

A data-driven computational framework for non-intrusive reduced-order modelling of turbulent flows passing around bridge piers

Chuanhua Zhu ^{a,b}, Dunhui Xiao ^{c,e}, Jinlong Fu ^{d,b,*}, Yuntian Feng ^b, Rui Fu ^b, Jinsheng Wang ^f

^a School of Environment and Energy Engineering, Anhui Jianzhu University, Hefei, 230601, China

^b Zienkiewicz Institute for Modelling, Data and AI, Faculty of Science and Engineering, Swansea University, Swansea, SA1 8EN, UK

^c School of Mathematical Sciences, Key Laboratory of Intelligent Computing and Applications (Ministry of Education), Tongji University, Shanghai 200092, China

^d Department of Mechanical Engineering, Faculty of Science and Engineering, Queen Mary University of London, London, E1 4NS, UK

^e State Key Laboratory of Disaster Reduction in Civil Engineering, Tongji University, Shanghai, 200092, China

^f School of Civil Engineering, University of Birmingham, Birmingham, B15 2TT, UK

ARTICLE INFO

Keywords:

Model order reduction
Unsteady/turbulent flow
Computational fluid dynamics
Stacked autoencoder
Dynamic mode decomposition
Non-intrusive

ABSTRACT

Repetitively conducting high-fidelity numerical simulations under varying conditions is often a crucial requirement in the optimisation design of offshore bridges and structures. Reduced-order modelling (ROM) provides an efficient approach to quickly and reliably obtain solutions by extracting low-dimensional representations from full-order numerical systems. This paper presents a novel data-driven computational framework for non-intrusive ROM of turbulent/unsteady flows passing around bridge piers, consisting of two interconnected components: the Stacked Autoencoder (SAE) and the Dynamic Mode Decomposition (DMD). The novelty lies in utilising SAE to achieve nonlinear dimensionality reduction by projecting the full-order dynamical system onto a low-dimensional latent space, followed by constructing reduced-order models through data-driven DMD to represent fluid dynamics in the latent feature space. This new SAE-DMD-based method is applied to develop reduced-order models for two unsteady flow problems, and it is also compared with classical DMD and high-fidelity numerical simulations in terms of modelling accuracy, forecasting efficiency and memory requirements. The results demonstrate that the proposed method can rapidly offer reliable predictions while significantly reducing memory usage and it exhibits excellent extrapolation capability by accurately preserving primary nonlinear characteristics of fluid dynamics. This new method shows potential to overcome computational challenges associated with high-resolution numerical modelling for complex large-scale flow problems.

1. Introduction

Cylindrical piers are the most commonly used substructures to provide vertical supports for bridges across rivers, lakes and seas. With flow passing bridge piers, current/wave induced forces (Miao et al., 2021; Chen and Jeng, 2022) have a strong and long-term impact on hydraulic/marine structures, which can cause structural tilt and even overturning of the bridges. In addition, current/wave-induced scour (Baghbadorani et al., 2018; Wu et al., 2023a) erodes the foundations of the soil and causes local holes around the bridge piers, which can seriously damage the bearing capacity of the foundation. These problems of fluid–structure interaction are closely related to the fluid flow characteristics near/around bridge piers (Aghaee and Hakimzadeh, 2010), which should be carefully considered in the design stage. Compared to experimental testing, numerical simulation (Khosronejad et al., 2012; Alemi et al., 2019) is an effective and cheaper way

to investigate relevant engineering problems, and the latter also allows flexible variation of system parameters for optimal design, such as pier geometry, flow rate, flow orientation and boundary conditions.

The fluid flow characteristics around circular piers are turbulent and complicated, and the scour process is also dynamic and unstable. Direct numerical simulation (DNS) of flow passing bridge piers using computational fluid dynamics (CFD) modelling can provide accurate solutions (Aghaee and Hakimzadeh, 2010; Zhu et al., 2019), but it is extremely compute intensive for turbulent flows with high Reynolds numbers, making it very inflexible and inefficient for engineering applications. Although other CFD approaches such as Reynolds-averaged Navier–Stokes equations (RANS) and large eddy simulation (LES) are more cost-effective than DNS (Jia et al., 2018; Heinz, 2020), repeatedly conducting high-fidelity simulation can bring an prohibitive computational burden, due to the iterative nature of bridge/structure

* Corresponding author at: Department of Mechanical Engineering, Faculty of Science and Engineering, Queen Mary University of London, London, E1 4NS, UK.

E-mail address: jinlong.fu@qmul.ac.uk (J. Fu).

optimisation design. Reduced order modelling (Noack et al., 2003; Fu et al., 2023a) can greatly decrease computational cost by deriving low-dimensional representations from full-order numerical systems, thus achieving fast (even real-time) and reliable solutions.

Reduced-order modelling (ROM) has been widely used in various applications, including fluid dynamics (Fu et al., 2023a; Srinil et al., 2009; Fu et al., 2020; Hammond and Sapsis, 2023), solid fracturing (Xiao et al., 2017b; Oliver et al., 2017), air pollution modelling (Fang et al., 2014; Xiao et al., 2019a), optimal control (Proctor et al., 2016), uncertainty quantification (Sashidhar and Kutz, 2022) and data assimilation (Hammond et al., 2019). Generally, ROM methods can be categorised into intrusive and non-intrusive types, depending on whether the source code for full-order numerical models is required or not (Benner et al., 2017; Xiao et al., 2019b). Intrusive ROM methods need to access this source code in order to project high-fidelity system operators onto the subspace formed by a number of basis functions (Quarteroni et al., 2015). Intrusive ROM can adequately maintain the underlying structure of the full-order numerical model, and the system-theoretic error can also be derived in a rigorous way. However, the level of access to the system operators often limits the usage of intrusive ROM methods (Benner et al., 2017). In addition, intrusive ROM methods also suffer from inefficiency and instability issues for strong nonlinear problems (Osth et al., 2014). Petrov–Galerkin methods (Collins et al., 2020; Xiao et al., 2013) and regularisation methods (Sabeghadam and Jafarpour, 2012) have been developed to deal with instability issues. To improve the efficiency of nonlinear model reduction, several methods have been presented, such as the empirical interpolation method (EIM) (Barrault et al., 2004), the discrete empirical interpolation method (DEIM) (Chaturantabut and Sorensen, 2010), the residual error version of DEIM (Xiao et al., 2014), and the least square Petrov–Galerkin method (Carlberg et al., 2011).

Recently, *non-intrusive reduced-order modelling* (NIROM) methods have become popular in various fields (Fang et al., 2009; Xiao et al., 2015, 2017a; Fu et al., 2023b). NIROM decouples the on-line stage from high-fidelity system operators and avoids the disadvantages of intrusive ROM methods. NIROM has a natural connection to artificial intelligence (Fu et al., 2023a) and takes full advantage of machine/deep learning in complex data analysis and hidden rule exploration to construct predictive models. Non-intrusive reduced-order models (Yu et al., 2019) are constructed by projecting the original full-order (high-dimensional) system onto a lower-dimensional latent space, where dimensionality reduction methods can be achieved by proper orthogonal decomposition (POD) (Star et al., 2021), balanced truncation (Iqbal et al., 2022), or the reduced-basis method (Benner et al., 2017; Lee and Carlberg, 2020). However, these model reduction methods restrict the dynamics to evolve in a linear subspace, which may bring accuracy issues for highly nonlinear problems (Lee and Carlberg, 2020; Rowley and Dawson, 2017). To address this linear subspace issue, nonlinear manifold methods based on autoencoder (Lee and Carlberg, 2020) have been developed for NIROM.

Autoencoder (Fu et al., 2021) is an unsupervised learning algorithm for dimensionality reduction or representation learning, and it has been becoming popular in recent ROM studies to project the high-dimensional numerical systems onto a nonlinear latent space. A NIROM method for fluid dynamics was developed by Wu et al. (2021), where convolutional autoencoder (CAE) and self-attention were integrated together. Kim et al. (2022) put forward a nonlinear manifold ROM method for advection-dominated problems, in which the physics-informed neural network was combined with autoencoders. Pawar et al. (2019) proposed a non-intrusive data-driven ROM framework based on the encoder–decoder approach and deep neural network for fluid flow. Lusch et al. (2018) designed a ROM framework using a modified autoencoder and Koopman operator to identify nonlinear coordinates. A data-driven NIROM method was proposed by Fu et al. (2023b), where the stacked auto-encoder was used to reduce the dimensionality of

full-order dynamical systems and self-attention was adopted to predict fluid dynamics. Fresca and Manzoni (2021) introduced a real-time simulation approach for parameter-dependent fluid flows using deep learning-based reduced order models, in which POD and convolutional autoencoder were combined together. Besides, Liu et al. (2022) presented a data-driven NIROM method in a similar way, where stacked autoencoder and long-short-term memory (LSTM) were used. Drakoulias et al. (2023) utilised LSTM together with CAE to construct reduced-order models for the dynamical systems relevant to fluid flows. Wei et al. (2024) proposed a three dimensional CAE-based method for flow field reconstruction, where CAE showed outperformance over POD in feature extraction.

In addition, *dynamic mode decomposition* (DMD) is another powerful tool for dimensionality reduction, which has been successfully used to extract spatial and temporal patterns from multi-dimensional time series of flow data (Schmid, 2010, 2022; Wu et al., 2023b). DMD possesses excellent prediction capability (Proctor et al., 2016) and has been used for data-driven predictive modelling in various disciplines. Kutz et al. (2016) developed a multi-resolution version of DMD, which is capable of separating complex dynamical systems into a hierarchy of multi-resolution components. Mohan et al. (2018) proposed a data-driven strategy to forecast the short-term electricity load via DMD. Li et al. (2022) proposed a novel DMD framework that can extract spatial-temporal features from non-uniformly sampled flow data. Wu et al. (2023b) proposed a deep-learning assisted non-intrusive reduced order model based on the compressed sensing-DMD, to achieve accurate prediction of high-dimensional flow dynamics from limited and sparse data. Diez et al. (2022) applied DMD to analyse and forecast the trajectories of ships, aiming to guide ship manoeuvring in waves. Streaming DMD was developed by Liew et al. (2022) to achieve real-time forecasts of wind energy variation in wind farms. Further, Haq et al. (2022) combined DMD and CAE to model the underlying dynamics of objects in video, and the developed method has been used for foreground extraction and video classification.

Based on the aforementioned research, it can be inferred that CAE is specifically tailored for processing grid-like data, such as 2D or 3D images, by utilising convolutional and pooling operations to capture spatial hierarchies in regular grids; whereas SAE is a type of neural network comprising multiple fully-connected layers of encoder and decoder to derive hierarchical representations of input data. LSTM, a variant of recurrent neural network, has been integrated with CAE to capture temporal dependencies in sequential data, while DMD is a model reduction technique used to analyse dominant spatial–temporal patterns in dynamical systems. The combination of SAE with DMD will provide interpretable and reduced-order representations of complex dynamical systems, enabling effective modelling of system behaviour and prediction of future states.

Inspired by the latest progress in ROM, autoencoder and DMD, this study seeks to develop a new data-driven computational framework for non-intrusive model order reduction of turbulent/unsteady flow dynamics. The basic idea is to use the stacked autoencoder (SAE) for nonlinear dimensionality reduction by projecting the full-order dynamical system onto a nonlinear latent subspace and then to construct a data-driven DMD model to represent the fluid dynamics by using the latent features. The combination of SAE and DMD for NIROM is a novel attempt to overcome the computational infeasibility of high-fidelity numerical simulations of turbulent/unsteady flows in engineering applications. The remainder of this paper is organised as follows: The governing equations and the full order numerical model corresponding to the investigated problem of turbulent/unsteady flows passing around bridge pillars are given in Section 2. Section 3 provides a description of SAE for nonlinear dimensionality reduction of snapshot data. In Section 4, the methodology of the data-driven framework based on SAE and DMD for non-intrusive ROM is explained in detail. Section 5 demonstrates the performance of the newly present SAE-DMD-based ROM method through two case studies. Finally, the strengths and weaknesses of the proposed method are discussed in Section 6, and the main contributions of this work are also summarised.

2. Governing equations and full-order numerical model

The evolution behaviour of a nonlinear dynamical system is governed by its underlying physical laws that are often represented in the form of partial differential equations (PDEs). Here, the fluid dynamics of turbulent/unsteady flows passing bridge piers can be described by the incompressible Navier–Stokes (N-S) equations, which mathematically express mass conservation and momentum balance for Newtonian fluids:

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}, \quad (2)$$

where $\mathbf{u} \equiv (u, v, w)^T$ denotes the flow velocity in x -, y - and z -directions, t denotes time, p is the fluid pressure, ρ denotes the constant fluid density, and ν represents the kinematic viscosity.

Analytically solving the above N-S equations is extremely difficult and usually impossible in most cases. The routine way is to approximate the solution of non-linear PDEs using numerical approaches (Blazek, 2015), such as the finite element method. The N-S equations are discretised and reformulated into a solvable system of algebraic equations that approximately characterise the nonlinear fluid dynamics. The full-order numerical model of the nonlinear dynamical system to represent the governing nonlinear PDEs can be generally written as:

$$D(\mathbf{u}) = \mathbf{0}, \quad (3)$$

$$\mathcal{M} \frac{\partial \mathbf{u}}{\partial t} + \mathcal{A}(\mathbf{u})\mathbf{u} + \mathcal{K}(\nu)\mathbf{u} + \mathcal{C}\mathbf{p} = \mathbf{0}, \quad (4)$$

where $D(\cdot)$ denotes the divergence operator, \mathcal{M} denotes the mass matrix, $\mathcal{A}(\cdot)$ denotes the solution-dependent streaming operator, $\mathcal{K}(\nu)$ denotes the matrix related to the remaining linear velocity terms, \mathbf{u} is a vector containing the velocity nodal values of all three velocity components, \mathcal{C} is a pressure gradient matrix, and \mathbf{p} is a vector containing the pressure nodal values.

To achieve accurate numerical approximations, the continuous physical fields are usually partitioned into fine meshes with large numbers of elements, and each element has a specific number of degrees of freedom. The full-order numerical model that embeds the N-S equations is thus in typically high dimension, which means the matrices and variable vectors in Eqs. (3) and (4) are huge. It can be an extremely compute-intensive process to solve a large-scale system of equations with a huge number of unknowns. Therefore, model order reduction techniques are of practical significance in seeking latent low-dimensional representations of the full-order numerical models, thereby efficiently resolving the primary fluid dynamics of the nonlinear system.

3. Stacked autoencoder for nonlinear dimensionality reduction

The full-order numerical model represents the nonlinear fluid dynamical system that maps space, time and system parameters (such as fluid viscosity, Reynolds number, initial and boundary conditions) onto physical field variables including velocity components \mathbf{u} and pressure p . The solutions of the physical field variables denoted by $\mathbf{q}(\mathbf{x}, t; \boldsymbol{\mu})$ vary with respect to space $\mathbf{x} \in \mathcal{X}$ and time $t \in \mathcal{T}$, and also depend on the parameters of the system $\boldsymbol{\mu} \in \mathcal{P}$, which can be denoted by the following function:

$$\mathbf{q}(\mathbf{x}, t; \boldsymbol{\mu}) : \mathcal{X} \times \mathcal{T} \times \mathcal{P} \rightarrow \mathbb{R}, \quad (5)$$

where \mathcal{X} , \mathcal{T} and \mathcal{P} denote the space, time and parameter domains, respectively.

This study aims to derive reduced-order approximate models of $\mathbf{q}(\mathbf{x}, t; \boldsymbol{\mu})$ from high-fidelity numerical solutions. Here, the stacked autoencoder (SAE) is adopted to conduct non-linear dimensionality reduction, where the high-fidelity dynamical system is projected onto a nonlinear latent subspace by preserving the primary dynamics.

3.1. Data preparation

Consider a set of high-fidelity numerical solutions of the physical field variable $\mathbf{q}(\cdot, t; \boldsymbol{\mu})$ at varying time points $t = \{t_1, t_2, \dots, t_i, \dots, t_n\} \in \mathcal{T}$, where $t_i = T_0 + i\Delta t$, and T_0 is the starting time point. Each numerical solution $\mathbf{q}(\cdot, t_i; \boldsymbol{\mu})$ is called a “snapshot”, and it can be reshaped into a column vector denoted by $q_i \in \mathbb{R}^{\mathcal{N}_x}$. A dataset \mathcal{Q} can be obtained by combining all column vectors together:

$$\mathcal{Q} = \{q_1, q_2, \dots, q_i, \dots, q_n\} \in \mathbb{R}^{\mathcal{N}_x \times n}, \quad (1 \leq i \leq n), \quad (6)$$

where \mathcal{N}_x is the dimension of the numerical discretisation of the spatial domain.

Before performing data dimensionality reduction, these snapshot data need to be standardised by removing the fixed component \bar{q} , called “particular solution” or “static correction”, and then a standardised dataset $\tilde{\mathcal{Q}}$ is obtained:

$$\begin{aligned} \tilde{\mathcal{Q}} &= \{q_1 - \bar{q}, q_2 - \bar{q}, \dots, q_i - \bar{q}, \dots, q_n - \bar{q}\} \\ &= \{\tilde{q}_1, \tilde{q}_2, \dots, \tilde{q}_i, \dots, \tilde{q}_n\} \in \mathbb{R}^{\mathcal{N}_x \times n}, \end{aligned} \quad (7)$$

where \bar{q} is computed as the mean of all column vectors reshaped from the snapshots (Hall et al., 2000), and it is considered irrelevant to the time dependence of fluid dynamics.

3.2. Nonlinear dimensionality reduction

High-fidelity snapshots usually have a high data dimensionality due to the fine mesh grid of full-order numerical simulation models. In this study, *autoencoder* (Fu et al., 2021; Hinton and Salakhutdinov, 2006) is used for data compression by learning non-linear latent representations from high-dimensional snapshots. An autoencoder is a type of unsupervised artificial neural network consisting of an encoder and a decoder, where the encoder compresses the input into lower-dimensional code and the decoder reconstructs the output from this latent space representation. The goal of autoencoder is to minimise the discrepancy between the initial input and the reconstructed output. Multiple basic autoencoders can be stacked to form a hierarchical deep model, where the output of each autoencoder serves as the input for the next one. This *stacked autoencoder* is able to learn more complex representations of the raw input, based on which hierarchical features can be compressed, organised and extracted. A stacked autoencoder (SAE) model composed of multiple layers of autoencoders is graphically illustrated in Fig. 1, in which the output of each encoder/decoder layer is fed into the successive encoder/decoder layer.

Encoder and decoder are the basic elements of a SAE model, and their mathematical explanations are provided in the following contents.

Let $\kappa^{(l)} = (\kappa_1^{(l)}, \kappa_2^{(l)}, \dots, \kappa_{m_l}^{(l)})^T$ denote the latent feature vector learned by the l th encoder layer in SAE, where m_l denotes the total number of hidden units in this encoder layer. The l th encoder $E_{\theta}^{(l)}(\cdot)$ compresses the input $\kappa^{(l-1)}$ into the hidden representation $\kappa^{(l)}$ using the following equation:

$$\kappa^{(l)} = E_{\theta}^{(l)}(\kappa^{(l-1)}) = f\left(\mathbf{W}_e^{(l)}\kappa^{(l-1)} + b_e^{(l)}\right), \quad (l = 1, 2, 3, \dots, L), \quad (8)$$

where the encoder $E_{\theta}^{(l)}(\cdot)$ is parameterised by $\theta = (\mathbf{W}_e^{(l)}, b_e^{(l)})$, $\mathbf{W}_e^{(l)}$ and $b_e^{(l)}$ denote the weight matrix and the bias of the encoder respectively, and L is the total number of layers in the stacked encoder. In this study, the original input $\kappa^{(0)}$ is the standardised snapshot data $\tilde{q} \in \mathbb{R}^{\mathcal{N}_x}$ (see Eq. (7)), and the final latent feature $\kappa^{(L)} \in \mathbb{R}^{m_L}$ is also called code. In the following text, the data compression result $\kappa^{(L)}$ will be denoted by κ for simplicity, and its length m_L will be denoted by m .

The l th decoder $D_{\hat{\theta}}^{(l)}(\cdot)$ transforms the hidden representation $\hat{\kappa}^{(l)}$ into the input space $\hat{\kappa}^{(l-1)}$ by:

$$\hat{\kappa}^{(l-1)} = D_{\hat{\theta}}^{(l)}(\hat{\kappa}^{(l)}) = f\left(\mathbf{W}_d^{(l)}\hat{\kappa}^{(l)} + b_d^{(l)}\right), \quad (l = 1, 2, 3, \dots, L), \quad (9)$$

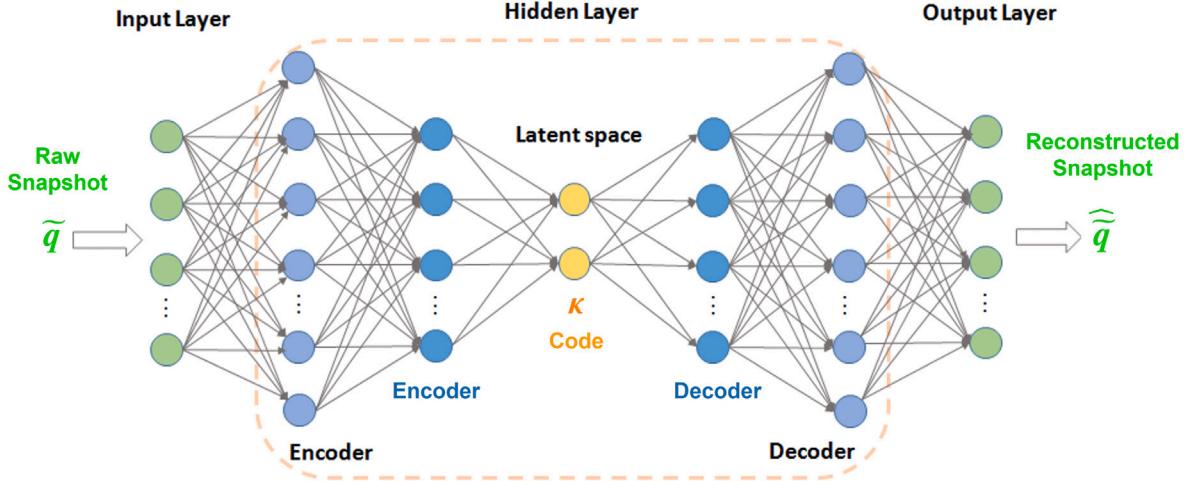


Fig. 1. The architecture of a typical stacked autoencoder (SAE) model.

where the decoder $D_{\hat{\theta}}^{(l)}(\cdot)$ is parameterised by $\hat{\theta} = (\mathbf{W}_d^{(l)}, b_d^{(l)})$, and $\mathbf{W}_d^{(l)}$ and $b_d^{(l)}$ denote the weight matrix and the bias of the decoder respectively. It should be noted that $\hat{\kappa}^{(L)} = \kappa^{(L)}$ here. Regarding the activation functions $f(\cdot)$, the ReLU function is adopted for the intermediate hidden layers, and the sigmoid function is used for the final output layer, which can be mathematically expressed as follows:

$$\text{ReLU: } f(x) = \max(0, x), \quad (10)$$

$$\text{Sigmoid: } f(x) = 1 / (1 + e^{-x}). \quad (11)$$

Training a SAE model is to minimise the discrepancy between the initial input $\tilde{\mathbf{q}}_i$ and the final output $\hat{\tilde{\mathbf{q}}}_i$ by optimally adjusting the parameters of the neural network, including weights \mathbf{W} and biases \mathbf{b} . And this discrepancy is represented by a cost function with three terms, given by:

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = \frac{1}{n} \sum_{i=1}^n L\left(\tilde{\mathbf{q}}_i, \hat{\tilde{\mathbf{q}}}_i\right) + \beta \sum_{j=1}^m KL\left(\psi \parallel \hat{\psi}_j\right) + \gamma \|\mathbf{W}\|_2^2, \quad (12)$$

where β and γ are constant coefficients. The first term of the loss function is the mean sum of squares error that accounts for the discrepancy between the input $\tilde{\mathbf{q}}_i$ and the output $\hat{\tilde{\mathbf{q}}}_i$ across the entire data set containing the information of the snapshots n :

$$\frac{1}{n} \sum_{i=1}^n L\left(\tilde{\mathbf{q}}_i, \hat{\tilde{\mathbf{q}}}_i\right) = \frac{1}{n} \sum_{i=1}^n \|\tilde{\mathbf{q}}_i - \hat{\tilde{\mathbf{q}}}_i\|^2 = \frac{1}{n} \sum_{i=1}^n \|\tilde{\mathbf{q}}_i - D_{\hat{\theta}}[E_{\hat{\theta}}(\tilde{\mathbf{q}}_i)]\|^2, \quad (13)$$

Encouraging sparsity of autoencoder is conducive to finding the latent representations of the input data and reducing the amount of redundant information. A sparsity cost can be added to the loss function for penalising activations of hidden layers, aiming to activate fewer hidden neurons while keeping data processing performance. Here, the Kullback–Leibler (KL) divergence is used to enforce a constraint on the sparsity of the output from hidden layers, as denoted by the second term in Eq. (12) that describes the sparsity cost (m is the number of neurons in the hidden layer, and the index j scans across all hidden neurons in the network):

$$KL\left(\psi \parallel \hat{\psi}_j\right) = \psi \log \frac{\psi}{\hat{\psi}_j} + (1 - \psi) \log \frac{1 - \psi}{1 - \hat{\psi}_j}, \quad (14)$$

where $\hat{\psi}_j$ is the average activation value of the hidden neuron j throughout the data set, and ψ is the desired value of average activation value. The third term in Eq. (12) defines the regularisation cost (also called the weight decay term), which tends to decrease the magnitude of the weights to prevent overfitting:

$$\|\mathbf{W}\|_2^2 = \text{tr}(\mathbf{W}^T \mathbf{W}), \quad (15)$$

In summary, the properly trained SAE model yields a stacked encoder (SE) function for non-linear dimensionality reduction of high-fidelity snapshots, given by:

$$SE\left(\tilde{\mathbf{q}}_i; \mathbf{W}_e, \mathbf{b}_e\right) : \tilde{\mathbf{q}}_i \in \mathbb{R}^{\mathcal{N}_x} \rightarrow \kappa_i \in \mathbb{R}^m, \quad (16)$$

where $SE(\cdot)$ denotes the SE function. A standardised snapshot data $\tilde{\mathbf{q}}_i$ is greatly compressed into a low-dimensional code κ_i through SE, where $\mathcal{N}_x \gg m$. Additionally, the stacked decoder (SD) function can recover the initial input $\hat{\tilde{\mathbf{q}}}_i$ from the code κ_i , given by:

$$SD\left(\kappa_i; \mathbf{W}_d, \mathbf{b}_d\right) : \kappa_i \in \mathbb{R}^m \rightarrow \hat{\tilde{\mathbf{q}}}_i \in \mathbb{R}^{\mathcal{N}_x}, \quad (17)$$

where $SD(\cdot)$ denotes the SD function.

4. Dynamic mode decomposition to construct predictive NIROM model

Dynamic mode decomposition (DMD) (Schmid, 2010, 2022) is commonly used as a dimensionality reduction tool, as it can decompose the dynamical system into a set of modes, each of which is associated with a fixed oscillation frequency. In recent studies (Mohan et al., 2018; Diez et al., 2022), DMD has been applied to build predictive models for time series data in different disciplines. In this study, DMD is combined with the stacked autoencoder (SAE) to construct predictive models for turbulent fluid dynamics in the reduced latent space. More details about the proposed SAE-DMD-based approach for model order reduction will be explained in the following contents.

Consider a code sequence \mathbf{K} generated from the pre-trained SAE model (as illustrated in Eq. (16)):

$$\mathbf{K} = \{\kappa_1, \kappa_2, \dots, \kappa_i, \dots, \kappa_n\} \in \mathbb{R}^{m \times n}, \quad (18)$$

where $\kappa_i \in \mathbb{R}^m$ represents the latent feature of the normalised snapshot data $\tilde{\mathbf{q}}_i \in \mathbb{R}^{\mathcal{N}_x}$. This code sequence is assumed to be related via a linear mapping that defines this linear dynamical system:

$$\kappa_{i+1} = \mathbf{A}\kappa_i, \quad (i = 1, 2, \dots, n-1), \quad (19)$$

The objective of DMD is to compute the eigen-decomposition by finding the best-fit linear operator \mathbf{A} , which can be obtained by solving the following minimisation problem:

$$\arg \min_{\mathbf{A}} \sum_{i=1}^{n-1} \|\kappa_{i+1} - \mathbf{A}\kappa_i\|_2. \quad (20)$$

To solve the above problem, the code sequence \mathbf{K} is divided and arranged into two data matrices:

$$\mathbf{K}_1 = \{\kappa_1, \kappa_2, \dots, \kappa_i, \dots, \kappa_{n-1}\} \in \mathbb{R}^{m \times (n-1)}, \quad (21)$$

$$\mathbf{K}_2 = \left\{ \boldsymbol{\kappa}_2, \boldsymbol{\kappa}_3, \dots, \boldsymbol{\kappa}_{i+1}, \dots, \boldsymbol{\kappa}_n \right\} \in \mathbb{R}^{m \times (n-1)}, \quad (22)$$

And then Eq. (20) can be converted to the following form:

$$\mathbf{K}_2 \approx \mathbf{A} \mathbf{K}_1, \quad (23)$$

from which the matrix \mathbf{A} can be computed as

$$\mathbf{A} = \mathbf{K}_2 \mathbf{K}_1^\dagger, \quad (24)$$

where $\mathbf{A} \in \mathbb{R}^{m \times m}$, and \mathbf{K}_1^\dagger is the Moore–Penrose pseudo-inverse of \mathbf{K}_1 .

In summary, the objective of the SAE-DMD algorithm is to perform the eigen-decomposition by finding the best-fit linear operator \mathbf{A} that relates the two code matrices \mathbf{K}_1 and \mathbf{K}_2 . In the classical DMD algorithm, the matrix \mathbf{A} is often not computed directly due to the high dimensionality of the snapshot data $\tilde{\mathbf{Q}}$. Singular value decomposition (SVD) is often adopted to efficiently compute the low-rank approximation of matrix \mathbf{A} in an efficient manner, where the computational cost is bounded by the rank r of the data. In this study, high-dimension snapshot data has been compressed to latent features (low-dimension codes) via SAE in advance. Taking into account the low dimensionality of the code matrices \mathbf{K}_1 and \mathbf{K}_2 , the *full rank approximation* of the matrix \mathbf{A} is used in SAE-DMD, in order to guarantee accuracy. The main computational procedure of stacked autoencoder-based dynamic mode decomposition (SAE-DMD) is summarised in Algorithm 1.

Algorithm 1: The main computational procedure of SAE-DMD

(i) Perform singular value decomposition (SVD) on the code matrix \mathbf{K}_1 :

$$\mathbf{K}_1 = \mathbf{U} \Sigma \mathbf{V}^T, \quad (25)$$

where \mathbf{K}_1 is a code sequence generated by the pre-trained SAE model, \mathbf{U} and \mathbf{V} denote the left and right singular vectors respectively, and Σ is a diagonal matrix consisting of singular values in decent order.

(ii) Implement a truncated SVD and compute the Koopman matrix $\tilde{\mathbf{B}}$:

$$\tilde{\mathbf{B}} = \mathbf{U}_r^T \mathbf{K}_2 \mathbf{V}_r \Sigma_r^{-1}, \quad (26)$$

where $\mathbf{U} \in \mathbb{C}^{m \times r}$, $\Sigma \in \mathbb{C}^{r \times r}$, $\mathbf{V} \in \mathbb{C}^{(n-1) \times r}$, and r is set to be equal to m (the code length) to compute the full-rank SVD approximation of \mathbf{K}_1 .

(iii) Perform eigenvalue decomposition on the Koopman matrix $\tilde{\mathbf{B}}$ to compute the eigenvalue vector ϕ :

$$\tilde{\mathbf{B}} = \mathbf{P} \phi \mathbf{P}^T, \quad (27)$$

where \mathbf{P} denotes eigenvectors, and ϕ denotes eigenvalues.

(iv) Compute the dynamic mode matrix Ψ and the best-fit linear operator \mathbf{A} :

$$\Psi = \mathbf{K}_2 \mathbf{V}_r \Sigma_r^{-1} \mathbf{P}, \quad (28)$$

$$\mathbf{A} = \Psi \phi \Psi^\dagger, \quad (29)$$

where \dagger denotes the operator of the Moore–Penrose pseudo-inverse.

With the stacked autoencoder-based dynamic model decomposition (SAE-DMD) method, a data-driven computational framework can be developed for non-intrusive reduced-order modelling (NIROM) of turbulent/unsteady flow dynamics, and its workflow is graphically described in Fig. 2. The high-dimensional snapshot data $\tilde{\mathbf{q}}_i$ is input into

the stacked encoder $\mathcal{SE}(\cdot)$, from which the low-dimensional code data $\boldsymbol{\kappa}_i$ is output via nonlinear dimensionality reduction, given by

$$\boldsymbol{\kappa}_i = \mathcal{SE}\left(\tilde{\mathbf{q}}_i; \mathbf{W}_e, \mathbf{b}_e\right), \quad (30)$$

A predictive DMD model $\mathcal{DMD}(\cdot)$ can then be constructed in the latent space:

$$\hat{\boldsymbol{\kappa}}_{i+1} = \mathcal{DMD}\left(\boldsymbol{\kappa}_i; \mathbf{A}\right), \quad (31)$$

The predicted code $\hat{\boldsymbol{\kappa}}_{i+1}$ will be later passed to a stacked decoder model $\mathcal{SD}(\cdot)$ to recover the approximation of predicted snapshot data, given by

$$\hat{\mathbf{q}}_{i+1} = \mathcal{SD}\left(\hat{\boldsymbol{\kappa}}_{i+1}; \mathbf{W}_d, \mathbf{b}_d\right) \quad (32)$$

The workflow of the new data-driven NIROM method based on SAE-DMD is summarised in the Algorithm 2.

Algorithm 2: The workflow of the data-driven NIROM method based on SAE-DMD

- (1) Generate snapshot data \mathbf{Q} by running the high-fidelity numerical simulation, and perform data standardisation on \mathbf{Q} yields $\tilde{\mathbf{Q}}$, as seen in Eq. (7);
- (2) Take each standardised snapshot $\tilde{\mathbf{q}}_i \in \tilde{\mathbf{Q}}$ as one observation, and use all of them as the inputs to train a SAE model for nonlinear dimensionality reduction, as explained by Eq. (8)–(15);
- (3) Calculate the latent space code $\boldsymbol{\kappa}_i$ of each standardised snapshot $\tilde{\mathbf{q}}_i$ using the stacked encoder model $\mathcal{SE}(\cdot)$, as explained by Eq. (16);
- (4) Perform dynamic mode decomposition (DMD) on the code data \mathbf{K} , and compute the best-fit linear operator \mathbf{A} , as explained by Eq. (18)–(29);
- (5) Make online prediction using the SAE-DMD model, as explained by Eq. (30)–(32):

while $t_i \leq T$ **do**

Step (i): Given the latent space code $\boldsymbol{\kappa}_i$ at the current time step t_i , calculate the approximation of the code $\hat{\boldsymbol{\kappa}}_{i+1}$ at the next time step t_{i+1} using the predictive DMD model $\mathcal{DMD}(\cdot)$:
 $\hat{\boldsymbol{\kappa}}_{i+1} = \mathcal{DMD}(\boldsymbol{\kappa}_i; \mathbf{A})$ or $\hat{\boldsymbol{\kappa}}_{i+1} = \mathbf{A} \boldsymbol{\kappa}_i$,
where $t_i = T_0 + i \Delta t$;

Step (ii): Project the predicted latent space codes $\hat{\boldsymbol{\kappa}}_{i+1}$ back into the original full-order space to obtain the predictions of physical quantities $\hat{\mathbf{q}}_{i+1}$ using the stacked decoder $\mathcal{SD}(\cdot)$:
 $\hat{\mathbf{q}}_{i+1} = \mathcal{SD}\left(\hat{\boldsymbol{\kappa}}_{i+1}; \mathbf{W}_d, \mathbf{b}_d\right)$.

5. Numerical results

To verify the effectiveness of the proposed SAE-DMD-based ROM approach, it is applied to construct non-intrusive reduced-order models for two typical engineering problems, namely, turbulent flow passing around a single bridge pillar and unsteady flow passing around multiple bridge pillars. The proposed SAE-DMD method is also compared with the classical DMD method and the high-fidelity numerical simulation, in terms of computational accuracy and efficiency. The root-mean-square error (RMSE) and correlation coefficient (CC) are used as the metrics to quantify the errors of reduced-order solutions. The RMSE measures the difference between the full-order solution $X(t)$ and the reduced-order solution $\hat{X}(t)$, which is mathematically expressed as

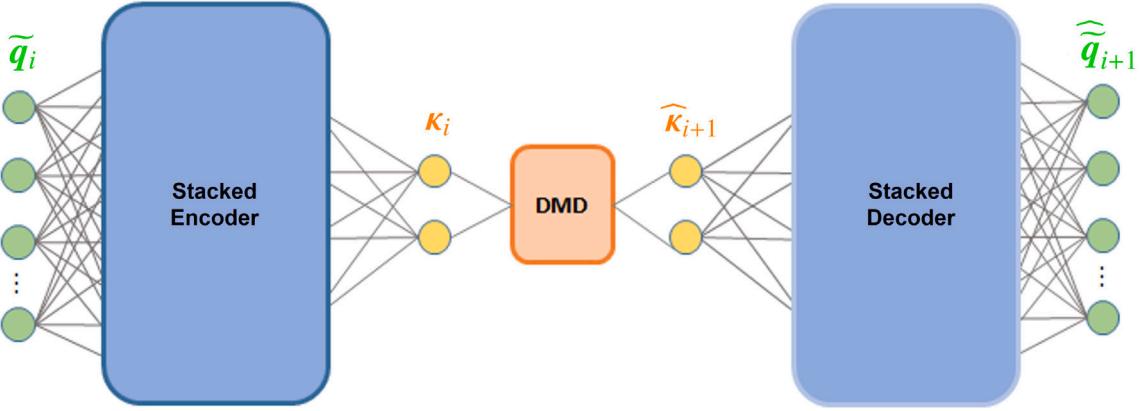


Fig. 2. The data-driven computational framework for non-intrusive reduced-order modelling of fluid flow via stacked autoencoder-based dynamic mode decomposition (SAE-DMD).

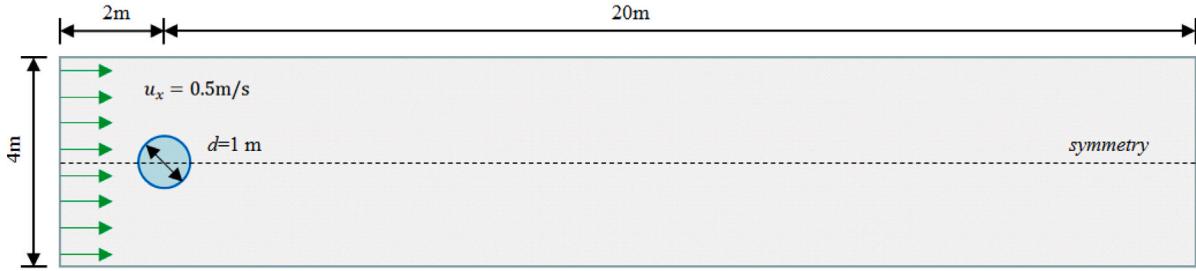


Fig. 3. Schematic diagram of the turbulent flow passing around a single bridge pillar.

follows:

$$RMSE(X(t), \hat{X}(t)) = \sqrt{\frac{1}{K} \sum_{j=1}^K (X_j(t) - \hat{X}_j(t))^2}, \quad (33)$$

where K is the total number of nodes in the mesh or computational domain. The CC measures the strength of the linear relationship between the full-order solution $X(t)$ and the reduced-order solution $\hat{X}(t)$, given by:

$$CC(X(t), \hat{X}(t)) = \frac{cov(X(t), \hat{X}(t))}{\sigma_{X(t)} \sigma_{\hat{X}(t)}} = \frac{E[(X(t) - \mu_{X(t)})(\hat{X}(t) - \mu_{\hat{X}(t)})]}{\sigma_{X(t)} \sigma_{\hat{X}(t)}}, \quad (34)$$

where $\mu_{X(t)}$ and $\mu_{\hat{X}(t)}$ are the means, and $\sigma_{X(t)}$ and $\sigma_{\hat{X}(t)}$ are the standard deviations.

5.1. Case 1: Turbulent flow passing around a single bridge pillar

The first case study is to investigate the turbulent flow passing around a single bridge pillar in a 2D channel, and the schematic diagram of geometry can be seen in Fig. 3. The computational domain is 22.0 m in length and 4.0 m in width, and it includes a circular bridge pillar with diameter $d = 1.0\text{ m}$ located at the point (2.0 m, 2.0 m). The open-source solver *Fluidity* (Pain et al., 2005) is adopted to construct the computational fluid dynamic (CFD) model using large eddy simulation. As shown in Fig. 4, the finite element method (FEM) with unstructured meshes is applied to construct the numerical simulation model of turbulent flow, which contains 12,568 mesh nodes in total.

As to the fluid-solid boundary conditions, the Dirichlet boundary condition (no slip) is applied to the pillar's wall, and zero outward flow boundaries are applied to both the lower and upper edges in the computational domain. Driven by an inlet velocity $u_x = 0.5\text{ m/s}$, the in-flowing liquid enters the 2D channel from the leftmost and passes

around the circular bridge pillar. The fluid kinematic viscosity is $\nu = 1.25 \times 10^{-4}\text{ m}^2/\text{s}$, and the Reynolds number of flow nearby the pillar is calculated to be $Re = u_x d / \nu = 4000$. This high Reynolds number means the turbulent flow regime happening inside the 2D channel. The numerical simulation lasts for 200 s with the time step equal to 0.01 s, from which 2000 snapshots (high-fidelity solutions of velocity components) are obtained at equal time intervals $\Delta t = 0.10\text{ s}$. In the offline stage, the first 1600 snapshots (from 0.1 s to 160.0 s) are used to construct the DMD models and the SAE-DMD models, and the remaining 400 snapshots (from 160.1 s to 200.0 s) are used to test the extrapolation prediction performances of the predictive models.

5.1.1. Dimensionality reduction via autoencoder

As explained in Section 3, the first step of model order reduction is to non-linearly compress the high-dimensional snapshot data (from 0.1 s to 160.0 s) into low-dimensional codes. To demonstrate the superiority of SAE over CAE in this case study, both of them are applied for dimensionality reduction, following which a comparison study is carried out. The SAE consists of multiple fully-connected layers with decreasing number of neurons to derive the latent features, while the CAE leverages convolutional and pooling operations for spatial pattern extraction. As recorded in Tables 1 and 2, the architectures and hyperparameters of SAE and CAE models used here are explained respectively.

The standardised snapshot data (see Eq. (7)) is used as the input for both SAE and CAE, and Adam optimiser with the same learning rate schedule is applied to them for the training purpose. To systematically assess the performances of SAE and CAE, the standardised snapshot data is compressed into the latent features with varying lengths (different values of m). The discrepancy between the raw input and the reconstruction data is used as the indicator to show the accuracy of dimensionality reduction, which can be quantified by RMSE and CC. As shown in Fig. 5, SAE models exhibit much higher accuracy than CAE models in terms of data compression. The main reason can be that

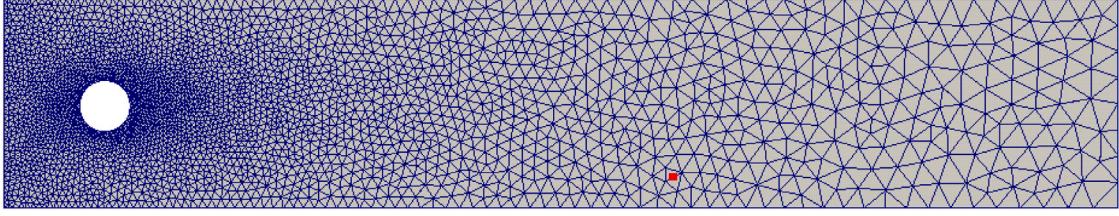


Fig. 4. The CFD model for numerical simulation of turbulent flow passing around a single bridge pillar, which is constructed by using the FEM with unstructured meshes (the red point will be used for further analysis).

Table 1
The architecture and hyperparameters of stacked autoencoder (SAE) models.

| Module | Layer | Number of nodes | Activation function | Adam optimiser | |
|----------------|----------------|-------------------|---------------------|----------------|---------------------------------|
| | | | | Learning rate | Learning rate schedule |
| Encoder | Input | \mathcal{N}_x^a | | | |
| | Dense | $m \times 8$ | ReLU | | |
| | Dense | $m \times 4$ | ReLU | | |
| Code | Dense | $m \times 2$ | ReLU | | |
| | Dense | m^b | ReLU | 0.001 | Reduce learning rate on plateau |
| | Dense | $m \times 2$ | ReLU | | |
| Decoder | Dense | $m \times 4$ | ReLU | | |
| | Dense | $m \times 8$ | ReLU | | |
| | Reconstruction | \mathcal{N}_x | Sigmoid | | |

^a \mathcal{N}_x means the dimension of the original input.

^b m represents the dimension of latent space.

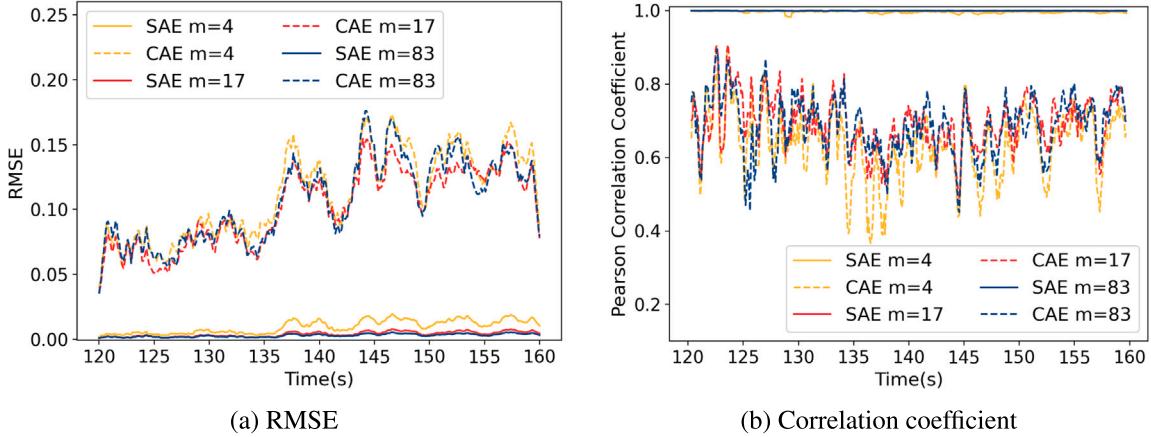


Fig. 5. Performance comparison between SAE and CAE in terms of dimensionality reduction: (a) RMSE and (b) Correlation coefficient are used as the indicators to quantify the discrepancy between the raw input and the reconstructed output.

CAE is specifically designed for processing grid-like data, such as 2D or 3D images, by leveraging convolutional and pooling operations to capture the spatial hierarchies in regular grid, but CAE cannot directly use the raw simulation data with unstructured meshes (see Fig. 4). Here, the raw simulation data is firstly reshaped and flattened into 1D form (called snapshot) and then input into the 1D CAE model for data compression. Compared to SAE, the neurons in 1D CAE are not fully connected, and the additional convolutional and pooling operations may introduce extra errors to data compression. The comparison results demonstrate that SAE is more suitable to deal with numerical simulation data with unstructured mesh, because SAE is more accurate than CAE in processing snapshot data in 1D form.

Furthermore, a parametric study is carried out to analyse the influence of code length m (the dimension of latent space) on dimensionality reduction, and the results are recorded in Table 3. As the code length m increases, more energy of flow dynamics can be captured and maintained by the SAE model. When the code length m reaches 17, the performance of the SAE model starts to become stable, and over 99.9000% of flow dynamical information can be maintained, where

the high-dimensional snapshots (the original length is 12,568) are significantly compressed into low-dimensional codes (the latent length is 17). Here, it should be mentioned that the percentage of captured energy is estimated through principal component analysis.

5.1.2. Constructing predictive model via SAE-DMD

In this section, predictive models of fluid dynamics are constructed by using the classical DMD and the proposed SAE-DMD methods. High-dimensional snapshots (from 0.1 s to 160.0 s) are directly used to build predictive DMD models, while corresponding latent features (codes) obtained from SAE are used to construct SAE-DMD models (as explained in Section 4). To investigate the influence of rank value r on the extrapolation performance of the constructed DMD models, another parametric study is carried out. The velocity magnitude $|u|$ over the time period from 160.1 s to 200.0 s is predicted from these DMD models, and the prediction results are then compared with high-fidelity numerical solutions. Here, RMSE and correlation coefficient are used to quantify the prediction errors, and the results are plotted in Fig. 6. As the rank value r increases, the extrapolation results predicted from

Table 2
The architecture and hyperparameters of convolutional autoencoder (CAE) models.

| Module | Layer | Output data shape | Kernel size | Activation function | Adam optimiser | |
|---------|----------------|---------------------------------------|-------------|---------------------|----------------|---------------------------------|
| | | | | | Learning rate | Learning rate schedule |
| Encoder | Input | (None, \mathcal{N}_x) ^a | | | | |
| | Reshape | (None, \mathcal{N}_x , 1) | | | | |
| | Conv1D | (None, \mathcal{N}_x , 64) | 3 × 1 | ReLU | | |
| | Max pooling | (None, $\mathcal{N}_x/2$, 64) | | | | |
| | Conv1D | (None, $\mathcal{N}_x/2$, 32) | 3 × 1 | ReLU | | |
| | Max pooling | (None, $\mathcal{N}_x/4$, 32) | | | | |
| | Conv1D | (None, $\mathcal{N}_x/4$, 16) | 3 × 1 | ReLU | | |
| | Max pooling | (None, $\mathcal{N}_x/8$, 16) | | | | |
| | Conv1D | (None, $\mathcal{N}_x/8$, 8) | 3 × 1 | ReLU | | |
| | Max pooling | (None, $\mathcal{N}_x/16$, 8) | | | | |
| | Conv1D | (None, $\mathcal{N}_x/16$, 4) | 3 × 1 | ReLU | | |
| | Flatten | (None, $\mathcal{N}_x/4$) | | | | |
| | Dense | (None, $m \times 8$) | | ReLU | | |
| Code | Dense | (None, $m \times 4$) | | ReLU | | |
| | Dense | (None, $m \times 2$) | | ReLU | 0.001 | Reduce learning rate on plateau |
| | Dense | (None, $m \times 4$) | | ReLU | | |
| | Dense | (None, $m \times 8$) | | ReLU | | |
| | Dense | (None, $\mathcal{N}_x/4$) | | ReLU | | |
| | Reshape | (None, $\mathcal{N}_x/16$, 4) | | | | |
| | Conv1D | (None, $\mathcal{N}_x/16$, 8) | 3 × 1 | ReLU | | |
| | Up sampling | (None, $\mathcal{N}_x/8$, 8) | | | | |
| | Conv1D | (None, $\mathcal{N}_x/8$, 16) | 3 × 1 | ReLU | | |
| | Up sampling | (None, $\mathcal{N}_x/4$, 16) | | | | |
| Decoder | Conv1D | (None, $\mathcal{N}_x/4$, 32) | 3 × 1 | ReLU | | |
| | Up sampling | (None, $\mathcal{N}_x/2$, 32) | | | | |
| | Conv1D | (None, $\mathcal{N}_x/2$, 64) | 3 × 1 | ReLU | | |
| | Up sampling | (None, \mathcal{N}_x , 64) | | | | |
| | Conv1D | (None, \mathcal{N}_x , 1) | 3 × 1 | Sigmoid | | |
| | Reshape/Output | (None, \mathcal{N}_x) | | | | |

^a \mathcal{N}_x means the dimension of the original input.

^b m represents the dimension of latent space.

Table 3
The influence of code length m on the results of dimensionality reduction using SAE.

| Code length (m) | Captured energy (%) | Hyper-parameters of the staked encoder | |
|-----------------|---------------------|--|---|
| | | Hidden layers | Neurons |
| 4 | 95.0000 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 7 | 99.0000 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 11 | 99.7000 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 13 | 99.8000 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 17 | 99.9000 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 37 | 99.9900 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 62 | 99.9990 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 83 | 99.9999 | 4 | $m \times 8; m \times 4; m \times 2; m$ |

DMD models become more and more accurate. This positive correlation tendency becomes very weak when the rank value r increases to 37, which means the DMD model reaches its stable state.

From the above parametric studies, we know that the prediction accuracy of a classical DMD model is mainly determined by the rank value r , while the accuracy of an SAE-DMD model is primarily controlled by the code length m because the full-rank value is used to compute the matrix A . In the following content, the proposed SAE-DMD will be systematically compared with the classical DMD by setting the code length m equal to the rank value r . Both the constructed DMD models and the SAE-DMD models are used to predict velocity magnitude, and the discrepancy between the prediction results and the numerical solutions from the high-fidelity CFD models is quantified by RMSE and correlation coefficient. As shown in Fig. 7, when the rank value r and code length m are in small values (≤ 7), the classical DMD models exhibit better extrapolation performance than the SAE-DMD models. As r and m increase to 13, the SAE-DMD model becomes more accurate than the classical DMD model. When r and m increase to 17, the accuracy of the SAE-DMD model is still comparable to that of the classical DMD model, with RMSE value lower than 0.010 and the correlation coefficient higher than 0.998. These results demonstrate

the effectiveness of the proposed SAE-DMD method in model order reduction by accurately preserving the primary characteristics of fluid dynamics.

As shown in Fig. 8, the reduced-order solutions of velocity magnitude field at $t = 160$ s and 200 s are predicted from the SAE-DMD models with varying code lengths m . These reduced-order solutions are compared with the full-order solutions obtained from the CFD models, and the absolute errors between them are computed as well, as can be seen in Fig. 9. Obviously, the extrapolation performance of the SAE-DMD models becomes better as the code length m increases. When the code length m reaches 17, the reduced-order solutions and the full-order solutions become indistinguishable, and the absolute error between them almost vanishes. The excellent prediction performance further demonstrates that the proposed SAE-DMD method is able to construct accurate reduced-order models to predict unseen scenarios of turbulent flow dynamics, when the code length m is sufficiently large.

To show more details of turbulent flow dynamics, the streamlines corresponding to the full-order solutions and reduced-order solutions (predicted from the SAE-DMD models) at $t = 200$ s are plotted in Fig. 10. Obviously, the discrepancy between the full-order and reduced-order solutions decreases as the code length m increases. When the

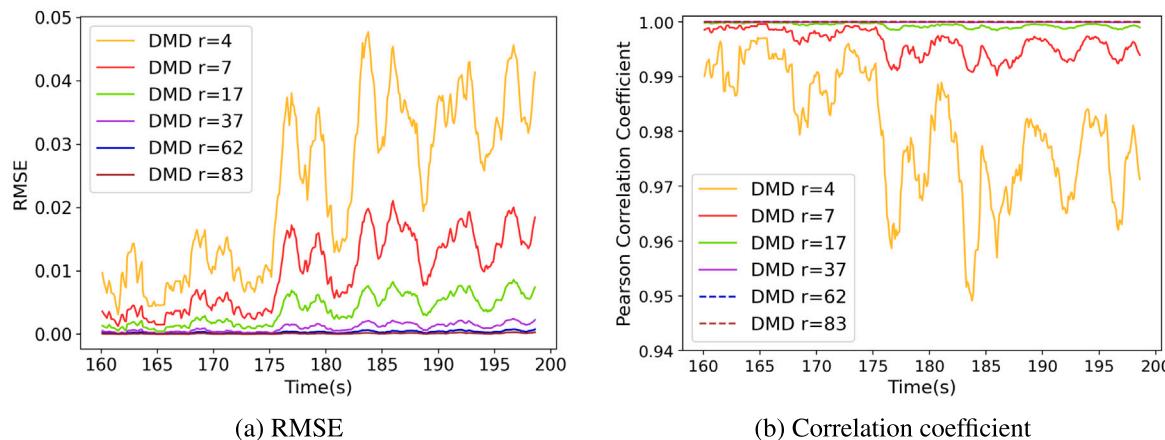


Fig. 6. The influence of the rank value r on the prediction performance of DMD models, where the prediction errors of velocity magnitude are quantified by (a) RMSE and (b) Correlation coefficient.

code length m reaches 37, there is almost no visible difference between the full-order and reduced-order solutions. These results demonstrate the strong capacity of the proposed SAE-DMD-based ROM method in capturing details of turbulent fluid flow.

Additionally, a particular nodal point located at ($x = 13.2233$ m, $y = 0.6065$ m) is selected from the computational domain for further analyses, and the point location is represented by the red node in Fig. 4. The velocity solutions obtained from both the full-order model and the reduced-order models are plotted in Fig. 11. The velocity magnitude at this point varies significantly over time, indicating the turbulent nature of fluid flow dynamics. When the code length $m \geq 17$, the velocity magnitude curves of reduced-order solutions match well with the reference curve of full-order solutions. In Fig. 12, absolute errors are computed to assess the prediction performance of these reduced-order models, and it is clear that the prediction accuracy becomes better as the code length m increases. When m reaches 37, the average magnitude of absolute errors over the time period from 160.1 s to 200 s is lower than 0.005. These results confirm that the proposed SAE-DMD method is capable of constructing reliable reduced-order models to represent turbulent fluid dynamics when an appropriate code length m is selected.

5.2. Case 2: Unsteady flow passing around multiple bridge pillars

The second case study is a real bridge crossing a river in southwest China, as shown in Fig. 13. Compared with Case 1, the unsteady fluid flow passing along multiple pillars can be more complicated, which can be more challenging to the proposed SAE-DMD-based ROM method. As graphically illustrated in Fig. 14, the 3D computational domain is 48.0 m in length, 24.0 m in width and 0.5 m in depth. This computational domain includes two circular pillars with the same diameter of 1.2 m located at the points (12.0 m, 12.0 m) and (17.6 m, 12.0 m) respectively. The open-source solver *OpenFoam* (Jasak et al., 2007) is used to numerically simulate the unsteady flow passing along two bridge pillars, where the finite volume method (FVM) with the non-uniform meshing grid is adopted. This 3D numerical simulation model totally contains 27,634 computational elements, as shown in Figs. 15 and 16.

As to the boundary conditions in this CFD model, periodic boundaries are applied to both the lower and upper borders of the computational domain, and Dirichlet (no slip) boundary conditions are applied to the walls of bridge pillars. The fluid flow is driven by an inlet velocity $u_x = 0.0166$ m/s from the leftmost border, and it passes around two circular bridge pillars in succession. The fluid kinematic viscosity is $\nu = 1 \times 10^{-4}$ m²/s, and the Reynolds number of the flow near the left pillar is calculated to be $Re = u_x d / \nu = 200$, which indicates its unsteady flow characteristics. The numerical simulation of fluid flow lasts for

40,000 s with the time step equal to 1.0 s, from which 4000 snapshots are obtained at equal time intervals $\Delta t = 10.0$ s for velocity components and pressure. In the offline phase, the 601 snapshots corresponding to the time period from $t = 30,000$ to 36,000 s are used to construct the classical DMD models and the new SAE-DMD models, and the final 400 snapshots (from $t = 36,010$ to 40,000 s) are used to test their extrapolation prediction performances in the online phase.

5.2.1. Dimensionality reduction via autoencoder

After acquiring numerical simulation data, the first step of ROM is to compress the high-dimensional snapshots into low-dimensional codes, as explained in Section 3. Here, both SAE and CAE are used for non-linear dimensionality reduction, and the architectures and hyperparameters of SAE and CAE models are explained in Tables 1 and 2 respectively. RMSE and CC are computed to assess the accuracy of data compression by measuring the discrepancy between raw input snapshots and the reconstructed snapshots. As shown in Fig. 17, SAE models show much better performance than CAE models in dimensionality reduction for different lengths (m) of latent feature. These results further confirm that SAE is more suitable than CAE to deal with numerical simulation data with irregular grid.

Besides, a parametric study is carried out to investigate the influence of code length m on the results of dimensionality reduction using SAE. As illustrated in Table 4, more and more energy of flow dynamics can be captured by the SAE model, as the code length m increases. When the code length m reaches 14, over 99.9990% of the flow dynamical energy can be well preserved after dimensionality reduction using SAE, where the high-dimensional snapshots (the original length is 27,634) are significantly compressed into low-dimensional codes (the latent length is 14). Here, it should be mentioned again that the percentage of captured energy is estimated by principal component analysis.

5.2.2. Constructing predictive model via SAE-DMD

Here, both the classical DMD and the proposed SAE-DMD methods are applied to construct predictive model for unsteady fluid dynamics. The DMD models are built based on the high-dimensional snapshot data, while the SAE-DMD models are constructed using the low-dimensional codes after data compression. To investigate the influence of the rank value r on the prediction performance of the classical DMD models, a parametric study is carried out and the results are shown in Fig. 18. The RMSE and correlation coefficient are used to measure the discrepancy between the velocity solutions obtained from the numerical CFD model and the DMD models. It is clear that the prediction performance of the DMD models is continuously improved as the rank value r increases. When r reaches 14, the DMD model possesses

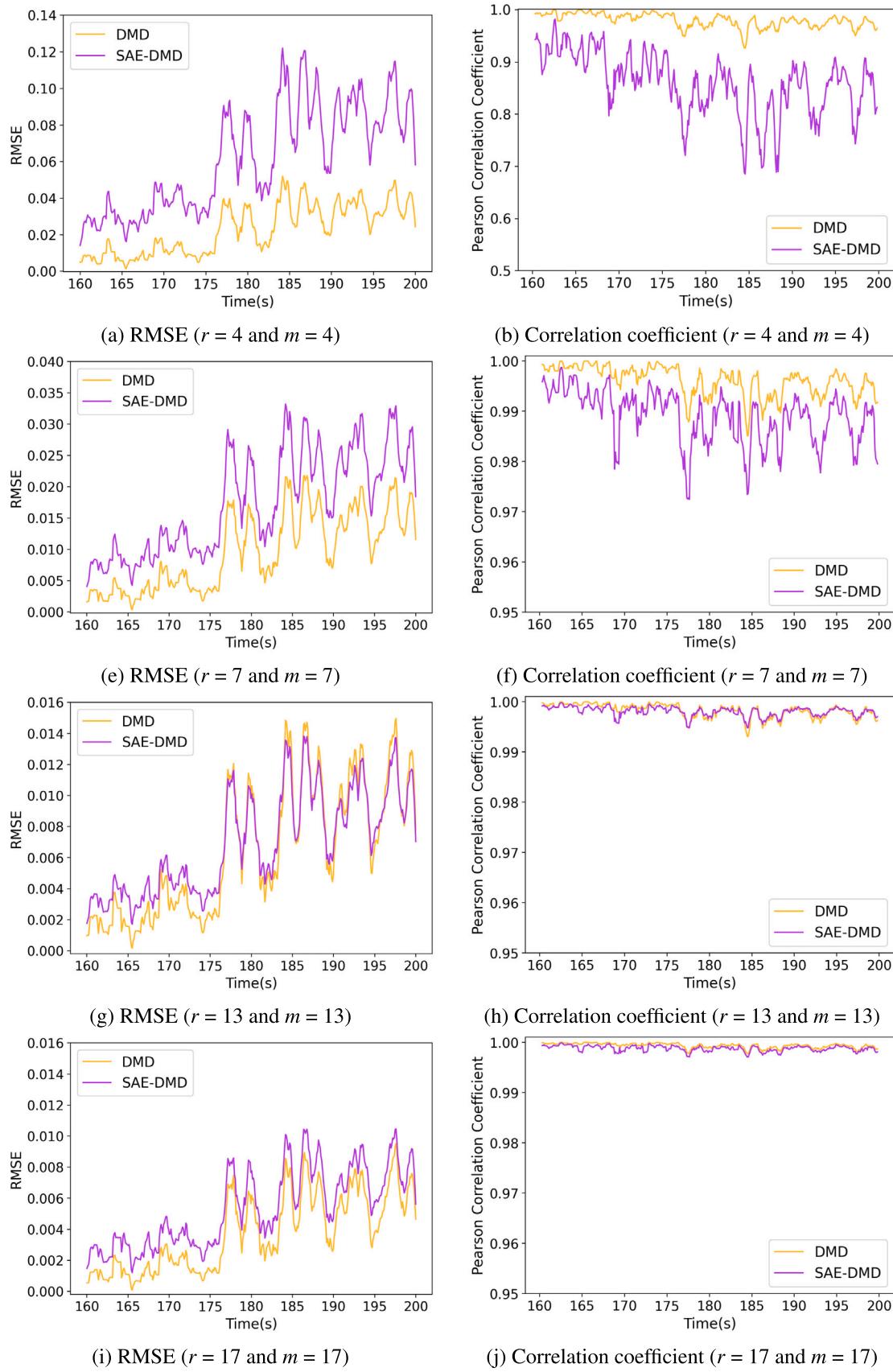


Fig. 7. Performance comparison between the classical DMD models and the developed SAE-DMD models in terms of prediction accuracy, where the high-fidelity numerical solutions of velocity magnitude are used as the ground truth.

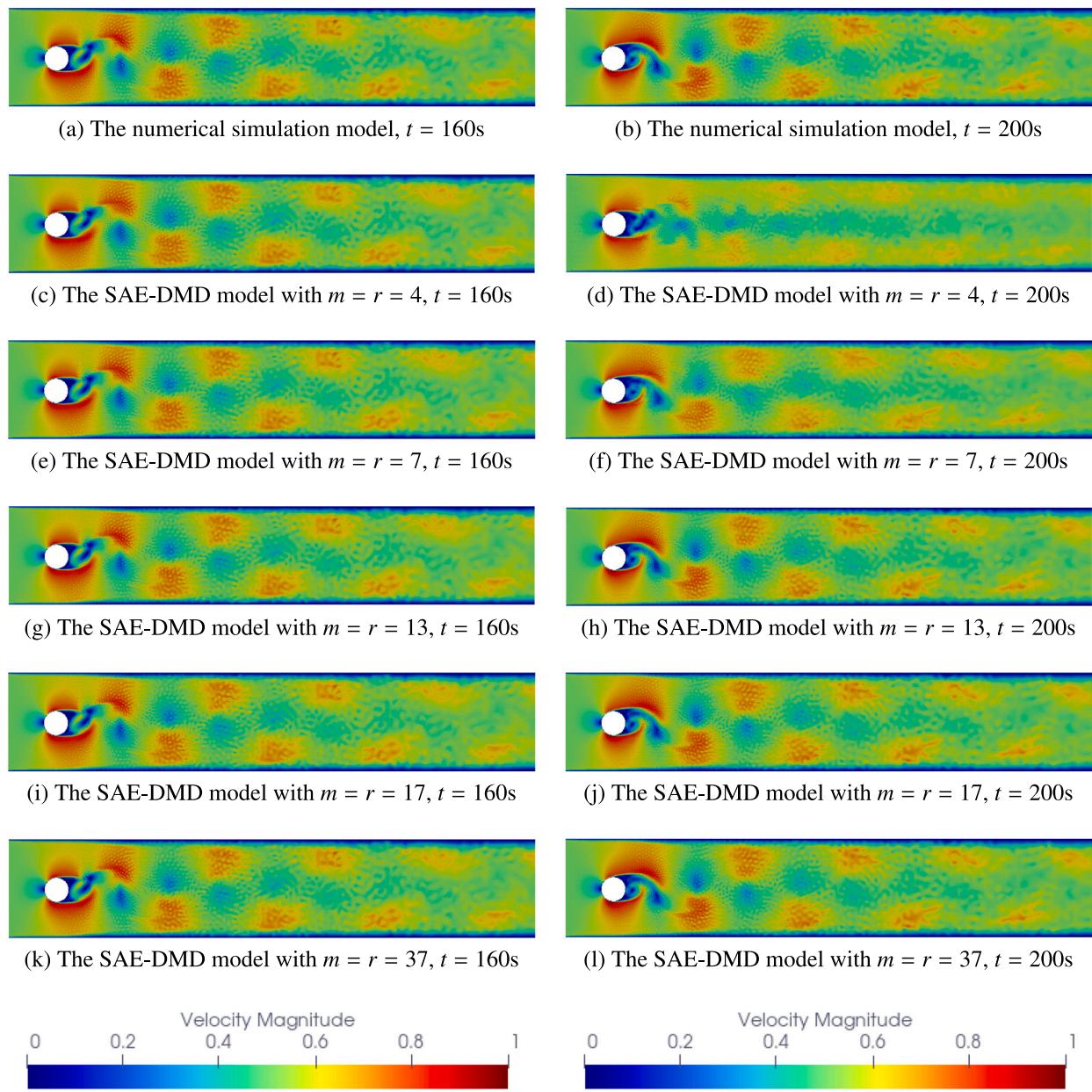


Fig. 8. Comparison between the full-order solutions obtained from the numerical CFD model and the reduced-order solutions predicted from the SAE-DMD models with varying code lengths m .

Table 4

The influence of code length m (the dimension of latent space) on the results of dimensionality reduction using SAE.

| Code length (m) | Captured energy (%) | Hyper-parameters of the staked encoder | |
|---------------------|---------------------|--|---|
| | | Hidden layers | Neurons |
| 3 | 95.0000 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 4 | 99.0000 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 7 | 99.9000 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 10 | 99.9900 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 14 | 99.9990 | 4 | $m \times 8; m \times 4; m \times 2; m$ |
| 19 | 99.9999 | 4 | $m \times 8; m \times 4; m \times 2; m$ |

a very high accuracy level, with the RMSE lower than 0.00025 and the correlation coefficient higher than 0.995.

According to the above parametric analysis results, the prediction accuracy of a classical DMD model is mainly determined by the rank value r , while the accuracy of an SAE-DMD model heavily relies on the code length m because the full-rank value is used to compute the

matrix A . To carry out a systematical comparison study between the classical DMD and the proposed SAE-DMD methods, the code length m is thus set to be equal to the rank value r in all of the following analyses. As can be seen in Fig. 19, both the constructed DMD models and the SAE-DMD models are applied for extrapolation prediction, and the RMSE and correlation coefficient are then calculated to quantify the

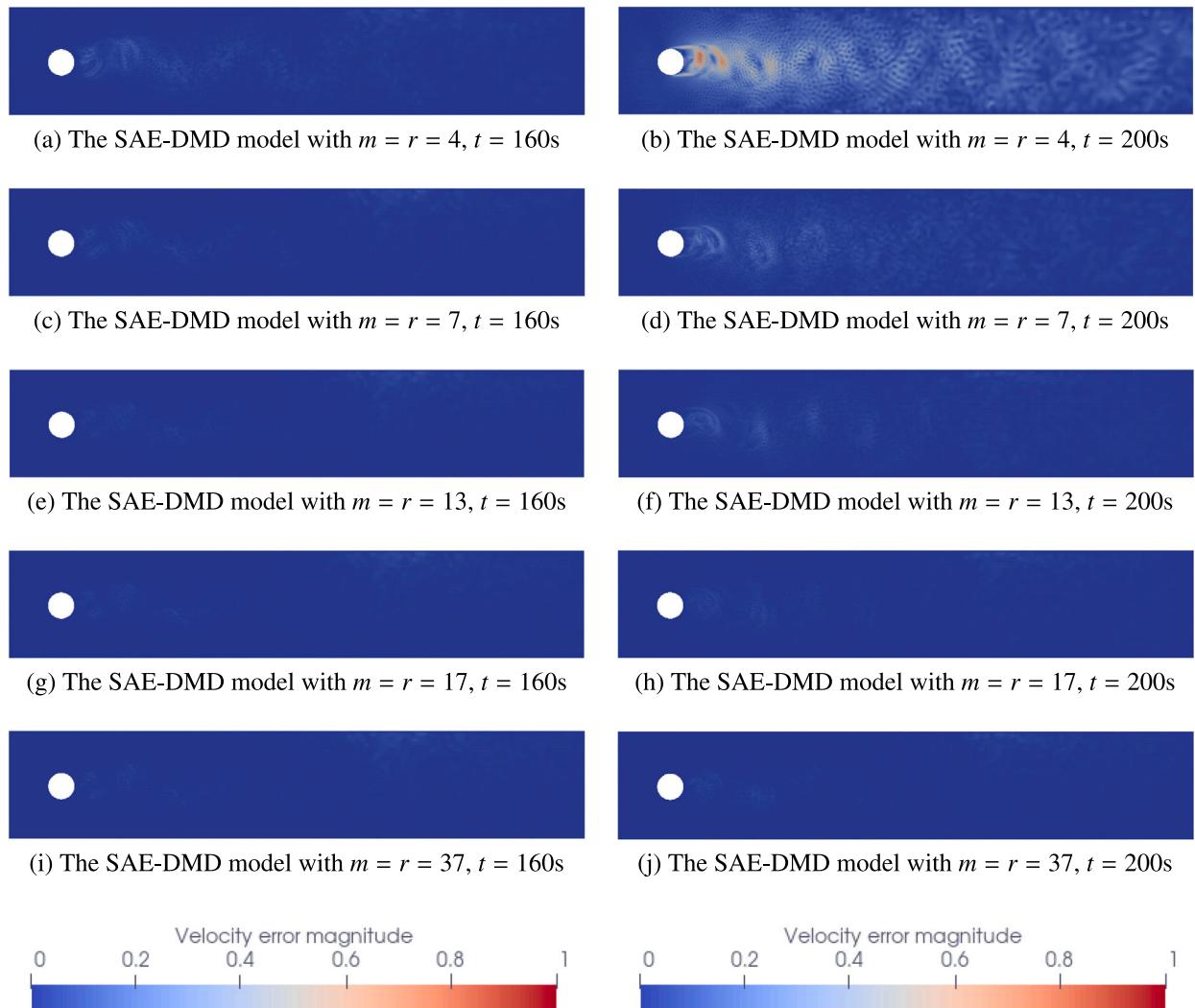


Fig. 9. The absolute errors between the velocity solutions obtained from the numerical CFD model and the SAE-DMD models with varying code lengths m .

discrepancy between the prediction results and the numerical solutions from the high-fidelity CFD model. As the rank value r and the code length m increase, both the classical DMD models and the developed SAE-DMD models exhibit better prediction performance. When r and m increase to 14, both the SAE-DMD models and the DMD models are of high prediction accuracy, where the RMSE values are less than 0.0008 and the correlation coefficients are higher than 0.995. Due to data dimensionality reduction, the extrapolation performance of the SAE-DMD models is slightly inferior to that of the classical DMD models, but the former can still accurately preserve the primary characteristics of unsteady flow dynamics.

In Fig. 20, the velocity magnitude fields obtained from the numerical CFD model and the SAE-DMD models at the time points $t = 36,780\text{s}$ and $39,950\text{s}$ are shown. As the rank value r and the code length m increase, the velocity fields predicted from the SAE-DMD models become closer to the numerical simulation results, and there is no visible difference between them can be found when r and m reach 14. In Fig. 21, the absolute prediction errors of these SAE-DMD models are computed as well, and these errors gradually vanish as r and m increase. The above results demonstrate that the proposed SAE-DMD-based ROM method is capable of maintaining the dynamic details of unsteady fluid flows.

In addition, a particular nodal point located at ($x = 13.7319\text{ m}$, $y = 1.88121\text{ m}$) is selected from the computational domain for further verification, as denoted by the red point in Fig. 15. The velocity

solutions predicted from the SAE-DMD models are compared with the numerical solutions obtained from the CFD model, as shown in Fig. 22. The velocity magnitude at this point varies significantly over time, indicating the unsteady nature of fluid flow dynamics. When the code length $m \geq 10$, the velocity magnitude curves of reduced-order solutions almost coincide with the reference curve of full-order solutions. In Fig. 23, absolute errors are computed to evaluate the prediction performance of the SAE-DMD models, and the prediction error gradually vanishes as the code length m becomes larger. When m reaches 14, the average magnitude of absolute errors over the time period from $36,000\text{s}$ to $40,000\text{s}$ is lower than 0.001. These results further confirm the effectiveness of the proposed SAE-DMD method in constructing reliable reduced-order models for unsteady fluid flows.

6. Discussion and conclusion

6.1. Discussion

In the above section, the proposed SAE-DMD-based ROM method is validated via two case studies, and the results confirm that this method is capable to construct reliable reduced-order models to accurately represent turbulent/unsteady flow dynamics. Considering the objective of model order reduction, computational efficiency is another important index to assess the effectiveness of this new non-intrusive ROM

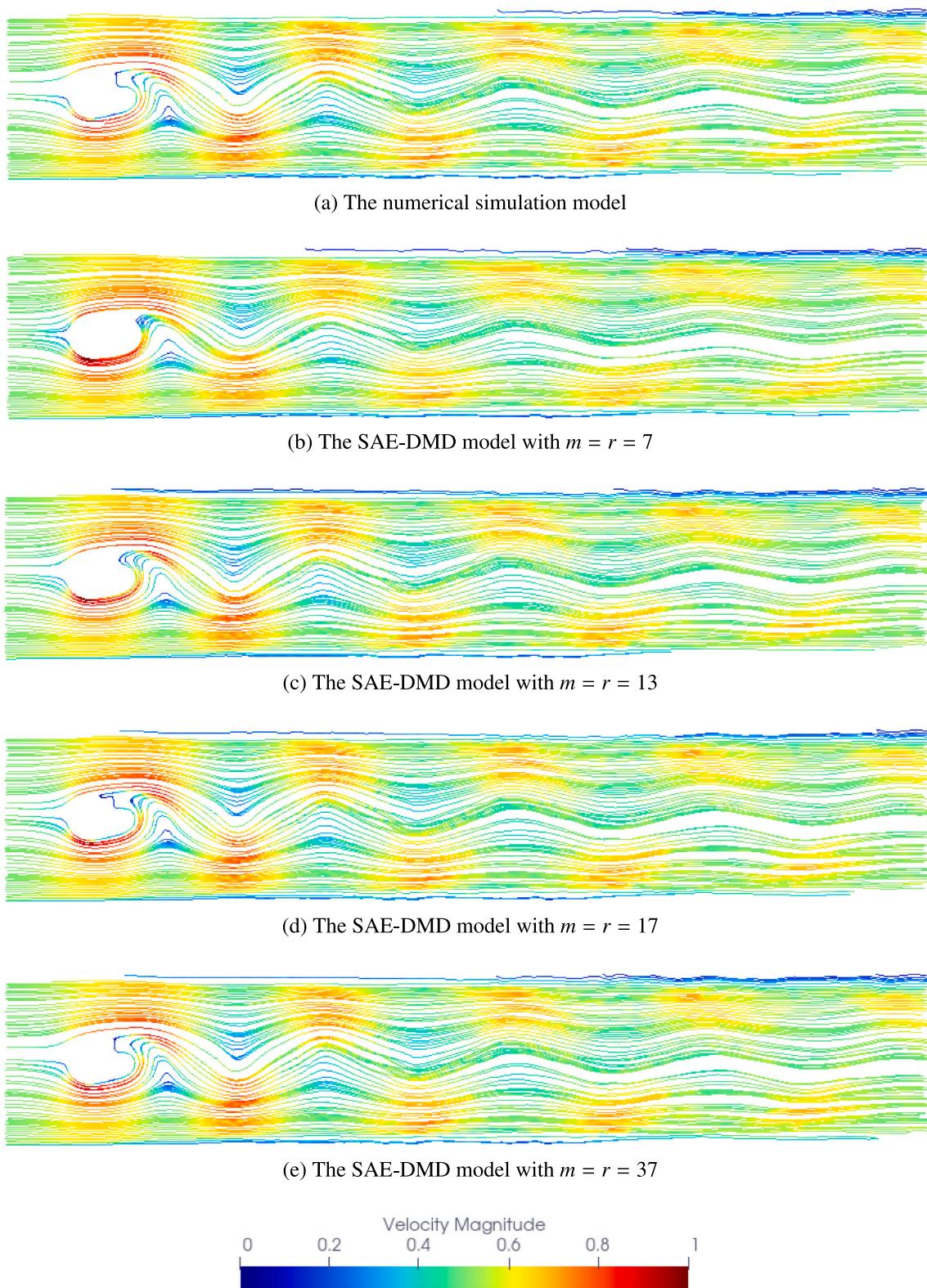


Fig. 10. The streamlines derived from the full-order solutions and the reduced-order solutions at $t = 200$ s.

method. As listed in [Table 5](#), the computational costs of the numerical simulation models, classical DMD models and newly developed SAE-DMD models for each test case are provided. In this work, numerical simulation models are built by using open-source solvers (namely, *Fluidity* and *OpenFoam*), and the constructions of SAE-DMD models and DMD models are implemented by using *Python*. Relevant programmes

are all run on a workstation with two Intel® Xeon® Gold 6226R processors (2.90 GHz frequency) and 256 GB RAM.

Compared to the full-order numerical models and the classical DMD models, the newly developed SAE-DMD models can instantly complete the online prediction of high-dimensional quantities (e.g. velocity field), and this fast prediction advantage will become much

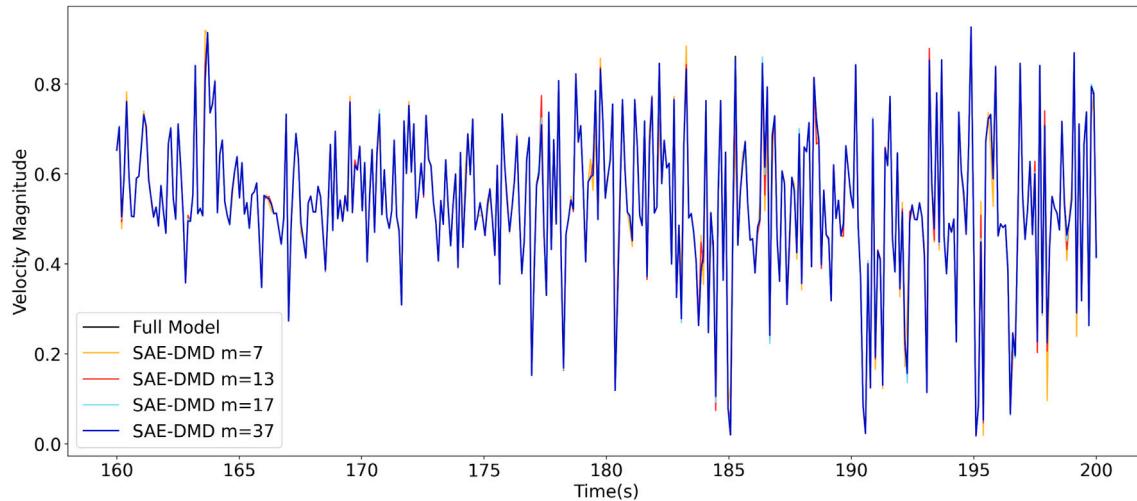
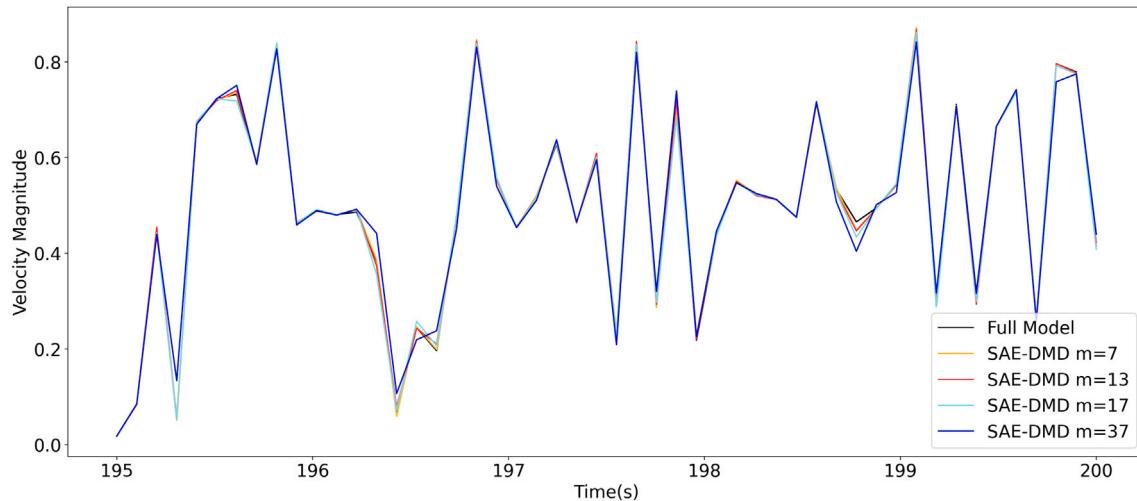
(a) Predicted results of velocity magnitude during the time period from $t = 160.1$ to 200.0s (b) Predicted results of velocity magnitude during the time period from $t = 195.0$ to 200.0s

Fig. 11. The reduced-order solutions of velocity magnitude at the point ($x = 13.2233 \text{ m}$, $y = 0.6065 \text{ m}$) over time, where the SAE-DMD models with different code lengths m are used for extrapolation prediction.

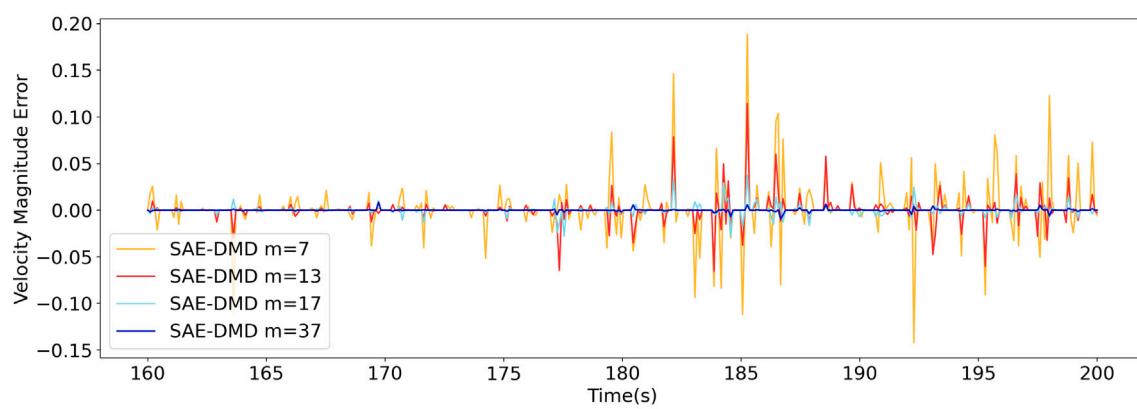


Fig. 12. The absolute errors of the reduced-order solutions of velocity magnitudes at ($x = 13.2233 \text{ m}$, $y = 0.6065 \text{ m}$) over time, where the SAE-DMD models with different code lengths m are used for extrapolation prediction.



Fig. 13. A real case of water flow passing multiple bridge pillars.

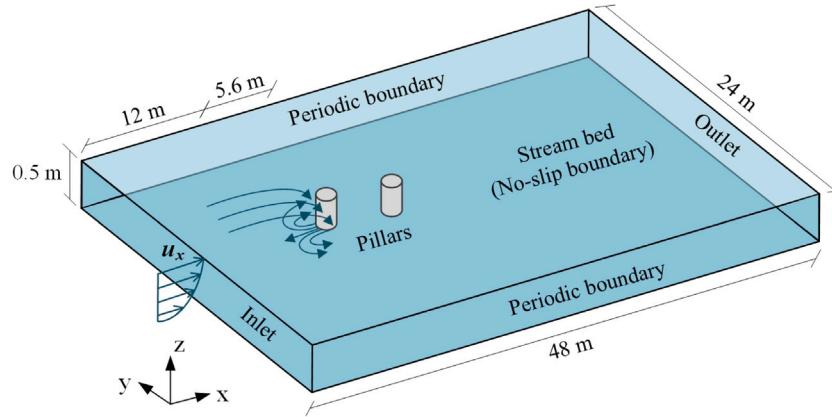


Fig. 14. Schematic diagram of the unsteady flow passing around two circular pillars (the diameters d of circular pillars are 1.2 m).

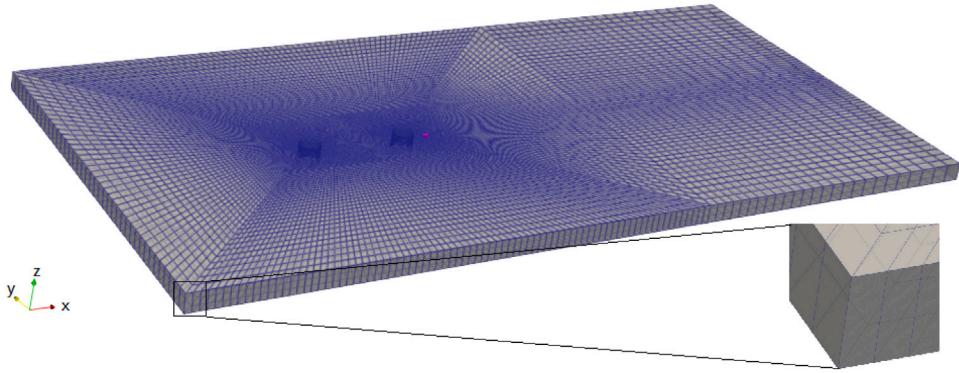


Fig. 15. The 3D CFD model constructed by using the FVM with non-uniform meshes (the red point will be used for further analysis).

more pronounced in coping with complicated and large-scale problems. However, the proposed SAE-DMD method requires around one hundred seconds to train the SAE models for data dimensionality reduction. The training speed of SAE relies heavily on the computing power of a computer, especially the GPU computing capacity. High-performance GPU-based parallel computing can easily accelerate this training process hundreds, even thousands of times. In addition, once the SAE

model is adequately trained, it can be stored for repeat usage, saving computational time in the long term.

Furthermore, the developed SAE-DMD method is also much more memory-saving than classical DMD. For example, directly applying the classical DMD method to the second test case requires a data memory of 72.00 GB due to the high dimensionality of snapshot data, which is beyond the data storage capacity of ordinary computers. By contrast,

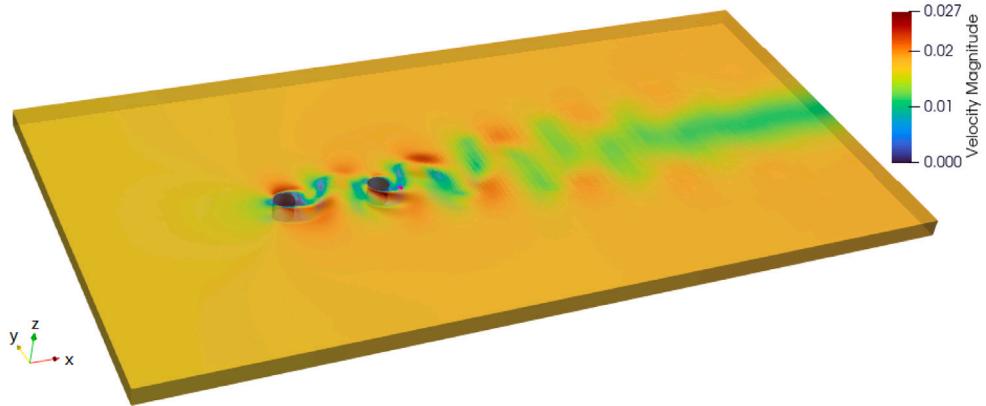


Fig. 16. The numerical simulation result of flow velocity obtained from the 3D CFD model.

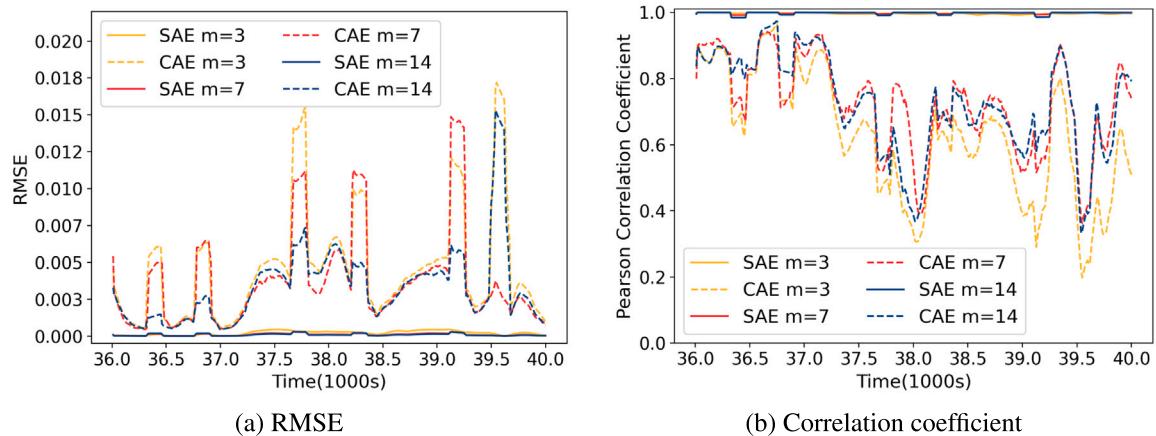


Fig. 17. Performance comparison between SAE and CAE in terms of dimensionality reduction: (a) RMSE and (b) Correlation coefficient are used as the indicators to quantify the discrepancy between the raw input and the reconstructed output.

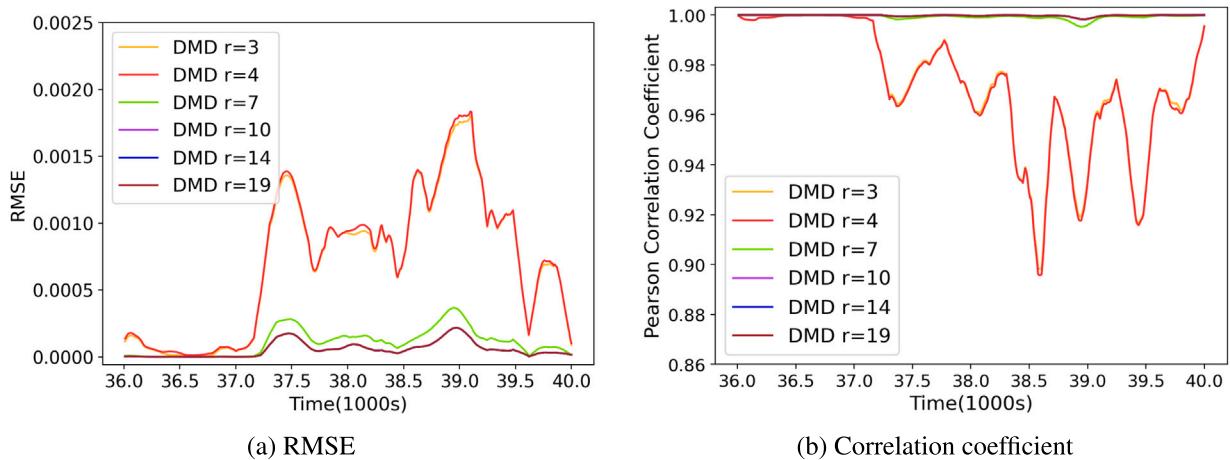


Fig. 18. The influence of the rank value r on the prediction performance of DMD models, where the prediction errors of velocity magnitude are quantified by (a) RMSE and (b) Correlation coefficient.

the memory requirement of the proposed SAE-DMD method is less than 2.20 GB for this case because high-dimensional snapshots have been compressed into low-dimensional codes before performing DMD. The low demand for data memory makes the SAE-DMD method much more applicable to dealing with complicated and large-scale fluid flow problems. Moreover, the prediction accuracy, computational cost and memory requirement of SAE-DMD models are dependent on the code

length m (the dimensionality of latent space), so it is flexible to make a trade-off between them for practical applications.

6.2. Conclusion

The primary contribution of this work is to present a new data-driven computational framework for non-intrusive reduced-order modelling (ROM) of turbulent/unsteady flows passing around bridge piers,

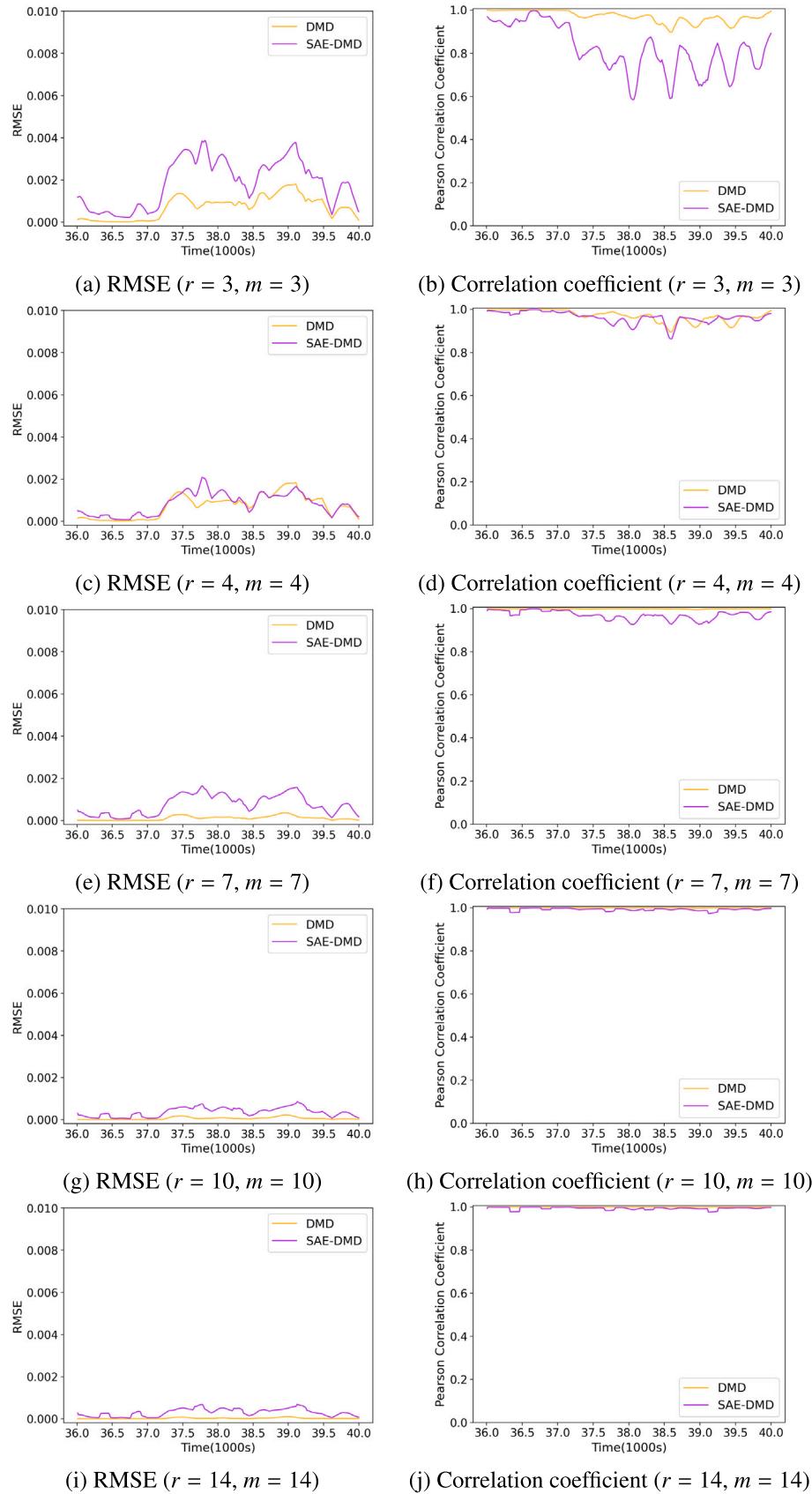


Fig. 19. Performance comparison between the classical DMD models and the new SAE-DMD models in terms extrapolation prediction, where the numerical simulation solutions of velocity magnitude are used as the ground truth.

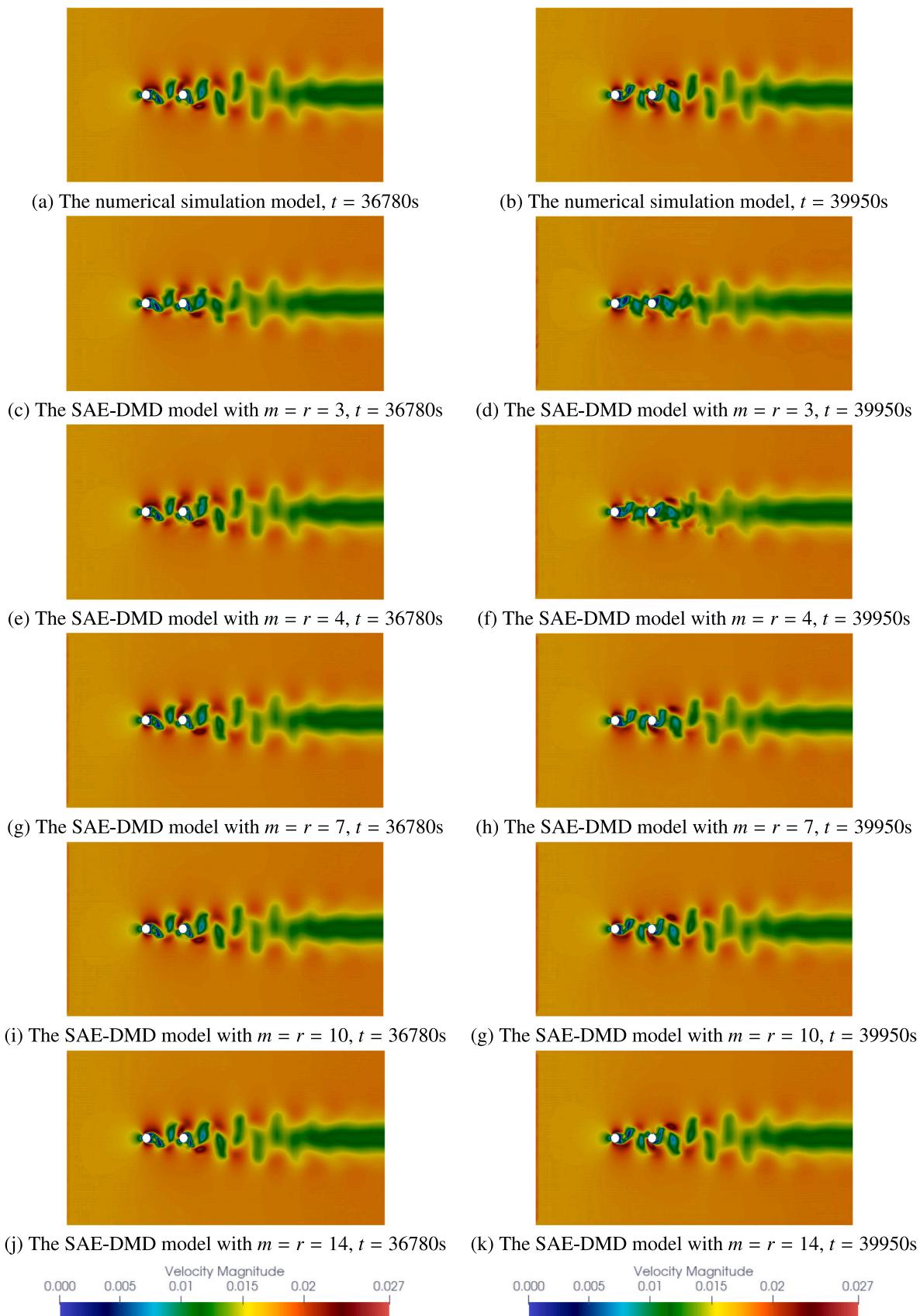


Fig. 20. Comparison between the full-order solutions obtained from the numerical CFD model and the reduced-order solutions predicted from the SAE-DMD models with different code lengths.

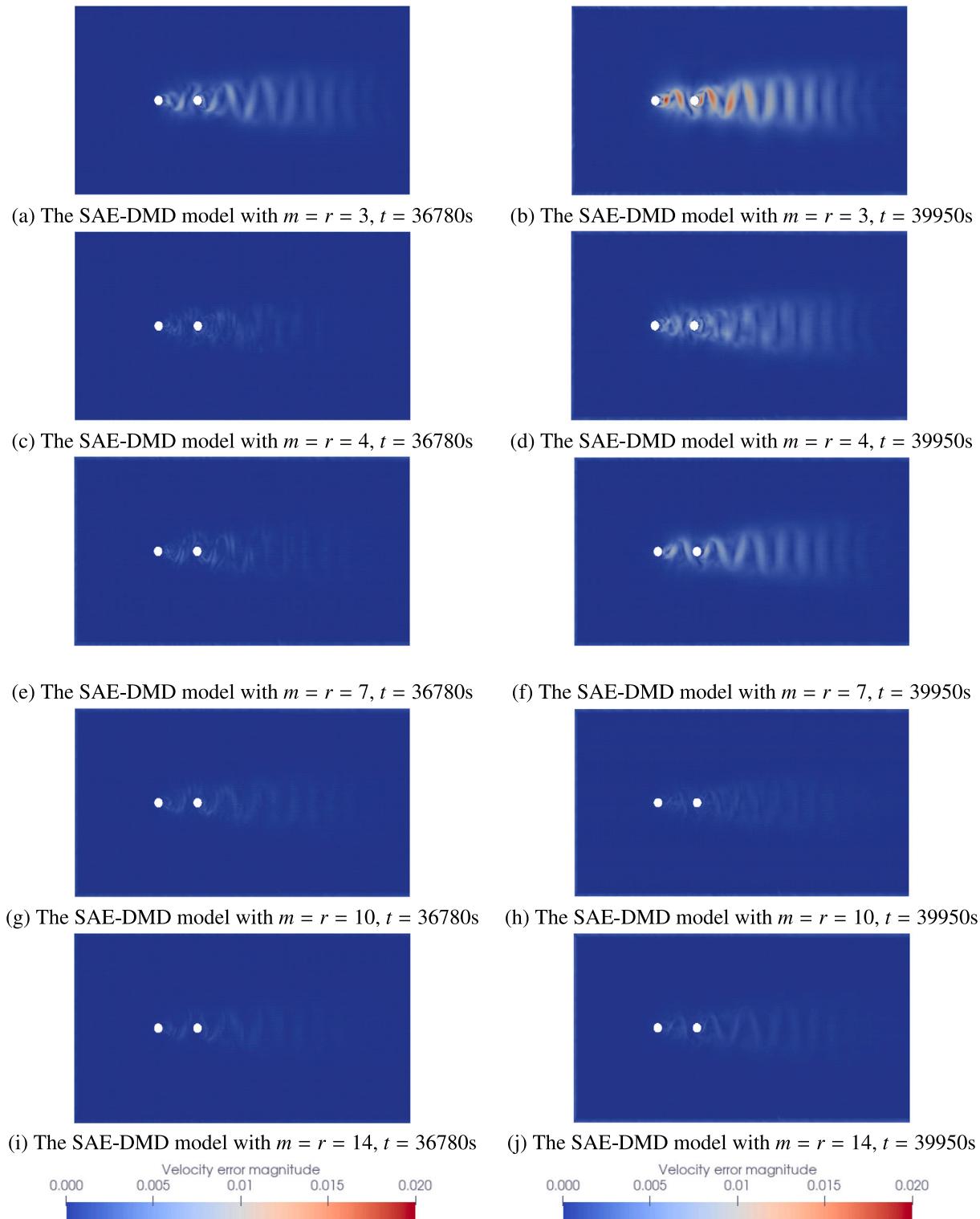


Fig. 21. The absolute errors between the velocity solutions obtained from the CFD model and the SAE-DMD models with different code lengths m .

which is achieved by effectively combining stacked autoencoder (SAE) and dynamic mode decomposition (DMD). More specifically, SAE is used for nonlinear dimensionality reduction to project the full-order numerical system onto a nonlinear latent space, and a data-driven predictive model is then constructed via DMD to represent fluid dynamics in the low-dimensional latent space. The proposed SAE-DMD-based ROM method is applied to two typical fluid flow problems, and

its performance is systematically compared with that of the classical DMD method and high-fidelity numerical simulations via CFD, in terms of modelling accuracy, forecast efficiency and memory requirement. These case studies demonstrate that the combination of SAE and DMD for non-intrusive ROM can considerably accelerate computational speed (300~1000 times faster) and greatly reduce memory requirement (lower than 2%~10%), and the primary nonlinear characteristics of

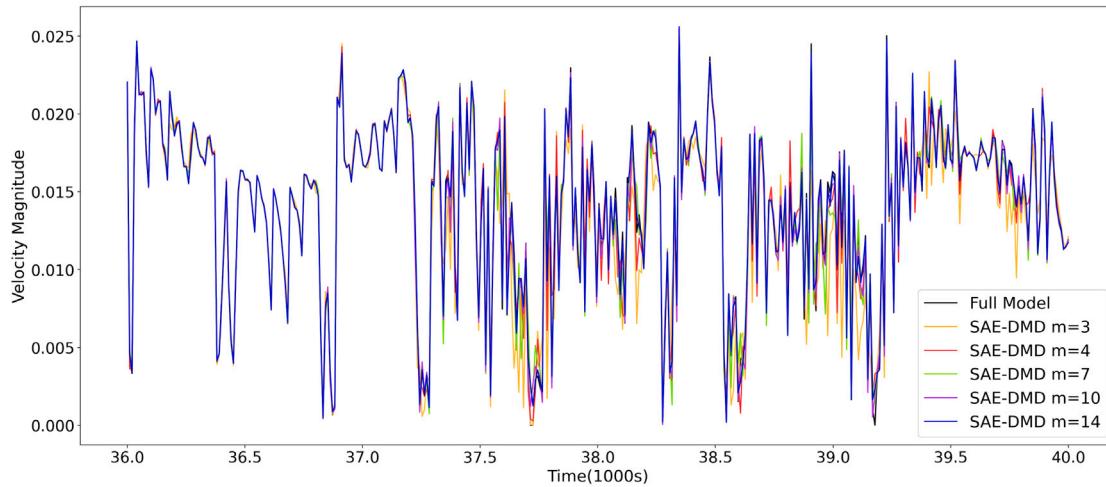
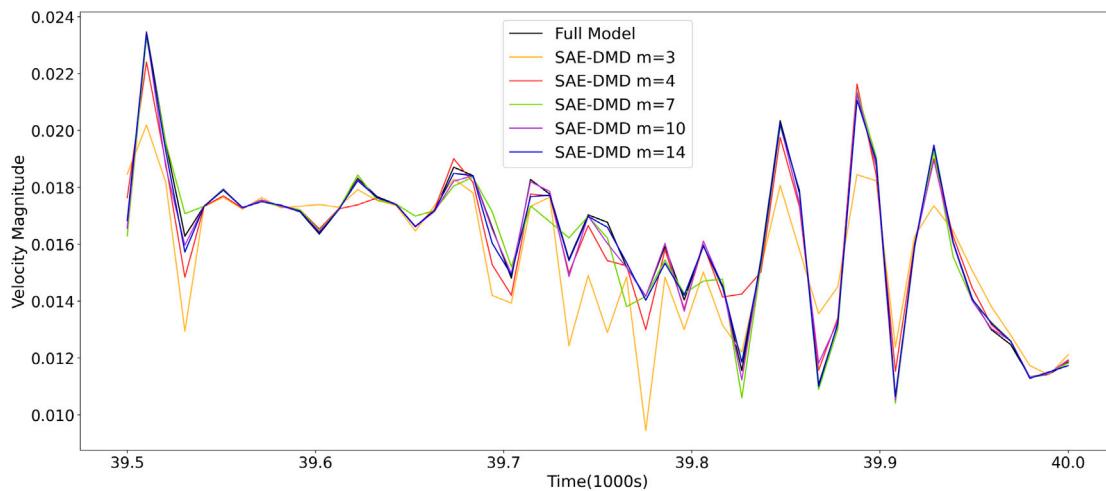
(a) Predicted results of velocity magnitude during the time period from $t=36,000$ to $40,000$ s.(b) Predicted results of velocity magnitude during the time period from $t=39,500$ to $40,000$ s.

Fig. 22. The reduced-order solutions of velocity magnitude at the point ($x = 13.7319$ m, $y = 1.88121$ m) over time, where the SAE-DMD models with different code lengths m are used for extrapolation prediction.

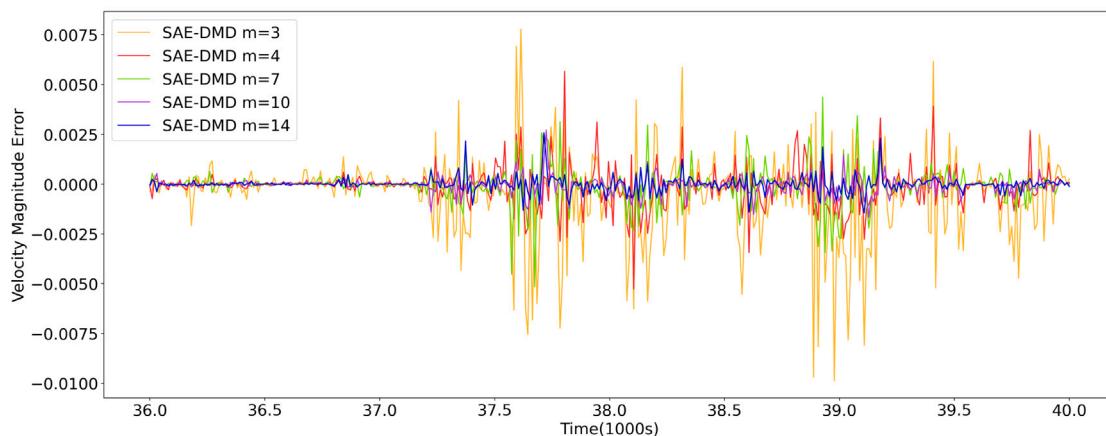


Fig. 23. The absolute errors of the reduced-order solutions of velocity magnitudes at ($x = 13.7319$ m, $y = 1.88121$ m) over time, where the SAE-DMD models with different code lengths m are used for extrapolation prediction.

Table 5

Performance comparisons between different predictive models in terms of computational cost and memory requirement.

| Test case | Predictive models | Number of nodes | Memory requirement (GB) | Time cost (s) | |
|-----------|---|-----------------|-------------------------|------------------------|--------------------------------|
| | | | | Offline model training | Online prediction ^a |
| Case 1 | Numerical simulation model ^b | 12,568 | 8.80 | — | 5.20 |
| | Classical DMD model ($r = 37$) | | 10.90 | 5.0 | 1.83 |
| | SAE-DMD model ($m = 37$) | | 1.11 | 199.5 | 7.92×10^{-3} |
| | SAE-DMD model ($m = 17$) | | 0.51 | 102.9 | 3.65×10^{-3} |
| Case 2 | SAE-DMD model ($m = 7$) | | 0.22 | 88.0 | 1.50×10^{-3} |
| | Numerical simulation model ^c | 27,634 | 18.00 | — | 2.37 |
| | Classical DMD model ($r = 14$) | | 72.00 | 12.0 | 3.50 |
| | SAE-DMD model ($m = 14$) | | 2.20 | 148.20 | 8.62×10^{-3} |
| | SAE-DMD model ($m = 10$) | | 1.54 | 125.70 | 5.10×10^{-3} |
| | SAE-DMD model ($m = 7$) | | 0.88 | 90.60 | 3.75×10^{-3} |

^a The time cost of predicting one snapshot at each time step.^b The finite element model constructed by using *Fluidity*.^c The finite volume model constructed by using *OpenFoam*.

fluid flow dynamics can also be accurately preserved by the constructed reduced-order models.

This new approach shows the potential to overcome the computational infeasibility of high-fidelity numerical modelling of complex flow problems by providing fast and reliable reduced-order solutions, which will be of practical significance for bridge/structure optimisation design and other engineering applications. Future work includes applying this new method to more complicated and realistic cases by considering the scour process and the fluid-structure interaction induced by high-velocity flows passing around bridge piers. A new parametric ROM method will also be developed based on the proposed SAE-DMD approach in the near future to permit fast predictions of fluid dynamics under different initial conditions, fluid properties, pier geometries and shapes, and (dynamical) boundary conditions.

CRediT authorship contribution statement

Chuanhua Zhu: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Dunhui Xiao:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization. **Jinlong Fu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Methodology, Investigation, Funding acquisition, Data curation, Conceptualization. **Yuntian Feng:** Writing – review & editing, Visualization, Supervision, Project administration, Investigation, Funding acquisition, Conceptualization. **Rui Fu:** Writing – review & editing, Validation, Investigation, Data curation. **Jinsheng Wang:** Writing – review & editing, Validation, Software, Resources, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors would like to acknowledge the support from the EPSRC, UK Grant (PURIFY: EP/V000756/1), the FSE IMPACT Fund from Swansea University, UK, and the European Regional Development Fund (the Supercomputing Wales Project). Besides, Prof. Dunhui Xiao acknowledges the support of the Top Discipline Plan of Shanghai Universities-Class I and Shanghai Municipal Science and Technology Major Project, China (No. 2021SHZDZX0100).

References

- Aghaee, Yasin, Hakimzadeh, Habib, 2010. Three dimensional numerical modeling of flow around bridge piers using LES and RANS. In: River Flow 2010. Karlsruhe, pp. 211–218.
- Alemi, Mahdi, Pêgo, João Pedro, Maia, Rodrigo, 2019. Numerical simulation of the turbulent flow around a complex bridge pier on the scoured bed. Eur. J. Mech. B Fluids 76, 316–331.
- Baghbadorani, Danial Amini, Ataei-Ashtiani, Behzad, Beheshti, Aliasghar, Hadjzman, Mahmoud, Jamali, Mirmosadegh, 2018. Prediction of current-induced local scour around complex piers: Review, revisit, and integration. Coast. Eng. 133, 43–58.
- Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T., 2004. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. C. R. Math. 339 (9), 667–672.
- Benner, Peter, Ohlberger, Mario, Cohen, Albert, Willcox, Karen, 2017. Model Reduction and Approximation: Theory and Algorithms. SIAM.
- Blazek, Jiri, 2015. Computational Fluid Dynamics: Principles and Applications. Butterworth-Heinemann.
- Carlberg, K., Bou-Mosleh, C., Farhat, C., 2011. Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. Internat. J. Numer. Methods Engrg. 86 (2), 155–181.
- Chaturantabut, S., Sorensen, D.C., 2010. Nonlinear model reduction via discrete empirical interpolation. SIAM J. Sci. Comput. 32, 2737–2764.
- Chen, Linya, Jeng, Dong-Sheng, 2022. Study on the seabed response around a dumbbell cofferdam under combined wave and current loading. Ocean Eng. 256, 111456.
- Collins, Gary, Fidkowski, Krzysztof, Cesnik, Carlos E., 2020. Petrov-Galerkin projection-based model reduction with an optimized test space. In: AIAA Scitech 2020 Forum. p. 1562.
- Diez, Matteo, Serani, Andrea, Campana, Emilio F., Stern, Frederick, 2022. Time-series forecasting of ships maneuvering in waves via dynamic mode decomposition. J. Ocean Eng. Mar. Energy.
- Drakoulas, G.I., Gortsas, T.V., Bourantas, G.C., Burganos, V.N., Polyzos, D., 2023. SastSVD-ML-ROM: A reduced-order modeling framework based on machine learning for real-time applications. Comput. Methods Appl. Mech. Engrg. 414, 116155.
- Fang, F., Pain, C.C., Navon, I.M., Gorman, G.J., Piggott, M.D., Allison, P.A., Farrell, P.E., Goddard, A.J.H., 2009. A POD reduced order unstructured mesh ocean modelling method for moderate Reynolds number flows. Ocean Model. 28 (1–3), 127–136.
- Fang, F., Zhang, T., Pavlidis, D., Pain, C.C., Buchan, A.G., Navon, I.M., 2014. Reduced order modelling of an unstructured mesh air pollution model and application in 2D/3D urban street canyons. Atmos. Environ. 96, 96–106.
- Fresca, S., Manzoni, A., 2021. Real-time simulation of parameter-dependent fluid flows through deep learning-based reduced order models. Fluids 6 (7), 259.
- Fu, Jinlong, Cui, Shaoqing, Cen, Song, Li, Chenfeng, 2021. Statistical characterization and reconstruction of heterogeneous microstructures using deep neural network. Comput. Methods Appl. Mech. Engrg. 373, 113516.
- Fu, Jinlong, Dong, Jiabin, Wang, Yongliang, Ju, Yang, Owen, D Roger J, Li, Chenfeng, 2020. Resolution effect: An error correction model for intrinsic permeability of porous media estimated from lattice boltzmann method. Transp. Porous Media 132 (3), 627–656.
- Fu, Jinlong, Xiao, Dunhui, Fu, Rui, Li, Chenfeng, Zhu, Chuanhua, Arcucci, Rossella, Navon, Ionel M., 2023a. Physics-data combined machine learning for parametric reduced-order modelling of nonlinear dynamical systems in small-data regimes. Comput. Methods Appl. Mech. Engrg. 404, 115771.
- Fu, R., Xiao, D., Navon, I.M., Fang, F., Yang, Liang, Wang, C., Cheng, S., 2023b. A non-linear non-intrusive reduced order model of fluid flow by auto-encoder and self-attention deep learning methods. Internat. J. Numer. Methods Engrg. 124, 3087–3111.
- Hall, Kenneth C., Thomas, Jeffrey P., Dowell, Earl H., 2000. Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows. AIAA J. 38 (10), 1853–1862.

- Hammond, Janelle Katherine, Chakir, Rachida, Bourquin, Frédéric, Maday, Yvon, 2019. PBDW: A non-intrusive reduced basis data assimilation method and its application to an urban dispersion modeling framework. *Appl. Math. Model.* 76, 1–25.
- Hammond, Brady, Sapsis, Themistoklis P., 2023. Reduced order modeling of hydrodynamic interactions between a submarine and unmanned underwater vehicle using non-myopic multi-fidelity active learning. *Ocean Eng.* 288, 116016.
- Haq, Israr Ul, Iwata, Tomoharu, Kawahara, Yoshihiko, 2022. Dynamic mode decomposition via convolutional autoencoders for dynamics modeling in videos. *Comput. Vis. Image Underst.* 216, 103355.
- Heinz, Stefan, 2020. A review of hybrid RANS-LES methods for turbulent flows: Concepts and applications. *Prog. Aerosp. Sci.* 114, 100597.
- Hinton, Geoffrey E., Salakhutdinov, Ruslan R., 2006. Reducing the dimensionality of data with neural networks. *Science* 313 (5786), 504–507.
- Iqbal, Kife I. Bin, Du, Xin, Uddin, M. Monir, Uddin, M. Forhad, 2022. Balanced truncation for reduced-order modeling of piezoelectric tonpilz transducer on the limited frequency interval. *Appl. Math. Model.* 111, 63–77.
- Jasak, Hrvoje, Jemcov, Aleksandar, Tukovic, Zeljko, et al., 2007. OpenFOAM: A c++ library for complex physics simulations. In: International Workshop on Coupled Methods in Numerical Dynamics. Vol. 1000, pp. 1–20.
- Jia, Yafei, Altinakar, Mustafa, Guney, M. Sukru, 2018. Three-dimensional numerical simulations of local scouring around bridge piers. *J. Hydraul. Res.* 56 (3), 351–366.
- Khosronejad, Ali, Kang, Seokkoo, Sotiropoulos, Fotis, 2012. Experimental and computational investigation of local scour around bridge piers. *Adv. Water Resour.* 37, 73–85.
- Kim, Youngkyu, Choi, Youngsoo, Widemann, David, Zohdi, Tarek, 2022. A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. *J. Comput. Phys.* 451, 110841.
- Kutz, J. Nathan, Fu, Xing, Brunton, Steven L., 2016. Multiresolution dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.* 15 (2), 713–735.
- Lee, Kookjin, Carlberg, Kevin T., 2020. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.* 404, 108973.
- Li, Binghua, Garicano-Mena, Jesús, Valero, Eusebio, 2022. A dynamic mode decomposition technique for the analysis of non-uniformly sampled flow data. *J. Comput. Phys.* 468, 111495.
- Liew, Jaime, Göçmen, Tuhfe, Lio, Wai Hou, Larsen, Gunner Chr., 2022. Streaming dynamic mode decomposition for short-term forecasting in wind farms. *Wind Energy* 25 (4), 719–734.
- Liu, C., Fu, R., Xiao, D., Stefanescu, R., Sharma, P., Zhu, C., Sun, S., Wang, C., 2022. EnKF data-driven reduced order assimilation system. *Eng. Anal. Bound. Elem.* 139, 46–55.
- Lusch, Bethany, Kutz, J. Nathan, Brunton, Steven L., 2018. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Commun.* 9 (1), 4950.
- Miao, Yu-ji, Chen, Xu-jun, Ye, Yong-lin, Ding, Jun, Huang, Heng, 2021. Numerical modeling and dynamic analysis of a floating bridge subjected to wave, current and moving loads. *Ocean Eng.* 225, 108810.
- Mohan, Neethu, Soman, K.P., Kumar, S. Sachin, 2018. A data-driven strategy for short-term electric load forecasting using dynamic mode decomposition model. *Appl. Energy* 232, 229–244.
- Noack, Bernd R., Afanasiev, Konstantin, Morzyński, Marek, Tadmor, Gilead, Thiele, Frank, 2003. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *J. Fluid Mech.* 497, 335–363.
- Oliver, J., Caicedo, M., Huespe, A.E., Hernández, J.A., Roubin, E., 2017. Reduced order modeling strategies for computational multiscale fracture. *Comput. Methods Appl. Mech. Engrg.* 313, 560–595.
- Osth, J., Noack, B.R., Krajnovic, S., Barros, D., Boree, J., 2014. On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body. *J. Fluid Mech.* 747, 518–544.
- Pain, CC, Piggott, MD, Goddard, AJH, Fang, F, Gorman, GJ, Marshall, DP, Eaton, MD, Power, PW, De Oliveira, CRE, 2005. Three-dimensional unstructured mesh ocean modelling. *Ocean Model.* 10 (1–2), 5–33.
- Pawar, S., Rahman, S.M., Vaddireddy, H., San, O., Rasheed, A., Vedula, P., 2019. A deep learning enabler for nonintrusive reduced order modeling of fluid flows. *Phys. Fluids* 31, 08510.
- Proctor, Joshua L., Brunton, Steven L., Kutz, J. Nathan, 2016. Dynamic mode decomposition with control. *SIAM J. Appl. Dyn. Syst.* 15 (1), 142–161.
- Quarteroni, Alfio, Manzoni, Andrea, Negri, Federico, 2015. Reduced Basis Methods for Partial Differential Equations: An Introduction. Vol. 92, Springer.
- Rowley, Clarence W., Dawson, Scott T.M., 2017. Model reduction for flow analysis and control. *Annu. Rev. Fluid Mech.* 49, 387–417.
- Sabetghadam, F., Jafarpour, A., 2012. α Regularization of the POD-Galerkin dynamical systems of the Kuramoto–Sivashinsky equation. *Appl. Math. Comput.* 218 (10), 6012–6026.
- Sashidhar, Diya, Kutz, J. Nathan, 2022. Bagging, optimized dynamic mode decomposition for robust, stable forecasting with spatial and temporal uncertainty quantification. *Phil. Trans. R. Soc. A* 380 (2229), 20210199.
- Schmid, Peter J., 2010. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* 656, 5–28.
- Schmid, Peter J., 2022. Dynamic mode decomposition and its variants. *Annu. Rev. Fluid Mech.* 54 (1), 225–254.
- Srinil, Narakorn, Wiercigroch, Marian, O'Brien, Patrick, 2009. Reduced-order modelling of vortex-induced vibration of catenary riser. *Ocean Eng.* 36 (17–18), 1404–1414.
- Star, Sabrina, Stabile, Giovanni, Rozza, Gianluigi, Degroote, Joris, 2021. A POD-Galerkin reduced order model of a turbulent convective buoyant flow of sodium over a backward-facing step. *Appl. Math. Model.* 89, 486–503.
- Wei, Li, Guo, Xiaoxian, Tian, Xinliang, Zhao, Yakun, 2024. Three-dimensional autoencoder for the flow field reconstruction of an inclined circular disk. *Ocean Eng.*..
- Wu, Pin, Gong, Siquan, Pan, Kaikai, Qiu, Feng, Feng, Weibing, Pain, Christopher, 2021. Reduced order model using convolutional auto-encoder with self-attention. *Phys. Fluids* 33 (7), 077107.
- Wu, Jiujiang, Wang, Lingjuan, Cheng, Qiangong, 2023a. Numerical modeling of current-induced scour around multi-wall foundation using large-eddy simulation. In: Frontiers of Structural and Civil Engineering. Springer, pp. 1–20.
- Wu, Jiaxin, Xiao, Dunhui, Luo, Min, 2023b. Deep-learning assisted reduced order model for high-dimensional flow prediction from sparse data. *Phys. Fluids* 35 (10).
- Xiao, D., Fang, F., Buchan, A.G., Pain, C.C., Navon, I.M., Du, J., Hu, G., 2014. Non-linear model reduction for the Navier–Stokes equations using residual DEIM method. *J. Comput. Phys.* 263, 1–18.
- Xiao, D., Fang, F., Buchan, A.G., Pain, C.C., Navon, I.M., Muggeridge, A., 2015. Non-intrusive reduced order modelling of the Navier–Stokes equations. *Comput. Methods Appl. Mech. Engrg.* 293, 522–541.
- Xiao, D., Fang, F., Du, J., Pain, C.C., Navon, I.M., Buchan, A.G., ElSheikh, A.H., Hu, G., 2013. Non-linear Petrov–Galerkin methods for reduced order modelling of the Navier–Stokes equations using a mixed finite element pair. *Comput. Methods Appl. Mech. Engrg.* 255, 147–157.
- Xiao, D., Fang, F., Pain, C.C., Navon, I.M., 2017a. Towards non-intrusive reduced order 3D free surface flow modelling. *Ocean Eng.* 140, 155–168.
- Xiao, D., Fang, F., Zheng, J., Pain, C.C., Navon, I.M., 2019a. Machine learning-based rapid response tools for regional air pollution modelling. *Atmos. Environ.* 199, 463–473.
- Xiao, D., Heaney, CE, Fang, F., Mottet, L., Hu, R., Bistrian, DA, Aristodemou, E, Navon, IM, Pain, CC, 2019b. A domain decomposition non-intrusive reduced order model for turbulent flows. *Comput. Fluids* 182, 15–27.
- Xiao, D., Yang, P., Fang, F., Xiang, J., Pain, C.C., Navon, I.M., Chen, M., 2017b. A non-intrusive reduced-order model for compressible fluid and fractured solid coupling and its application to blasting. *J. Comput. Phys.* 330, 221–244.
- Yu, Jian, Yan, Chao, Guo, Mengwu, 2019. Non-intrusive reduced-order modeling for fluid problems: A brief review. *Proc. Inst. Mech. Eng. G* 233 (16), 5896–5912.
- Zhu, Hongjun, Zhao, Honglei, Zhou, Tongming, 2019. Direct numerical simulation of flow over a slotted cylinder at low Reynolds number. *Appl. Ocean Res.* 87, 9–25.