. The haemodynamic force in diastole was greater than that in systole in our study; however, the magnitude of the force in these two phases was approximately the same as that reported in a previous numerical study by Pedrizzetti (2019). This may because the cardiac output in our experiment (3.6 l/min) was lower than that of a normal person (5.0 l/min). The clinical assessment of intraventricular flow is beyond the scope of this article, please refer to Pedrizzetti *et al.* (2014), Bermejo *et al.* (2015) and Vallelonga *et al.* (2021) for more information.

3.4. *Example 4: 3-D pressure in the left ventricle*

Thus far, the proposed method has been validated and investigated only for different types of 2-D flows. In this section, the feasibility of the proposed method for investigations of 3-D intraventricular flow is explored. Here as well, time-resolved tomographic PIV (Tomo-PIV) experiments were conducted based on the LV simulator described in § 3.3. Four high-speed cameras with 100 mm/f2.8 lenses were placed laterally around the nine-sided hydraulic chamber, as shown in figure 16. The view angle between two adjacent cameras was set to 36^{o}. A cylindrical laser with a diameter of 6 cm was expanded using a series of concave and convex lenses to illuminate all of the particles in the 3-D-printed silicone LV. Compared with the 2-D experiment, larger fluorescent Rhodamine-B coated particles with a diameter of 50 μm were used in the 3-D experiment due to lower energy of the expanded laser. The response frequency was $\sim$18 kHz, which means that the fluorescent particles can still follow the flow motion. The flow conditions were the same as those in the 2-D experiment. The raw particle images were preprocessed to remove the background and normalise the particle intensity. The particle concentration after preprocessing was approximately 0.045 particles per pixel (ppp). The 3-D intensity distribution was reconstructed using the multiplicative algebraic reconstruction technique proposed by Elsinga *et al.* (2006), based on the mapping function after self-calibration (Wieneke 2008). The reconstructed volume was 70 $\times$ 70 $\times$ 50 mm^3, containing 1400 $\times$ 1400 $\times$ 1000 voxels. The displacement was estimated using 3-D cross-correlation with a final interrogation volume of 48 $\times$ 48 $\times$ 48 voxels at 62.5 % overlap. This configuration yields 75 $\times$ 75 $\times$ 52 vectors with a spatial resolution of 2.4 mm and a vector pitch of 0.9 mm. Finally, all velocity fields were postprocessed using a divergence-free smoothing

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

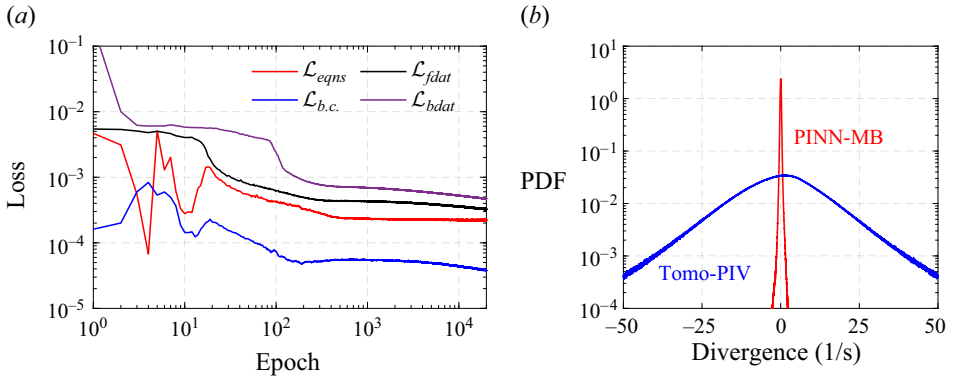


Figure 17. (*a*) Loss functions (equation loss $\mathcal{L}_{eqns}$, boundary condition loss $\mathcal{L}_{b.c.}$, flow data loss $\mathcal{L}_{fdat}$ and boundary data loss $\mathcal{L}_{bdat}$) as a function of epoch for the 3-D experimental case of LV. (*b*) PDFs of the divergence for the tomographic PIV and PINN-MB results.

algorithm to clearly visualise the flow structures (Wang *et al.* 2016*a*). The whole Tomo-PIV process was implemented using a validated in-house MATLAB code (Gao *et al.* 2012; Wang *et al.* 2016*b*; Zhu *et al.* 2017; Tu *et al.* 2022).

Three-dimensional three-component (3-D3C) measurements were obtained through Tomo-PIV; however, it is challenging to accurately and directly determine the wall of LV. In this work, our aim was to investigate the feasibility of the proposed method in the application to a 3-D moving boundary; therefore, an alternative method for obtaining a coarse boundary from the velocity is proposed. First, the mask data are identified as the region containing non-zero velocity (the velocity magnitude is larger than 0.01 m s^{-1} in practice) because tracer particles are distributed only in the LV. Second, the boundary of the LV is extracted as an isosurface of the mask data. Third, the boundary points are matched by assuming that the vertices move along the wall-normal direction. Therefore, the walls at different times are described by a series of Lagrangian points with the same number and connectivity. In the present case, there are 4001 spatial points on the wall, and the time sequence is the same as that of the velocity data. The configuration of the PINN-MB model is as follows: the boundary network consists of nine hidden layers and 64 neurons per hidden layer, and the flow network contains 15 hidden layers and 200 neurons per hidden layer. This configuration yields 598 423 trainable parameters. The location of the LV wall at the first time step is chosen as the initial input X_b^0 for the boundary network. Approximately 6.6×10^6 velocity vectors are randomly selected from three hundred Tomo-PIV fields with a time separation of 0.01 s (100 Hz) to serve as supervised data for the flow network, 2.0×10^6 boundary data are randomly selected as the supervised data for the boundary network, 6.5×10^6 equation points are used to collect the residuals of the 3-D incompressible N–S equations and 2.0×10^6 points are used for the boundary conditions. The PINN-MB model was trained for 20 000 epochs with 30 iterations per epoch.

Figure 17(*a*) presents the corresponding loss functions as a function of the training epoch. The training losses gradually converge to stable values after 1000 epochs. The final equation loss $\mathcal{L}_{eqns}$, boundary condition loss $\mathcal{L}_{b.c.}$, flow data loss $\mathcal{L}_{fdat}$ and boundary data loss $\mathcal{L}_{bdat}$ are approximately 2×10^{-4}, 4×10^{-5}, 3×10^{-4} and 5×10^{-4}. Figure 17(*b*) shows the probability distribution function (PDF) of the divergence before and after optimisation. The maximum PDF of the PINN-MB model is approximately two

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

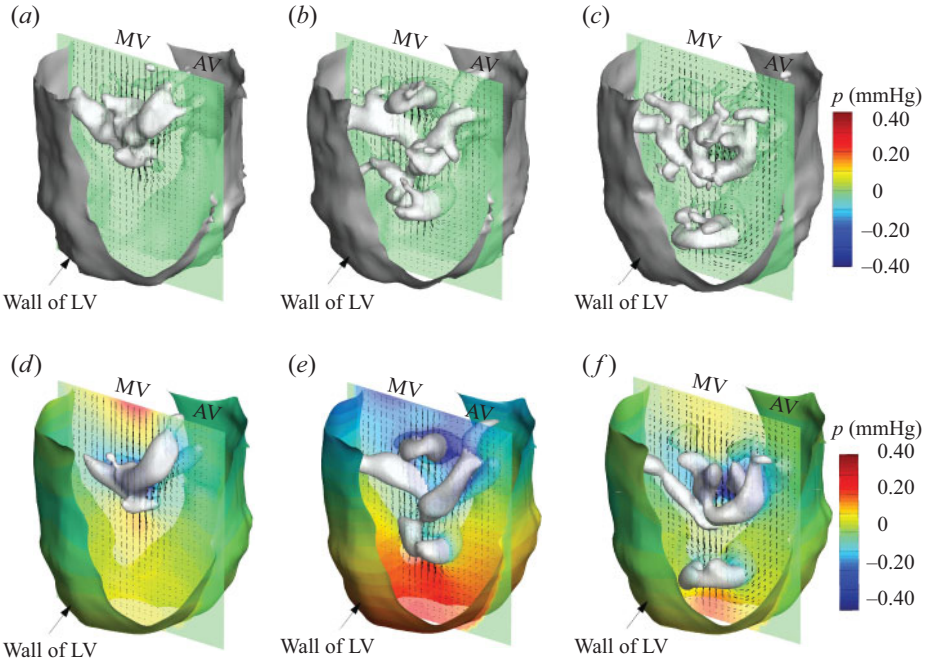


Figure 18. Instantaneous left ventricular flow fields of (*a, b, c*) the raw tomographic PIV data and (*e, f, g*) the present machine learning-based results. The surface pressure can be directly obtained from the flow network. The vortex structures are visualised using the Q criterion with $Q = 500$. From left to right, the time separation is 0.05 s.

orders of magnitude greater than that of the raw data, implying that the reconstructed velocities better satisfy the physical constraints. The instantaneous flow fields of the raw tomographic PIV (panels *a–c*) and the PINN-MB results (panels *d–f*) are displayed in figure 18, from left to right, the time separation is 0.05 s. The vortex structures are visualised as isosurfaces with $Q = 500$, where Q denotes the vortex identification criterion estimated as the second invariant of the velocity gradient tensor and the contour map denotes the pressure distribution. Note that there are no pressure data in the raw tomographic PIV data. The large-scale vortex structures are clearly approximately the same in the raw data and the optimised results. Some small structures, which may be contamination by measurement noise, disappear for the correction of the N–S equations. The pressure fields of greatest concern are represented by the colourful contour map in figures 18(*d*)–18(*f*). The pressure is converted to mmHg for clinical applicability. At the first time instant when the mitral vortex enters the LV, two high-pressure regions exist at the front and back of the vortex structure. A low-pressure region is observed, which is consistent with the vortex core, to balance the centrifugal force and pressure force. The pressure gradient between the apex and base reaches a maximum when the vortex travels towards the middle of the LV. At this time, a vortex ring separates from the main vortex and continues to move ahead until it impinges on the apex of the LV, as shown in figure 18(*f*). The pressure acting on the wall of the LV can be directly obtained from the flow network; therefore, it is convenient to compute the haemodynamic force to describe cardiac function. From this 3-D example, we can conclude that it is feasible to reconstruct the pressure field in a scenario with a moving boundary using the proposed method.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

4. Discussion and conclusions

In the present work, a machine-learning-based method, named PINN-MB, is proposed to reconstruct pressure fields with time-dependent moving boundaries. PINN-MB consists of two neural networks: a boundary network to track the motion of boundaries and a physics-informed neural network to approximate the flow fields (Raissi *et al.* 2019*a*,*b*, 2020). The former is constrained by a series of Lagrangian points over the boundaries and provides boundary conditions for the flow network. The latter is constrained by the flow data, boundary conditions and physical laws. These two networks are synchronously optimised by minimising the total loss function, which consists of the residual of the N–S equations, the loss between the measured and predicted flow data, the loss between the measured and predicted boundary data, and the loss of the boundary conditions. This novel framework can simultaneously optimise the flow fields and boundaries. The flow fields are optimised according to the N–S equations and the boundaries are optimised according to the boundary conditions. Moverover, it is also possible to impose geometric constraints of the boundaries by adding an additional loss term. For example, the boundary can be forced to be a circle by adding a constraint in the loss function. Zhu *et al.* (2024) extended the PINN to obtain incompressible flow with moving boundaries by directly imposing Dirichlet boundary conditions at the moving interfaces. However, their loss function considers only rigid motion with translation and rotation, and cannot address the motion of non-rigid bodies. The PINN-MB framework is more generally applicable to any flow problem with moving boundaries.

To effectively train this network, we propose a new strategy given in (2.5) to update the dynamic weights corresponding to the loss terms. The dynamic weights are estimated using both the loss statistics and loss gradients to balance the loss terms. This implies that the dynamic weight increases to a large value only for loss terms with a large value and small gradient. We compared the performance of our method with that of the method proposed by Jin *et al.* (2021), in which the dynamic weights are updated based on the loss gradients only. Particularly for a large-scale neural network, the solution is sensitive to the initial trainable variables and always fails to return an accurate prediction if we use the dynamic weight proposed by Jin *et al.* (2021). However, our method reasonably generates larger dynamic weights for the supervised data to train the network. In our opinion, the correct approach for the network is to converge based on the data rather than the equations or boundary conditions at the beginning of training. Therefore, our method for updating the dynamic weights is more robust and efficient than the previous approaches.

This developed machine-learning-based pressure reconstruction method for scenarios with moving boundaries was validated with flows over an oscillating cylinder and swimming fish. In addition, the proposed method was applied to determine the pressure according to experimental intraventricular flow data. Conventional methods, including the pressure Poisson equation and pressure gradient integration, face difficulties in the reconstruction of the pressure from experimental data with moving boundaries because of the complex boundary conditions and contaminated velocities and boundaries by measurement noise (Fan *et al.* 2023). In the case of an oscillating cylinder, the accuracy and robustness of the proposed method are demonstrated by inputting sparse low-resolution velocity fields with Gaussian random noise. The relative ℓ_2 norm pressure error is less than 4 % across the whole test range. In the more complex case of swimming fish, the kinematic error of the fish body predicted by PINN-MB is approximately 0.1 %, and the pressure error is approximately 14 %. This performance is similar to the results reported by Calicchia *et al.* (2023), in which the pressure predicted by PINN has a relative error of 10–18 %, and Queen 2.0 (a pressure integration method developed by Dabiri *et al.* 2014) obtains a larger relative error of 13–25 %. However, PINN-MB can predict

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

the pressure and boundaries at any instant. The proposed method was also applied to determine the pressure from 2-D and 3-D PIV velocity measurements of the LV. The results indicate that the proposed method can correctly track non-rigid motion. Due to the advantages of PINN, the measurement noise and gaps in the PIV data can be repaired by the proposed method through considering physical constraints. Using surface pressure data, we can also calculate the haemodynamic force, which may be an important clinical index of cardiac function (Pedrizzetti *et al.* 2014; Vallelonga *et al.* 2021). The application of the tomographic PIV on the LV demonstrated that the proposed method can be used for 3-D scenarios with moving boundaries. However, the proposed machine-learning-based method does not provide a built-in uncertainty quantification for noisy data, which may restrict its applications. To the best of our knowledge, the probabilistic neural network proposed by Maulik *et al.* (2020) and the Bayesian PINN proposed by Liu *et al.* (2021) can be used to quantify the uncertainty. In applying of the Bayesian PINN, the weights and biases are assumed to be Gaussian distributions with zero means. However, the Bayesian inference is often expensive for a deeper network. The probabilistic neural network can directly predict uncertainty by maximising the average likelihood of training data.

The accuracy and robustness of PINN-MB for pressure reconstruction with a moving boundary were confirmed based on our validation and experimental applications. We expect that the proposed network will be highly beneficial for addressing flow problems with moving boundaries, including pressure and high-resolution flow field reconstruction according to the measurement data. We also find that the probabilistic neural network is an appropriate approach to estimate the uncertainty in PINN-MB if we assume a Gaussian form for the predictive probability distribution (Maulik *et al.* 2020). Additionally, the choice of hyperparameters in machine learning depends strongly on the user's experience. These problems deserve further study to improve the performance of machine-learning-based methods.

Acknowledgments This work was supported by the National Natural Science Foundation of China (NSFC) Basic Science Center Program for 'Multiscale Problems in Nonlinear Mechanics' Grant No. 11988102, NSFC Grant Nos. 12072348, 92252203, 12425207, and the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB0620102.

Declaration of interests The authors report no conflicts of interest.

Appendix A. On the dynamic weights

Despite encouraging results, proper training of the PINN is still difficult due to the presence of multiple competing loss terms (Wang *et al.* 2021). According to the backpropagated gradient analysis proposed by Wang *et al.* (2021), the interplay among all of the loss terms in (2.4) should be appropriately balanced by using dynamic weights. In the present work, we investigated three different approaches to set these weights. The first approach is to set these weights to constant values, as in (A1). However, poor results are always obtained because the loss functions do not converge to the correct values. Therefore, we do not present the results obtained using constant weights.

$$\hat{\alpha}^{k+1} = 1, \quad \hat{\beta}^{k+1} = 1, \quad \hat{\gamma}^{k+1} = 1. \tag{A1}$$

The second approach is to update the dynamic weights using the method proposed by Jin *et al.* (2021):

$$\hat{\alpha}^{k+1} = \frac{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{eqns}|}}{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{b.c.}|}}, \quad \hat{\beta}^{k+1} = \frac{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{eqns}|}}{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{fdat}|}}, \quad \hat{\gamma}^{k+1} = \frac{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{eqns}|}}{\overline{|\nabla_{\boldsymbol{\Theta}} \mathscr{L}_{bdat}|}}. \tag{A2}$$

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

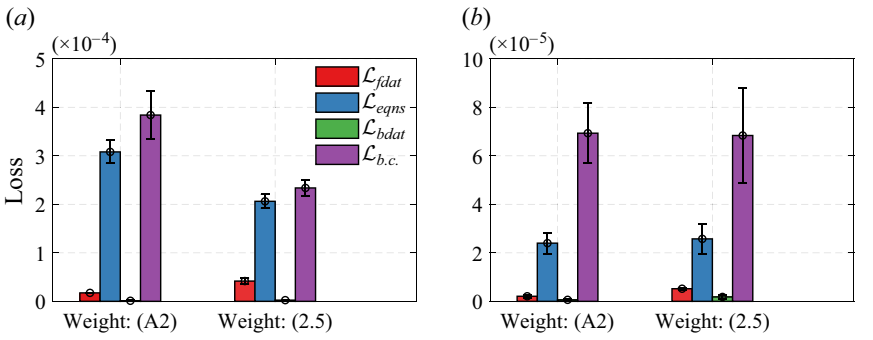


Figure 19. Losses for the numerical benchmark of a 2-D flow over an oscillating cylinder after 5000 epochs with different dynamic weight calculation approaches. The colour bars and error bars represent the mean and standard deviation of the loss obtained from three independent runs. (*a*) In the small neural network, the boundary network contains nine hidden layers and 32 neurons per hidden layer, and the flow network contains 13 hidden layers and 80 neurons per hidden layer, yielding a total of 86 987 trainable parameters. (*b*) In the large neural network, the boundary network contains nine hidden layers and 32 neurons per hidden layer, and the flow network contains 13 hidden layers and 160 neurons per hidden layer, yielding a total of 318 907 trainable parameters. No measurement noise is considered in this test case.

These equations imply that the loss terms are balanced using gradient statistics, which can be effectively computed by using automatic differentiation. However, in practice, (A2) always overestimates the weight of the boundary condition loss term (α). Moreover, the final loss was sensitive to the initial trainable variables and difficult to converge, especially in the PIV experiment of LV when the dynamic weights are calculated via (A2). In the paper by Jin *et al.* (2021), the weights were manually tuned to obtain satisfying results for the case of turbulent channel flow. To avoid manual adjustments, a third approach is proposed in the present work, as shown in (2.5). The major difference is that both the gradient and loss are balanced in our method. In this appendix, a comparative assessment based on the 2-D flow over a moving cylinder and 2-D flow in the LV is given.

Figure 19 displays the loss terms for the case of 2-D flow over an oscillating cylinder after 5000 epochs. The colour bars and error bars represent the mean value and standard deviation, respectively, obtained from three independent runs. For a small neural network with 86 987 trainable variables (figure19*a*), the velocity loss $\mathscr{L}_{fdat}$ obtained using the proposed dynamic weights converges to approximately 0.4×10^{-4}, which is slightly larger than the 0.2×10^{-4} obtained via (A2). The residual of equations $\mathscr{L}_{eqns}$ decreases from 3.0×10^{-4} to 2.0×10^{-4}. For a large network with 318 907 trainable variables, these two dynamic weight strategies present similar performance because of the relatively simple flow (figure 19*b*).

A more complex case of 2-D PIV flow fields at the LVOT plane is investigated to evaluate the performance of these two dynamic weight calculation strategies. Similar to figure 19, the losses after 5000 epochs are displayed in figure 20. The velocity loss $\mathscr{L}_{fdat}$ is much greater than the equation loss $\mathscr{L}_{eqns}$ for the dynamic weights of Jin *et al.* (2021) for both the small-scale and large-scale networks. Moreover, the error bars are also very large, particularly in the large network. This result implies that the dynamic weights proposed by Jin *et al.* (2021) are sensitive to the initial values especially for the large-scale network. However, the proposed dynamic weight calculation method converges to correct flow fields, as both the mean value and standard deviation are small. To directly compare the performance of the different dynamic weight calculation methods, figure 21 displays the dynamic weights obtained by different strategies in the experimental case of the LV. With the dynamic weights proposed by Jin *et al.* (2021), the dynamic weight of the boundary

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

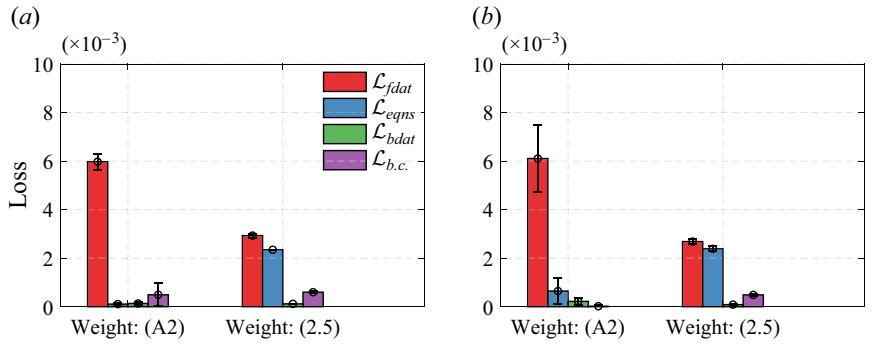


Figure 20. Losses for the experimental case of a 2-D flow in the LV after 5000 epochs for different dynamic weight calculation approaches. The colour bars and error bars represent the mean and standard deviation of the loss obtained by three independent runs. (*a*) In the small neural network, the boundary network contains nine hidden layers and 48 neurons per hidden layer, and the flow network contains 13 hidden layers and 128 neurons per hidden layer, yielding a total of 218 171 trainable parameters. (*b*) In the large neural network, the boundary network contains nine hidden layers and 48 neurons per hidden layer, and the flow network contains 13 hidden layers and 200 neurons per hidden layer, yielding a total of 502 931 trainable parameters.

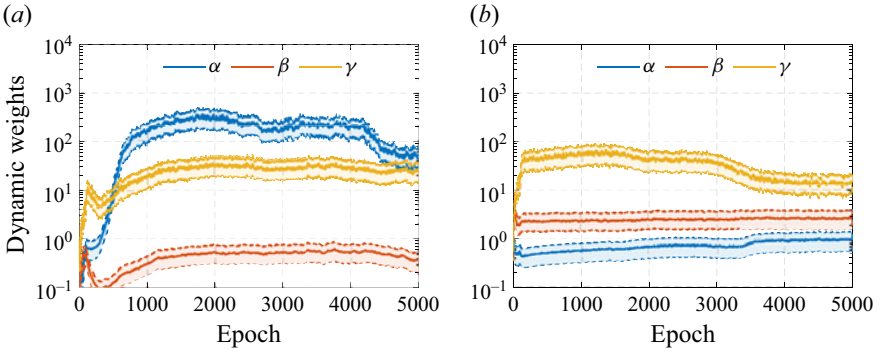


Figure 21. Dynamic weights as a function of epochs for the experimental case of 2-D flow in the left ventricle. (*a*) Dynamic weights proposed by Jin *et al.* (2021); (*b*) the present method. The curves are obtained from a large-scale neural network in which the boundary network contains nine hidden layers and 48 neurons per hidden layer, and the flow network contains 13 hidden layers and 200 neurons per hidden layer. The shadow regions represent the standard deviation of the dynamic weights from three independent runs.

condition α increases to approximately 200 after 800 epochs (the maximum weight is limited to 1000 in the algorithm for stability); this large weight causes the boundary condition loss term to converge to a very low value of $O(10^{-6})$. However, the velocity data loss term is large because of the small weight $\beta < 1$. With the proposed approach, the boundary data loss term has the maximum weight, followed by the flow data loss term. The boundary condition term has the smallest weight. It makes sense for the network to converge according to the data rather than the equation or boundary condition terms at the beginning of training.

Appendix B. Spatial discretisation of the equation points

In this appendix, the influence of the spatial discretisation of the equation points on the performance of the proposed method is assessed in the case of flow over an oscillating cylinder, as described in § 3.1. The boundary network contains nine hidden layers and

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

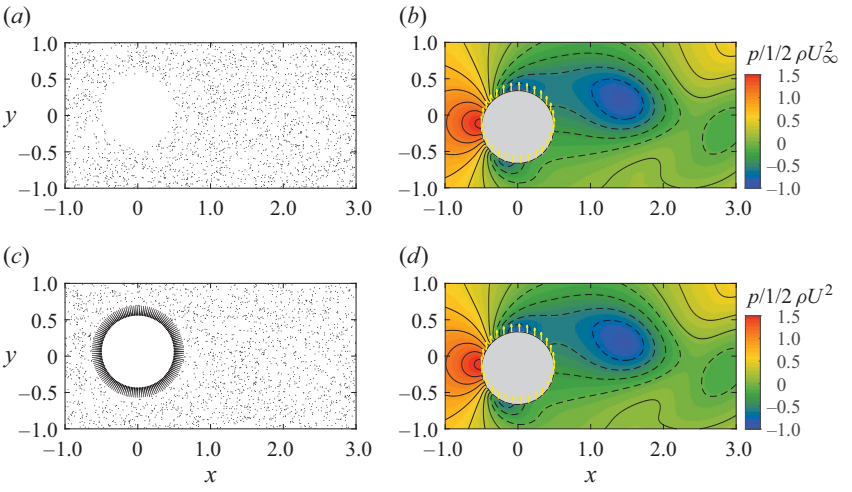


Figure 22. (*a, b*) Uniformly distributed equation points and corresponding predicted pressure field. (*c, d*) Near-wall grid refined equation points and corresponding predicted pressure field. For the training data, the vector space $1/R$ is set to $1/40$, and the noise level is set to 0 %. The learned data were predicted over a mesh with an interval of $1/R = 1/100$. Therefore, the original CFD data were interpolated on the predicted mesh to compute the corresponding error. The yellow arrows indicate the motion of the cylinder. The dashed black lines correspond to negative contour levels from -1.5 to -0.1 with an interval of 0.2, and the solid black lines correspond to positive contour levels from 0.1 to 1.5 with an interval of 0.2.

32 neurons per hidden layer, and the flow network contains 13 hidden layers and 128 neurons per hidden layer, yielding a total of 207 707 trainable parameters. Note that the scattered velocity data are collected by interpolating the flow fields on a uniform grid to mimic regular PIV data. Two strategies are adopted to generate equation points. The first approach involves a random uniform distribution over the computational domain, as shown in figure 22(*a*), and the second approach involves a near-wall grid refinement, which better captures the boundary layer, as shown in figure 22(*c*). Each strategy is trained for 4000 epochs, and the pressure fields are predicted at the same locations and times.

Figure 22 shows a comparison of the pressures predicted using the uniform and wall-refined equation points at time $t = 15$. A visual inspection of the results suggests that there are no differences between the contours. For a quantitative comparison, we present the relative ℓ_2 norm error of the pressure as a function of time in figure 23. The error fluctuation in the non-uniform case is larger than that in the uniform case, and the global error in the uniform case is slightly smaller than that in the near-wall grid refinement case. In both cases, larger errors are observed at the beginning and end of the tested time due to the fewer physical constraints. In the present work, therefore, a more convenient random uniform distribution is adopted to generate equation points.

Appendix C. On the coordinate system

PINN-MB is an extension of the PINN used to reconstruct the pressure with moving boundaries, where the inputs are space-time coordinates $(t, \mathbf{X})$, and the outputs are the velocity $\mathbf{U}$ and pressure p. After training with the constraints of the supervised data (flow data and boundary data) and physical laws, a mapping relationship between the input and output is established with the neural network. This implies that the PINN is a network-based data assimilation method that combines the physical laws and observed data to seek the optimal states. Therefore, the physical model and the observed data should

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

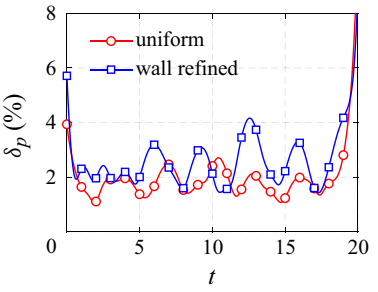


Figure 23. Time series of the relative ℓ_2 norm error of the pressure for the uniform and wall-refined strategies in the case of flow over an oscillating cylinder.

have the same coordinate system. The neural network can be trained under Cartesian or cylindrical coordinates, whereas the model trained with the Cartesian coordinates cannot be directly applied to other coordinates since the inputs and outputs depend on the boundary conditions, initial conditions and flow properties defined on a specific coordinate system. If the coordinate is rotated, the neural network must be retrained to adapt to the new coordinate, or the data should be preprocessed by coordinate transformation to satisfy the requirement of the model. However, for Reynolds averaged turbulence modelling using a deep neural network, Ling *et al.* (2016) introduced an invariant tensor basis to embed Galilean invariance into the predicted anisotropy tensor. This is because the predicted anisotropy tensor and its eigenvalues do not depend on the the coordinate system's orientation for identical flows.

Further assessment of the PINN performance in cylindrical coordinates was carried out using the data of two axisymmetric co-rotating vortex rings (Verzicco *et al.* 1997). Note that PINN has been used to estimate the pressure field from the axisymmetric synthetic jet (Fan *et al.* 2023). The numerical set-up of the co-rotating vortex rings is the same as that in the paper by Verzicco *et al.* (1997). The Reynolds number $Re = \Gamma/\nu$ is equal to 2500, in which Γ and ν denote the circulation of the ring and the kinematic viscosity, respectively. The initial configuration consists of two identical vortex rings with Gaussian vorticity distributions, and the initial spacing between the rings is equal to the ring radius R, the ratio core-to-ring radius $\epsilon/R = 0.3$. The evolution of the initial flow field follows the axisymmetric N–S equations:

$$\begin{aligned}
&\frac{\partial u_z}{\partial t} + u_r\frac{\partial u_z}{\partial r} + u_z\frac{\partial u_z}{\partial z} = -\frac{\partial p}{\partial z} + \frac{1}{Re}\left(\frac{\partial^2 u_z}{\partial r^2} + \frac{\partial^2 u_z}{\partial z^2} + \frac{1}{r}\frac{\partial u_z}{\partial r}\right),\\
&\frac{\partial u_r}{\partial t} + u_r\frac{\partial u_r}{\partial r} + u_z\frac{\partial u_r}{\partial z} = -\frac{\partial p}{\partial r} + \frac{1}{Re}\left(\frac{\partial^2 u_r}{\partial r^2} + \frac{\partial^2 u_r}{\partial z^2} + \frac{1}{r}\frac{\partial u_r}{\partial r} - \frac{u_r}{r^2}\right),\\
&\frac{\partial u_r}{\partial r} + \frac{\partial u_z}{\partial z} + \frac{u_r}{r} = 0.
\end{aligned} \tag{C1}$$

Here, the subscripts 'r' and 'z' indicate the radial and axial coordinates, respectively. After the reference velocity and pressure fields are solved via OpenFoam, the scattered velocity data for training are collected by interpolating the flow fields on a uniform grid to mimic typical PIV data. The vector space is set to $R/40$, and only two hundred flow fields with a time interval of $100\Delta t$ (Δt is the time step of the numerical simulation) are used for PINN training, in which the axisymmetric N–S equation (C1) is used.

The network consists of 11 hidden layers and 128 neurons per hidden layer, and contains 166 030 trainable parameters. The network was trained for 6000 epochs with 30 iterations

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

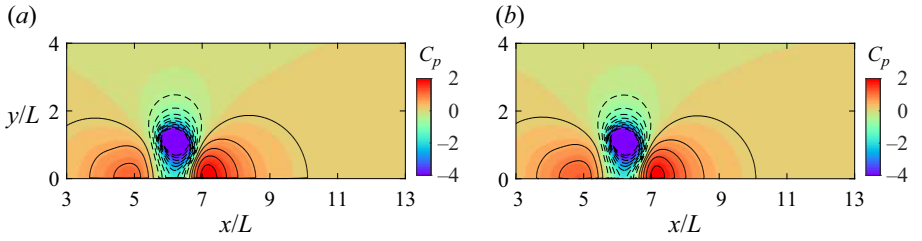


Figure 24. A representative snapshot of the pressure given by (*a*) CFD and (*b*) PINN for the case of two axisymmetric co-rotating vortex rings. The dashed black lines represent negative contour levels from −4 to −0.2 with an interval of 0.4, and the solid black lines represent positive contour levels from 0.2 to 4 with an interval of 0.4. Note that for comparative assessment, the spatially averaged pressure is subtracted from both the exacted and predicted data for comparative assessment. The predicted mesh is the same as that of the training data without noise.

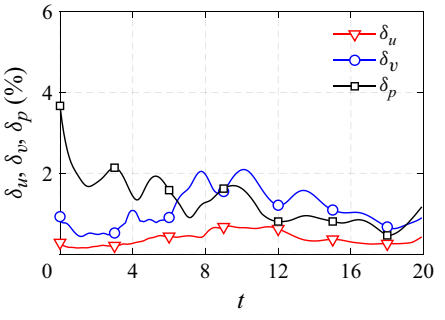


Figure 25. Relative ℓ_2 norm errors for the velocity u, v and pressure p as a function of time.

per epoch, and the batch size was set to 10 000. This configuration is only used to prove that PINN can be applied to cylindrical coordinates. According to Verzicco *et al.* (1997), the velocity is normalised with the speed of translation of the steady vortex ring in isolation, $U_0 = \Gamma/4\pi R[ln(8R/\epsilon) - 1/4]$, which is equal to 0.2414. Therefore, the pressure coefficient C_p presented in this work is scaled by $0.5\rho U_0^2$. Figure 24 compares the pressure reconstructed using PINN with the CFD result at time instant $t = 0.5$, where these two vortex rings have merged into only one ring. A visual inspection indicates that the predicted pressure is the same as the reference pressure. The relative ℓ_2 norm errors for the velocity u, v and pressure p are also presented in figure 25. The maximum error of pressure is less than 4 %.

Appendix D. On the number of training data

In the present study, a machine-learning-based method is proposed to reconstruct the pressure fields from measured velocity fields with moving boundaries. Because the spatial resolution of PIV/PTV is limited by the seeding concentration of tracing particles, the measured velocities are sparsely distributed in space. In this example, we investigated the effect of the amount of training data on the performance of pressure reconstruction based on the 2-D flow over an oscillating cylinder (§ 3.1) and swimming fish (§ 3.2). In our numerical simulations, 35 940 and 76 982 CFD points are used in the test region for the oscillating cylinder and fish swimming, respectively. It is reasonable that the simulation of the swimming fish requires more CFD points to capture the complex flow comprehensively. We randomly selected training data points from the CFD points to

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

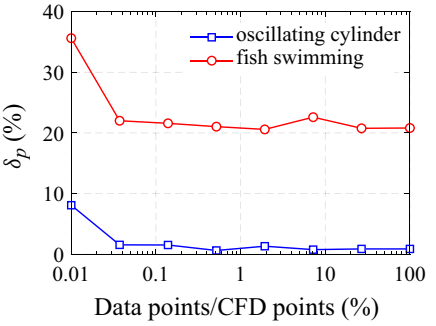


Figure 26. Relative ℓ_2 norm error of the pressure as a function of the ratio of the number of data points to the number of CFD points. For the flow over oscillating cylinder, the boundary network contains nine hidden layers and 32 neurons per hidden layer, and the flow network contains 13 hidden layers and 128 neurons per hidden layer, yielding a total of 207 707 trainable parameters. For the flow of fish swimming, the boundary network contains nine hidden layers and 48 neurons per hidden layer, and the flow network contains 13 hidden layers and 150 neurons per hidden layer, yielding a total of 291 981 trainable parameters.

test the performance of PINN-MB. Figure 26 shows the relative ℓ_2 norm error of the pressure as a function of the ratio of the number of data points to the number of CFD points. We find that the pressure errors converge to a stable value when the ratio of the data points is greater than 0.1 %, which means that the supervised data needed by the proposed PINN-MB are only 0.1 % of the CFD points. Note that the pressure error of fish is slightly larger than the error in figure 10 because we used a smaller neural network to reduce the training time. It is difficult to predict the lowest amount of training data for complex and high-dimensional flows. Generally, the low-dimensional flow dynamic system required less points to be represented than the high-dimensional system. Therefore, the sparse and limited training data can be used to reconstruct the low-dimensional manifold, as the pressure error decreases rapidly when the data ratio less than 0.1 % in figure 26. This phenomenon inspires us that modal analysis such as proper orthogonal decomposition (Berkooz *et al.* 1993), dynamic mode decomposition (Schmid 2010) and machine-learning-based autoencoder (Murata *et al.* 2020), extracting the low-dimensional flow features, can be used to reduce the requirement of training data and the network size. Fukami & Taira (2023) compressed extreme aerodynamics to only three variables with a lift-augmented autoencoder, enabling the sparse reconstruction, dynamical modelling and control of extremely unsteady gusty flows. A data-driven framework was also developed to reduce a dynamical system onto a low-dimensional manifold and time evolution on that manifold (Linot & Graham 2020). If the amount of training data continues to increase, as shown in figure 26, the reduction of error is extremely limited. This is because the inherent feature of neural networks, spectral bias, prefers to fit the low-dimensional models rather than high-dimensional models (Rahaman *et al.* 2019), which has been discussed in § 3.1.

Appendix E. Comparison with the PINN

In § 2.2, we have discussed the advantages of PINN-MB by comparing it with the original PINN and the numerical method. Both PINN and numerical method can solve the pressure if the boundary condition, including the position and velocity as known data, is given. However, the pressure obtained by these two methods may be contaminated by measurement errors. To visually show the difference between PINN and PINN-MB, the pressure over an oscillating cylinder is reconstructed using these two machine-learning-based methods. Noise with $\sigma_n = 8\,\%$ was added to both the velocity data and boundary

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

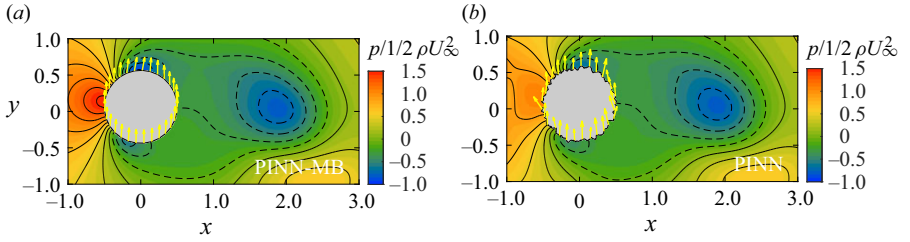


Figure 27. A representative snapshot of the pressure reconstructed by (*a*) PINN-MB and(*b*) PINN (*b*). The PINN is unable to optimise noisy boundary conditions, therefore, the pressure near the cylinder is incorrect.

locations. For PINN, the boundary velocity was estimated usinga second-order polynomial fit over seven points, and the simple finite difference was not enough to suppress the noise. The boundary condition and flow data were fed into the PINN as supervised data. For PINN-MB, the input of the boundary network is only the noisy boundary position, and the input of the flow network is the flow data. The boundary velocity is estimated by the automatic differentiation of the boundary network and should be equal to the velocity predicted by the flow network. Figure 27 shows an example snapshot of the pressure reconstructed by PINN-MB (panel *a*) and PINN (panel *b*). The network of PINN is the same as that of PINN-MB without the boundary network. The surface pressure over the oscillating cylinder reconstructed by PINN-MB is more accurate than that reconstructed by PINN because PINN-MB can optimise the noisy boundary condition.

References

Abe, S., Okamoto, K. & Madarame, H. 2004 The development of PIV-PSP hybrid system using pressure sensitive particles. *Meas. Sci. Technol.* **15** (6), 1153–1157.

Adrian, R.J. 2005 Twenty years of particle image velocimetry. *Exp. Fluids* **39** (2), 159–169.

Arzani, A., Wang, J.X. & D'Souza, R.M. 2021 Uncovering near-wall blood flow from sparse data with physics-informed neural networks. *Phys. Fluids* **33** (7), 071905.

Berkooz, G., Holmes, P. & Lumley, J.L. 1993 The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid Mech.* **25** (1), 539–575.

Bermejo, J., Martínez-Legazpi, P. & del Álamo, J.C. 2015 The clinical assessment of intraventricular flows. *Annu. Rev. Fluid Mech.* **47** (1), 315–342.

Brindise, M.C., Busse, M.M. & Vlachos, P.P. 2018 Density- and viscosity-matched Newtonian and non-Newtonian blood-analog solutions with PDMS refractive index. *Exp. Fluids* **59** (11), 173.

Brunton, S.L., Noack, B.R. & Koumoutsakos, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52** (1), 477–508.

Brunton, S.L., Proctor, J.L. & Kutz, J.N. 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113** (15), 3932–3937.

Cai, S.Z., Mao, Z.P., Wang, Z.C., Yin, M.L. & Karniadakis, G.E. 2021*a* Physics-informed neural networks (PINNs) for fluid mechanics: a review. *Acta Mechanica Sin.* **37** (12), 1727–1738.

Cai, S.Z., Wang, Z.C., Fuest, F., Jeon, Y.J., Gray, C. & Karniadakis, G.E. 2021*b* Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks. *J. Fluid Mech.* **915**, A102.

Cai, S.Z., Zhou, S.C., Xu, C. & Gao, Q. 2019 Dense motion estimation of particle images via a convolutional neural network. *Exp. Fluids* **60** (4), 73.

Cai, Z.M., Liu, Y., Chen, T. & Liu, T.S. 2020 Variational method for determining pressure from velocity in two dimensions. *Exp. Fluids* **61** (5), 118.

Calicchia, M.A., Mittal, R., Seo, J.H. & Ni, R. 2023 Reconstructing the pressure field around swimming fish using a physics-informed neural network. *J. Expl Biol.* **226** (8), jeb244983.

Callaham, J.L., Loiseau, J.C. & Brunton, S.L. 2023 Multiscale model reduction for incompressible flows. *J. Fluid Mech.* **973**, A3.

Chen, H. & Dasi, L.P. 2023 An in-vitro study of the flow past a transcatheter aortic valve using time-resolved 3D particle tracking. *Ann. Biomed. Engng* **51** (7), 1449–1460.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

CHURCHILL, V., MANNS, S., CHEN, Z. & XIU, D.B. 2023 Robust modeling of unknown dynamical systems via ensemble averaged learning. *J. Comput. Phys.* **474**, 111842.

DABIRI, J.O., BOSE, S., GEMMELL, B.J., COLIN, S.P. & COSTELLO, J.H. 2014 An algorithm to estimate unsteady and quasi-steady pressure fields from velocity field measurements. *J. Expl Biol.* **217** (3), 331–336.

DURAISAMY, K., IACCARINO, G. & XIAO, H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51** (1), 357–377.

ELDAN, R. & SHAMIR, O. 2016 The power of depth for feedforward neural networks. In 29th Annual Conference on Learning Theory (ed. V. Feldman, A. Rakhlin & O. Shamir), Proceedings of Machine Learning Research, Columbia University, New York, New York, USA, 23–26 Jun, vol. 49, pp. 907–940. PMLR.

ELSINGA, G.E., SCARANO, F., WIENEKE, B. & VAN OUDHEUSDEN, B.W. 2006 Tomographic particle image velocimetry. *Exp. Fluids* **41** (6), 933–947.

FAN, D., XU, Y., WANG, H.P. & WANG, J.J. 2023 Comparative assessment for pressure field reconstruction based on physics-informed neural network. *Phys. Fluids* **35** (7), 077116.

FERRARI, S. & ROSSI, L. 2008 Particle tracking velocimetry and accelerometry (PTVA) measurements applied to quasi-two-dimensional multi-scale flows. *Exp. Fluids* **44** (6), 873–886.

FUKAMI, K., FUKAGATA, K. & TAIRA, K. 2019 Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* **870**, 106–120.

FUKAMI, K., FUKAGATA, K. & TAIRA, K. 2021 Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *J. Fluid Mech.* **909**, A9.

FUKAMI, K. & TAIRA, K. 2023 Grasping extreme aerodynamics on a low-dimensional manifold. *Nat. Commun.* **14** (1), 6480.

GAO, Q., PAN, S.W., WANG, H.P., WEI, R.J. & WANG, J.J. 2021 Particle reconstruction of volumetric particle image velocimetry with the strategy of machine learning. *Adv. Aerodyn.* **3** (1), 28.

GAO, Q., WANG, H.P. & SHEN, G.X. 2013 Review on development of volumetric particle image velocimetry. *Chinese Sci. Bull.* **58** (36), 4541–4556.

GAO, Q., WANG, H.P. & WANG, J.J. 2012 A single camera volumetric particle image velocimetry and its application. *Sci. Chin. Technol. Sci.* **55** (9), 2501–2510.

VAN GENT, P.L., *et al.* 2017 Comparative assessment of pressure field reconstructions from particle image velocimetry measurements and Lagrangian particle tracking. *Exp. Fluids* **58** (4), 33.

GOODEN, S.C.M., HATOUM, H., ZHANG, W., BOUDOULAS, K.D. & DASI, L.P. 2022 Multiple MitraClips: the balancing act between pressure gradient and regurgitation. *J. Thorac. Cardiovasc. Surg.* **163** (4), 1319–1327.e1.

JAGTAP, A.D., KAWAGUCHI, K. & KARNIADAKIS, G.E. 2020 Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *J. Comput. Phys.* **404**, 109136.

JIANG, X.Y., KERSWELL, R.R., LEFAUVE, A., LINDEN, P.F. & ZHU, L. 2024 New insights into experimental stratified flows obtained through physics-informed neural networks. *J. Fluid Mech.* **981**, R1.

JIN, X.W., CAI, S.Z., LI, H. & KARNIADAKIS, G.E. 2021 NSFnets (Navier–Stokes flow nets): physics-informed neural networks for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **426**, 109951.

KARNIADAKIS, G.E., KEVREKIDIS, I.G., LU, L., PERDIKARIS, P., WANG, S.F. & YANG, L. 2021 Physics-informed machine learning. *Nat. Rev. Phys.* **3** (6), 422–440.

DE KAT, R. & VAN OUDHEUSDEN, B.W. 2012 Instantaneous planar pressure determination from PIV in turbulent flow. *Exp. Fluids* **52** (5), 1089–1106.

KAT, R.D. & GANAPATHISUBRAMANI, B. 2012 Pressure from particle image velocimetry for convective flows: a Taylor's hypothesis approach. *Meas. Sci. Technol.* **24** (2), 024002.

KEANE, R.D. & ADRIAN, R.J. 1992 Theory of cross-correlation analysis of PIV images. *Appl. Sci. Res.* **49** (3), 191–215.

KIM, H., KIM, J., WON, S. & LEE, C. 2021 Unsupervised deep learning for super-resolution reconstruction of turbulence. *J. Fluid Mech.* **910**, A29.

LAPINSKAS, T., PEDRIZZETTI, G., STOIBER, L., DÜNGEN, H.D., EDELMANN, F., PIESKE, B. & KELLE, S. 2019 The intraventricular hemodynamic forces estimated using routine cmr cine images: a new marker of the failing heart. *Cardiovasc. Imaging* **12** (2), 377–379.

LIAO, Z.M., ZHAO, Z.Y., CHEN, L.B., WAN, Z.H., LIU, N.S. & LU, X.Y. 2023 Reduced-order variational mode decomposition to reveal transient and non-stationary dynamics in fluid flows. *J. Fluid Mech.* **966**, A7.

LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016 Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

LINOT, A.J. & GRAHAM, M.D. 2020 Deep learning to discover and predict dynamics on an inertial manifold. *Phys. Rev. E* **101** (6), 062209.
LIU, X.F. & KATZ, J. 2006 Instantaneous pressure and material acceleration measurements using a four-exposure PIV system. *Exp. Fluids* **41** (2), 227–240.
LIU, Y., MENG, X.H. & KARNIADAKIS, G.E. 2021 B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *J. Comput. Phys.* **425**, 109913.
LIU, Y., WANG, G. & YE, Z.Y. 2018 Dynamic mode extrapolation to improve the efficiency of dual time stepping method. *J. Comput. Phys.* **352**, 190–212.
LIU, Y., WANG, H.P., WANG, S.Z. & HE, G.W. 2023 A cache-efficient reordering method for unstructured meshes with applications to wall-resolved large-eddy simulations. *J. Comput. Phys.* **480**, 112009.
LORENZONI, V., TUINSTRA, M. & SCARANO, F. 2012 On the use of time-resolved particle image velocimetry for the investigation of rod-airfoil aeroacoustics. *J. Sound Vib.* **331** (23), 5012–5027.
LUI, H.F.S. & WOLF, W.R. 2019 Construction of reduced-order models for fluid flows using deep feedforward neural networks. *J. Fluid Mech.* **872**, 963–994.
MAULIK, R., FUKAMI, K., RAMACHANDRA, N., FUKAGATA, K. & TAIRA, K. 2020 Probabilistic neural networks for fluid flow surrogate modeling and data recovery. *Phys. Rev. Fluids* **5** (10), 104401.
MELE, D., SMARRAZZO, V., PEDRIZZETTI, G., CAPASSO, F., PEPE, M., SEVERINO, S., LUISI, G.A., MAGLIONE, M. & FERRARI, R. 2019 Intracardiac flow analysis: techniques and potential clinical applications. *J. Am. Soc. Echocardiog.* **32** (3), 319–332.
MILDENHALL, B., SRINIVASAN, P.P., TANCIK, M., BARRON, J.T., RAMAMOORTHI, R. & NG, R. 2020 NeRF: representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV. 2020*, (ed. A. VEDALDI, H. BISCHOF, T. BROX & J.-M. FRAHM), pp. 405–421. Springer International Publishing.
MURATA, T., FUKAMI, K. & FUKAGATA, K. 2020 Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *J. Fluid Mech.* **882**, 15.
VAN OUDHEUSDEN, B.W. 2013 PIV-based pressure measurement. *Meas. Sci. Technol.* **24** (3), 032001.
PEDRIZZETTI, G. 2019 On the computation of hemodynamic forces in the heart chambers. *J. Biomech.* **95**, 109323.
PEDRIZZETTI, G. & DOMENICHINI, F. 2015 Left ventricular fluid mechanics: the long way from theoretical models to clinical applications. *Ann. Biomed. Engng* **43** (1), 26–40.
PEDRIZZETTI, G., LA CANNA, G., ALFIERI, O. & TONTI, G. 2014 The vortex—an early predictor of cardiovascular outcome? *Nat. Rev. Cardiol.* **11** (9), 545–553.
PEDRIZZETTI, G. & PERKTOLD, K. 2003 *Cardiovascular Fluid Mechanics*. Springer.
PIRNIA, A., MCCLURE, J., PETERSON, S.D., HELENBROOK, B.T. & ERATH, B.D. 2020 Estimating pressure fields from planar velocity data around immersed bodies; a finite element approach. *Exp. Fluids* **61** (2), 55.
RABAULT, J., KUCHTA, M., JENSEN, A., REGLADE, 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.
RAFFEL, M., WILLERT, C.E., SCARANO, F., KÄHLER, C.J., WERELEY, S.T. & KOMPENHANS, J. 2018 *Particle Image Velocimetry: A Practical Guide*. Springer.
RAHAMAN, N., BARATIN, A., ARPIT, D., DRAXLER, F., LIN, M., HAMPRECHT, F., BENGIO, Y. & COURVILLE, A. 2019 On the spectral bias of neural networks, Proceedings of the 36th International Conference on Machine Learning, (ed. K. CHAUDHURI & R. SALAKHUTDINOV), vol. **97**, pp. 5301–5310. PMLR.
RAISSI, M., PERDIKARIS, P. & KARNIADAKIS, G.E. 2019*a* Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707.
RAISSI, M., WANG, Z.C., KARNIADAKIS, G.E. & TRIANTAFYLLOU, M.S. 2019*b* Deep learning of vortex-induced vibrations. *J. Fluid Mech.* **861**, 119–137.
RAISSI, M., YAZDANI, A. & KARNIADAKIS, G.E. 2020 Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* **367** (6481), 1026–1030.
SAAID, H., *et al.* 2019 Tomographic PIV in a model of the left ventricle: 3D flow past biological and mechanical heart valves. *J. Biomech.* **90**, 40–49.
SCARANO, F. 2002 Iterative image deformation methods in PIV. *Meas. Sci. Technol.* **13** (1), R1–R19.
SCARANO, F. 2013 Tomographic PIV: principles and practice. *Meas. Sci. Technol.* **24** (1), 012001.
SCHANZ, D., GESEMANN, S. & SCHRDER, A. 2016 Shake-the-box: Lagrangian particle tracking at high particle image densities. *Exp. Fluids* **57** (5), 1–27.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

SCHMID, P.J. 2010 Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28.

SCHRÖDER, A. & SCHANZ, D. 2023 3D Lagrangian particle tracking in fluid mechanics. *Annu. Rev. Fluid Mech.* **55** (1), 511–540.

SHAO, S., MALLERY, K., KUMAR, S.S. & HONG, J. 2020 Machine learning holography for 3D particle field imaging. *Opt. Express* **28** (3), 2987–2999.

STOLL, V.M., HESS, A.T., RODGERS, C.T., BISSELL, M.M., DYVERFELDT, P., EBBERS, T., MYERSON, S.G., CARLHÄLL, C.J. & NEUBAUER, S. 2019 Left ventricular flow analysis. *Circ. Cardiovasc. Imaging* **12** (5), e008130.

TANCIK, M., SRINIVASAN, P.P., MILDENHALL, B., FRIDOVICH-KEIL, S., RAGHAVAN, N., SINGHAL, U., RAMAMOORTHI, R., BARRON, J.T. & NG, R. 2020 *Fourier features let networks learn high frequency functions in low dimensional domains*, Curran Associates Inc., Vancouver, BC, Canada.

THANDIACKAL, R. & LAUDER, G.V. 2020 How zebrafish turn: analysis of pressure force dynamics and mechanical work. *J. Expl Biol.* **223** (16), jeb223230.

TING, S.C. & YANG, J.T. 2009 Extracting energetically dominant flow features in a complicated fish wake using singular-value decomposition. *Phys. Fluids* **21** (4), 041901.

TRIANTAFYLLOU, M.S., TRIANTAFYLLOU, G.S. & YUE, D.K.P. 2000 Hydrodynamics of fishlike swimming. *Annu. Rev. Fluid Mech.* **32** (1), 33–53.

TU, H., WANG, F.J., WANG, H.P., GAO, Q. & WEI, R.J. 2022 Experimental study on wake flows of a live fish with time-resolved tomographic PIV and pressure reconstruction. *Exp. Fluids* **63** (1), 25.

VALLELONGA, F., AIRALE, L., TONTI, G., ARGULIAN, E., MILAN, A., NARULA, J. & PEDRIZZETTI, G. 2021 Introduction to hemodynamic forces analysis: moving into the new frontier of cardiac deformation analysis. *J. Am. Heart Assoc.* **10** (24), e023417.

VERZICCO, R., IAFRATI, A., RICCARDI, G. & FATICA, M. 1997 Analysis of the sound generated by the pairing of two axisymmetric co-rotating vortex rings. *J. Sound Vib.* **200** (3), 347–358.

VIDELER, J.J. & HESS, F. 1984 Fast continuous swimming of two pelagic predators, Saithe (*Pollachius virens*) and Mackerel (*Scomber scombrus*): a kinematic analysis. *J. Expl Biol.* **109** (1), 209–228.

WANG, C.Y., GAO, Q., WANG, H.P., WEI, R.J., LI, T. & WANG, J.J. 2016*a* Divergence-free smoothing for volumetric PIV data. *Exp. Fluids* **57** (1), 1–23.

WANG, C.Y., GAO, Q., WEI, R.J., LI, T. & WANG, J.J. 2017*a* Spectral decomposition-based fast pressure integration algorithm. *Exp. Fluids* **58** (7), 84.

WANG, H.P., GAO, Q., WEI, R.J. & WANG, J.J. 2016*b* Intensity-enhanced MART for tomographic PIV. *Exp. Fluids* **57** (5), 87.

WANG, H.P., HE, G.W. & WANG, S.Z. 2020*a* Globally optimized cross-correlation for particle image velocimetry. *Exp. Fluids* **61** (11), 228.

WANG, H.P., LIU, Y. & WANG, S.Z. 2022*a* Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network. *Phys. Fluids* **34** (1), 017116.

WANG, H.P., YANG, Z.X., LI, B.L. & WANG, S.Z. 2020*b* Predicting the near-wall velocity of wall turbulence using a neural network for particle image velocimetry. *Phys. Fluids* **32** (11), 115105.

WANG, S.F., TENG, Y.J. & PERDIKARIS, P. 2021 Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.* **43** (5), A3055–A3081.

WANG, S.F., YU, X.L. & PERDIKARIS, P. 2022*b* When and why PINNs fail to train: A neural tangent kernel perspective. *J. Comput. Phys.* **449**, 110768.

WANG, S.Z., VANELLA, M. & BALARAS, E. 2019 A hydrodynamic stress model for simulating turbulence/particle interactions with immersed boundary methods. *J. Comput. Phys.* **382**, 240–263.

WANG, Z.Y., GAO, Q., PAN, C., FENG, L.H. & WANG, J.J. 2017*b* Imaginary particle tracking accelerometry based on time-resolved velocity fields. *Exp. Fluids* **58** (9), 113.

WANG, Z.Y., GAO, Q., WANG, C.Y., WEI, R.J. & WANG, J.J. 2016*c* An irrotation correction on pressure gradient and orthogonal-path integration for PIV-based pressure reconstruction. *Exp. Fluids* **57** (6), 104.

WESTERWEEL, J., ELSINGA, G.E. & ADRIAN, R.J. 2013 Particle image velocimetry for complex and turbulent flows. *Annu. Rev. Fluid Mech.* **45**, 409–436.

WIENEKE, B. 2008 Volume self-calibration for 3D particle image velocimetry. *Exp. Fluids* **45** (4), 549–556.

XU, H., ZHANG, W. & WANG, Y. 2021 Explore missing flow dynamics by physics-informed deep learning: the parameterized governing systems. *Phys. Fluids* **33** (9), 095116.

XU, Z.Y., ZHANG, X.L., WANG, S.Z. & HE, G.W. 2022 Artificial neural network based response surface for data-driven dimensional analysis. *J. Comput. Phys.* **459**, 111145.

YU, M., HUANG, W.X. & XU, C.X. 2019 Data-driven construction of a reduced-order model for supersonic boundary layer transition. *J. Fluid Mech.* **874**, 1096–1114.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press

ZHANG, X.L., XIAO, H., LUO, X.D. & HE, G.W. 2022 Ensemble Kalman method for learning turbulence models from indirect observation data. *J. Fluid Mech.* **949**, A26.

ZHAO, B.D., HE, M.M. & WANG, J.W. 2023 Data-driven discovery of the governing equation of granular flow in the homogeneous cooling state using sparse regression. *Phys. Fluids* **35** (1), 013315.

ZHOU, Z.D., HE, G.W. & YANG, X.L. 2021*a* Wall model based on neural networks for LES of turbulent flows over periodic hills. *Phys. Rev. Fluids* **6** (5), 054610.

ZHOU, Z.T., WANG, H.P., WANG, S.Z. & HE, G.W. 2021*b* Lighthill stress flux model for ffowcs Williams-Hawkings integrals in frequency domain. *AIAA J.* **59** (11), 4809–4814.

ZHU, H.Y., WANG, C.Y., WANG, H.P. & WANG, J.J. 2017 Tomographic PIV investigation on 3D wake structures for flow over a wall-mounted short cylinder. *J. Fluid Mech.* **831**, 743–778.

ZHU, Y.Z., KONG, W.Z., DENG, J. & BIAN, X. 2024 Physics-informed neural networks for incompressible flows with moving boundaries. *Phys. Fluids* **36** (1), 013617.

https://doi.org/10.1017/jfm.2025.91 Published online by Cambridge University Press