

RESEARCH ARTICLE

Decoder decomposition for the analysis of the latent space of nonlinear autoencoders with wind-tunnel experimental data

Yaxin Mo¹, Tullio Traverso^{2,1} and Luca Magri^{1,2,3}

¹ Department of Aeronautics, Imperial College London, London, SW7 2AZ, UK E-mail: l.magri@imperial.ac.uk .

² The Alan Turing Institute, London, NW1 2DB, UK .

³ Politecnico di Torino, DIMEAS, Torino, 10129, Italy .

Keywords: Machine learning, autoencoder, mode decomposition, interpretability, fluid mechanics

Abstract

Turbulent flows are chaotic and multi-scale dynamical systems, which have large numbers of degrees of freedom. Turbulent flows, however, can be modelled with a smaller number of degrees of freedom when using the appropriate coordinate system, which is the goal of dimensionality reduction via nonlinear autoencoders. Autoencoders are expressive tools, but they are difficult to interpret. The goal of this paper is to propose a method to aid the interpretability of autoencoders. This is the *decoder decomposition*. First, we propose the decoder decomposition, which is a post-processing method to connect the latent variables to the coherent structures of flows. Second, we apply the decoder decomposition to analyse the latent space of synthetic data of a two-dimensional unsteady wake past a cylinder. We find that the dimension of latent space has a significant impact on the interpretability of autoencoders. We identify the physical and spurious latent variables. Third, we apply the decoder decomposition to the latent space of wind-tunnel experimental data of a three-dimensional turbulent wake past a bluff body. We show that the reconstruction error is a function of both the latent space dimension and the decoder size, which are correlated. Finally, we apply the decoder decomposition to rank and select latent variables based on the coherent structures that they represent. This is useful to filter unwanted or spurious latent variables, or to pinpoint specific the coherent structures of interest. The ability to rank and select latent variables will help users design and interpret nonlinear autoencoders.

Impact Statement

Nonlinear dimensionality reduction by autoencoders can efficiently compress high-dimensional data into a low-dimensional latent space, but the results may be difficult to interpret. We propose the *decoder decomposition* to select and rank the latent variables based on the coherent structures that they represent. This opens opportunities for building interpretable models with nonlinear autoencoding.

1. Introduction

Turbulent flows are nonlinear and multi-scale systems, which have large numbers of degrees of freedom. High-fidelity simulations of turbulent flows can be performed by solving the governing equations on fine spatiotemporal grids, but the computational cost can be prohibitively high [e.g., 31].

When computationally cheaper modelling of turbulent flows is needed, reduced-order models are applied to approximate the flows with fewer degrees of freedom [e.g., 26, 31]. Commonly, reduced-order models are constructed via projection-based methods, such as proper orthogonal decomposition (POD) [e.g., 26, 31]. POD is a decomposition method dating back to 1970 [21], which enables the computation of an optimal linear subspace based on the energy norm [e.g., 34]. Each POD mode is associated with an energy, which ranks the importance of the mode, and POD modes can be interpreted

as coherent structures of flows [e.g., 3, 15, 29]. There are other methods of linear decomposition, such as the spectral POD [21, 33], dynamic mode decomposition [32, 35], and wavelet analysis [2], which have been employed for dimensionality reduction and discovery of coherent structures. These linear methods are relatively straightforward to implement, but may require large numbers of modes to reduce the approximation error for accurate modelling of nonlinear flows [3, 24]. On the other hand, nonlinear reduced-order modelling seeks nonlinear manifolds onto which perform dimensionality reduction to approximate the dynamics [e.g., 23, 27].

Machine learning has been increasingly applied for dimensionality reduction of fluids, in particular via nonlinear autoencoders [7, 9, 10, 12, 23]. Autoencoders consist of an encoding and a decoding part: the encoding part maps the input (the physical flow field) into a lower-dimensional latent space, whereas the decoding part maps the latent space back to the physical space. The purpose of the autoencoders is to approximate the identity mapping. The turbulent system's dynamics can be predicted on the low-dimensional latent space with sequential methods such as reservoir computers [8, 27] and Long short term memory networks [25] for flow forecasting. The latent space of autoencoder, which is a nonlinear manifold [23], may be difficult to interpret. This is because the latent variables are entangled and their coordinate bases may not be locally orthogonal [23].

The mode-decomposing autoencoder (MD-AE) was developed by Murata et al. [24] to improve the interpretability of nonlinear decomposed fields. MD-AEs assign a decoder to each variable of the latent space vector (latent variable), and then superpose the single-decoder outputs to generate the MD-AE output. MD-AEs improve the visualization of flow decomposition because the effect of each latent variable is isolated in the decoding part, unlike standard AEs in which one decoder contains the effect of all latent variables. With the same latent space dimension, however, an AE outperforms an MD-AE in terms of reconstruction error [10, 24]. This is because the design of MD-AEs does not capture the nonlinear coupling of the decoded latent variables. For highly nonlinear flows, the difference between the reconstruction errors may be significant [7]. Other autoencoder architectures, such as the hierarchical autoencoder [13] and the β -variational autoencoder [10], have been proposed to improve the interpretability of the decomposed fields. The decomposed fields, nonetheless, may be still influenced by more than one latent variable, so the exact effect of each latent variable on the output is difficult to isolate. A comparison between mode decomposition methods, including MD-AE and POD, can be found in the work by Csala et al. [7]. The interpretability of autoencoder latent space is an open problem [23, 36].

The overarching objective of this paper is to aid the interpretability of the latent space of two common autoencoders. For this purpose, we propose the *decoder decomposition*, which is a post-processing method for raking and selecting the latent space of nonlinear autoencoders. The decoder decomposition is based on POD modes and the decoders' sensitivities computed as gradients. Specifically, the objectives are (i) propose the decoder decomposition, which disentangles the contribution of latent variables in the decoded field; (ii) analyse and verify the decoder decomposition of two commonly used autoencoders (the standard autoencoder and the mode-decomposing autoencoder) with synthetic data from numerical simulation of the unsteady two-dimensional wake past a cylinder; and (iii) apply the decoder decomposition to gain physical insight into a realistic flow (three-dimensional turbulent flow from by wind-tunnel experiments) and isolate the latent variables of physical interest with a filtering strategy. The three-dimensional turbulent flow is the high-speed wake past a bluff body.

The paper is organized as follows: First, we provide an overview of the proper orthogonal decomposition (POD) in Section 2 and detail our datasets in Section 3. Next, we introduce the autoencoder architectures in Section 4 and propose the decoder decomposition in Section 5. We apply the decoder decomposition to decompose the unsteady laminar cylinder wake and investigate the impact of the dimension of the latent space on the interpretability of autoencoders (AEs) in Section 6. We decompose

the wind-tunnel wake in Section 7 and demonstrate how to rank and select latent variables to filter for coherent structures in the output of the AEs. We present our conclusions in Section 8.

2. Proper Orthogonal Decomposition

Let $\mathbf{Q} \in \mathbb{R}^{N \times N_t}$ be a generic dataset of a measured fluctuating quantity from a flow field such that each snapshot is a column, where N is the product of the number of grid points, the number of variables, and N_t is the number of snapshots. The i -th row, $\mathbf{Q}_{i,:}$, is the time series of the measured quantity at grid point i . The t -th column, $\mathbf{Q}_{:,t}$, is the snapshot of the measured quantities at the discrete time step t . (In this paper, either the fluctuating velocities or the pressure is measured.) The covariance matrix $\mathbf{C} \in \mathbb{R}^{N \times N}$ is

$$\mathbf{C} = \frac{1}{N_t - 1} \mathbf{W}_p \mathbf{Q} \mathbf{Q}^T \mathbf{W}_p^T = \mathbf{\Phi} \mathbf{\Lambda} \mathbf{\Phi}^T, \quad (2.1)$$

where $\mathbf{W}_p$ is the POD weight matrix given to each element of $\mathbf{Q}$, $\mathbf{\Lambda}$ is the diagonal matrix of the eigenvalues and $\mathbf{\Phi} \in \mathbb{R}^{N \times N}$ is the matrix of the eigenvectors. The weight matrix for each dataset is given in section 3. The i -th column of the matrix of eigenvalues, $\mathbf{\Phi}_{:,i}$, is the POD mode i , which represents the i -th principal axis of the space occupied by the observed data ranked by the eigenvalues, i.e., the energy of the mode [34, 37]. The matrix of temporal coefficients $\mathbf{A}$ is obtained by projecting the data matrix onto the POD modes

$$\mathbf{A} = \mathbf{Q}^T \mathbf{\Phi}, \quad (2.2)$$

which are the temporal coordinates of the data snapshots along the principal axes [37]. The i -th row of $\mathbf{A}$, denoted $\mathbf{A}_{i,:}$, contains the time series of the time coefficient of the i -th POD mode. The time coefficient for the mode i at time step t is $A_{i,t}$. By setting $N_m < N$ as the truncated number of modes for reconstructing the flow, we approximate the flow field in the subspace spanned by the first N_m POD modes

$$\tilde{\mathbf{Q}} = \sum_{i=1}^{N_m} \mathbf{\Phi}_{:,i} \mathbf{A}_{i,:}^T. \quad (2.3)$$

If $N_m = N$, then $\tilde{\mathbf{Q}} = \mathbf{Q}$, i.e., no approximation is made, and only a linear change of coordinate system is performed.

3. Datasets and pre-processing

Two datasets are considered in this paper. First, an unsteady laminar wake behind a cylinder, which is the benchmark case whose dynamics are well-known (section 3.1). Second, an wind-tunnel dataset of a three-dimensional turbulent bluff body wake [30] (section 3.2).

3.1. Unsteady laminar wake of a two-dimensional cylinder

The unsteady laminar wake behind a 2D circular cylinder at Reynolds number $Re = 100$ is generated by solving the dimensionless Navier-Stokes Equations

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \frac{d\mathbf{u}}{dt} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \Delta \mathbf{u} \end{cases}, \quad (3.1)$$

where the vector $\mathbf{u} \in \mathbb{R}^{N_u}$ with $N_u = 2$ is the velocity, p is the pressure, and t is the time. The velocity and length are non-dimensionalised by the inlet velocity U_∞ and the diameter of the cylinder D . The computation domain has size $L_1 = 12$, $L_2 = 5$ and $L_3 = 1$ (Figure 1), divided uniformly into 513, 129 and 8 nodes in streamwise, wall-normal and spanwise directions, respectively. The boundary conditions are

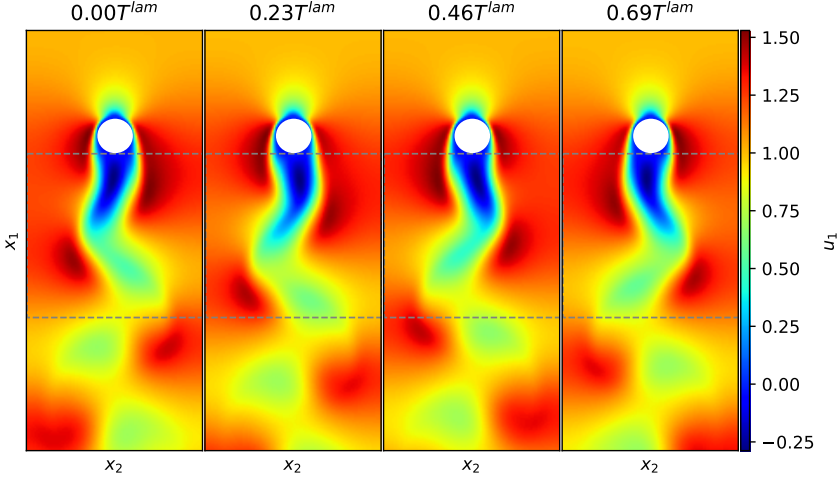


Figure 1: Snapshots of the streamwise velocity of the laminar wake dataset at different times within the same period. A vortex shedding period is denoted with T^{lam} . The area bounded by the grey box is used for training.

Dirichlet boundary condition at $x_1 = 12$; slip walls at $x_2 = 0$ and $x_2 = 5$; and the periodic at $x_3 = 0$ and $x_3 = 1$. The centre of the cylinder is at $(3, 2.5)$. The dataset is simulated with direct numerical simulation using Xcompact3D [6]. A time step $\Delta t = 0.0002$ is chosen to satisfy the Courant–Friedrichs–Lewy condition. The numerical schemes are the 6th-order compact scheme in space [17] and the 3rd-order Adams–Bashforth in time. The simulation matches the results of Ganga Prasath et al. [14].

The transient period of the first 100 time units at the beginning of the simulation is discarded to ensure the dataset contains only the periodic flow [14]. Snapshots of the streamwise velocity, u_1 , are saved every 0.125 time units, giving over 40 snapshots per vortex shedding period, T^{lam} . The final dataset contains 720 snapshots. We consider a 200-by-129 grid behind the cylinder (grey box in Figure 1). The domain includes areas from 0 to $4.5D$ downstream of the body, capturing the near wake and the vortex shedding [22]. The dataset used by the AEs, $\mathbf{U} \in \mathbb{R}^{N_1 \times N_2 \times N_t}$, consists of the fluctuating streamwise velocity, where $N_t = 720$ is the number of snapshots, $N_1 = 200$ and $N_2 = 129$ are the number of grid points in the streamwise and wall-normal directions, respectively.

We perform POD on the dataset $\mathbf{U}$ to obtain the POD modes Φ^{lam} and the matrix of time coefficients $\mathbf{A}^{lam}$, referred to as the “data modes” and “data time coefficients”, respectively, where *lam* stands for “laminar”. Examination of the eigenvalues shows that the data modes are out-of-phase pairs, the two modes in a pair contain a similar amount of flow energy (Figure 2, left panel) and their time coefficients are 90° out-of-phase (Figure 2, centre panel). The first two data modes represent the vortex shedding, at the vortex shedding frequency, and the higher modes oscillate at the harmonic frequencies [19]. The magnitude of the fast Fourier transform in time, summed over the grid points, is used to measure the overall frequency content of the dataset, shown in Figure 2 (right panel). The frequency content of the flow is described in terms of the Strouhal number, $St = fD/U_\infty$, where f is the frequency measured in Hz. The dominant frequency is the vortex shedding frequency at $St = 0.23$, and the dataset also contains higher harmonic frequencies. Each pair of data modes oscillates at a single, distinct frequency (Figure 2, right panel). The first pair, comprised of data modes 1 and 2, oscillate at the vortex shedding frequency, while the subsequent pairs, composed of data modes 3 and 4, oscillate at the first harmonic, and so forth. The concentration of the flow energy in the leading data modes and the distinct frequencies of the data

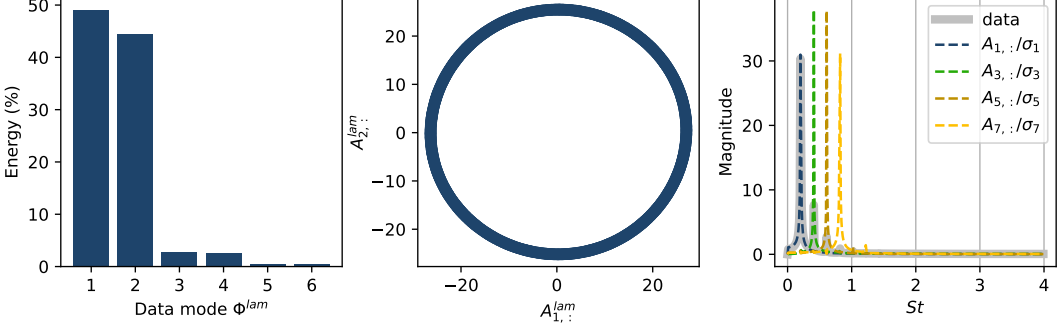


Figure 2: POD of the laminar wake $\mathbf{U}$. Left: the percentage energy contained in the first six POD modes of the unsteady wake Φ^{lam} (data mode). Data modes 1&2, 3&4 and 5&6 contain similar flow energy and oscillate at the same frequency but out of phase. Centre: phase plot of the first two data time coefficients. Right: the frequency spectrum of the data and the data time coefficients 1, 3, 5 and 7, normalized by the standard deviation. The data contains the vortex shedding frequency and its harmonics. (Since each pair has the same frequency spectrum, only the odd data modes are shown here).

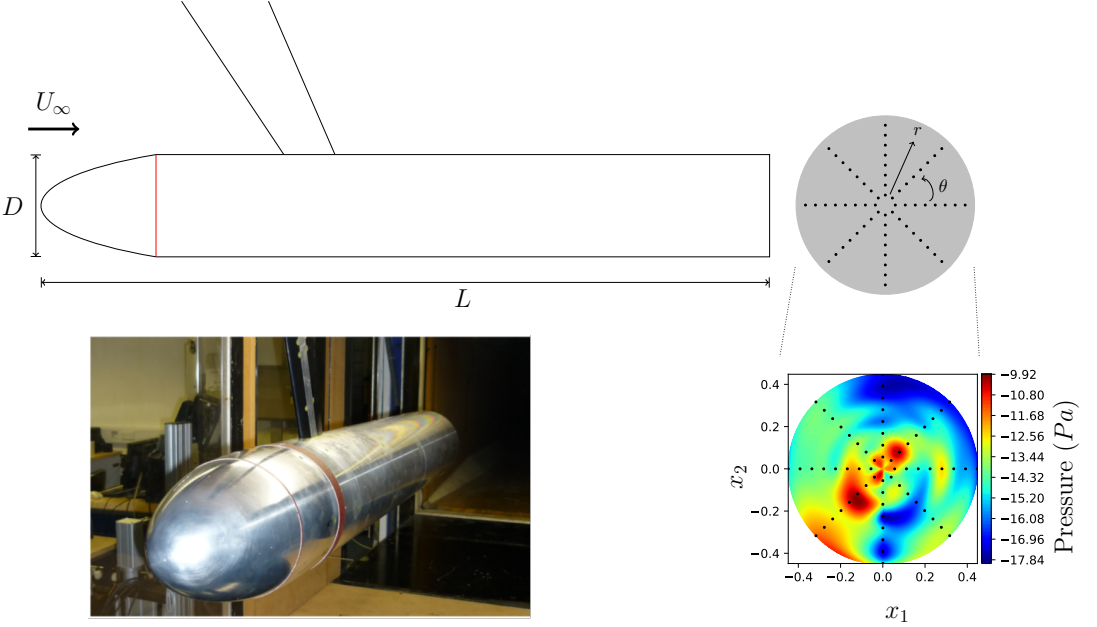


Figure 3: Experimental set-up, reproduced from [28]. The dimensions x_1 and x_2 are the measured location nondimensionalized by the diameter D . The black dots mark the location of the pressure sensors.

time coefficients make the unsteady wake behind a cylinder suitable for dimensionality reduction and interpretation with POD.

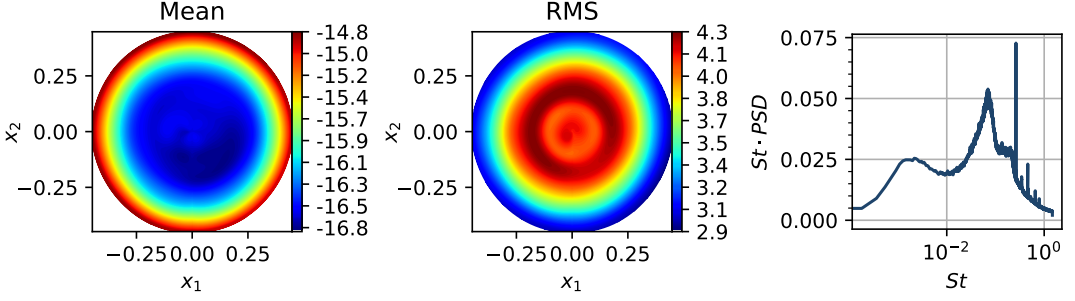


Figure 4: The wind-tunnel pressure dataset $\mathbf{P}$. Left: Mean pressure. Centre: RMS pressure. Right: The premultiplied PSD ($St \cdot \text{PSD}$) of the wind-tunnel dataset, with peaks at $St \approx 0.002, 0.06$ and $St \approx 0.2$ and its harmonics. The peaks correspond to the three-dimensional rotation of the wake, the pulsation of the vortex core, and the vortex shedding and its harmonics, respectively.

3.2. Turbulent wake of a bluff body from wind tunnel experiments

The turbulent wake of an axisymmetric body from wind tunnel experiments [29, 30] is employed in the second part of this paper. Figure 3 shows the experimental setup. The axisymmetric body has a diameter of $D = 196.5$ mm and a length-to-diameter ratio $L/D = 6.5$. Pressure measurements at the base of the body are collected via 64 static pressure sensors placed on a regular polar grid, with eight grid points in the radial direction and eight grid points in the azimuthal direction (Figure 3). We provide here an overview of the dataset. For a more in-depth analysis of the dataset and the experimental setup, the readers are referred to Rigas et al. [29].

The mean and root-mean-squared (RMS) pressures are both axisymmetric (Figure 4). The power spectral density (PSD) of the pressure data is shown in Figure 4 (right), premultiplied by the Strouhal number to improve the visualization [29]. The peaks at $St \approx 0.002, 0.06$ and 0.2 are associated with the 3D rotation of the symmetry plane, pulsation of the vortex core and vortex shedding, respectively.

The weighted POD (each data point is weighted by the area of the element of the polar grid) is applied to the dataset [29, 30]. The resulting POD modes Φ^{exp} and time coefficients $\mathbf{A}^{exp}$ are the data modes and data time coefficients, respectively, where *exp* stands for “experimental”. Figure 5 shows the energy contained in each data mode of the wind-tunnel dataset and the cumulative energy of the modes. The flow energy is spread over more modes than the laminar case, and 21 data modes are needed to reconstruct the dataset to recover 95% of the energy. The modes and the premultiplied PSD of their time coefficients are shown in Figure 6. Data modes 1 and 2 are antisymmetric and have frequency peaks at $St \approx 0.002, 0.1$ and 0.2 . As a pair, data modes 1 and 2 represent vortex shedding and the ‘very-low-frequency’ 3D rotation of the symmetry plane [29]. Data mode 3 has a frequency peak at $St \approx 0.06$ and is associated with the pulsation of the vortex core [29, 30]. The data modes 4 and 5 contain the subharmonic frequencies. Data modes 1 to 5 contain the coherent structures of the dataset. We base our analysis and latent variable selection on the interpretation of the data modes Φ^{exp} provided here. We consider the fluctuating field, following the same logic as for the laminar cylinder wake dataset. The final experimental pressure dataset employed for training is $\mathbf{P} \in \mathbb{R}^{N \times N_t}$, where $N_t = 420000$ is the number of available snapshots in time, sampled at a sampling frequency 225 Hz and $N = 64$ is the number of pressure sensors. The dataset $\mathbf{P}$ contains data collected for over 1800 s.

4. Autoencoder architectures

An autoencoder approximates the identity function by means of an encoder and a decoder. We define a generic $\mathbf{Y} \in \mathbb{R}^{N^* \times N_t}$, where N^* is the spatial dimension. For all time steps, the encoder $F_{en} : \mathbb{R}^{N^*} \rightarrow \mathbb{R}^{N_z}$ maps the snapshot of the data at time t , $\mathbf{Y}_{:,t}$ into a latent space, represented by a latent vector $\mathbf{Z}_{:,t}$, where

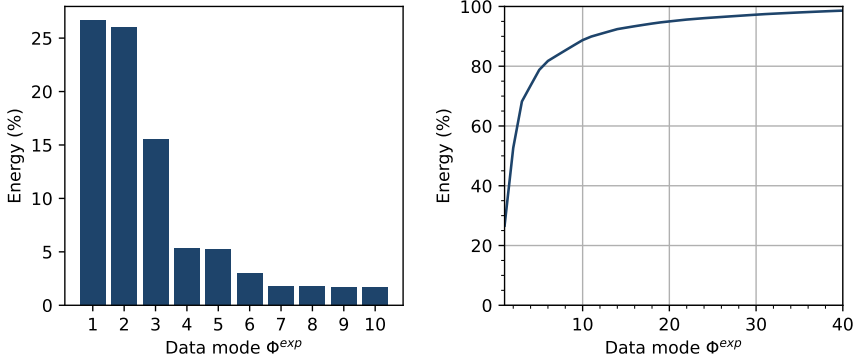


Figure 5: POD of the wind-tunnel dataset $\mathbf{P}$. Left: percentage energy of the first 10 data modes. Right: cumulative percentage energy of POD modes. The reconstruction of the pressure dataset to 95% energy needs 21 data modes.

$N_z \ll N^*$ is the dimension of the latent space. The i -th latent variable is $\mathbf{Z}_{i,:}$. The encoder, F_{en} , is a composition of layers $f_{en}^{(1)}, f_{en}^{(2)}, \dots, f_{en}^{(n)}$ and activation functions $\theta_{en}^{(1)}, \theta_{en}^{(2)}, \dots, \theta_{en}^{(n)}$ applied to each layer. The decoder, F_{de} , is a composition of layers $f_{de}^{(1)}, f_{de}^{(2)}, \dots, f_{de}^{(n)}$ and the activation functions $\theta_{de}^{(1)}, \theta_{de}^{(2)}, \dots, \theta_{de}^{(n)}$ applied to each layer

$$\begin{aligned} F_{en} &= \theta_{en}^{(n)} \circ f_{en}^{(n)} \circ \theta_{en}^{(n-1)} \circ f_{en}^{(n-1)} \dots \circ \theta_{en}^{(1)} \circ f_{en}^{(1)}, \\ F_{de} &= \theta_{de}^{(n)} \circ f_{de}^{(n)} \circ \theta_{de}^{(n-1)} \circ f_{de}^{(n-1)} \dots \circ \theta_{de}^{(1)} \circ f_{de}^{(1)}, \end{aligned} \quad (4.1)$$

where n is the number of layers. Each layer ($f_{en}^{(*)}$ or $f_{de}^{(*)}$) maps the output of the previous layer to the input of the next layer, and is a function of the trainable parameters ($\omega_{en}^{(*)}$ or $\omega_{de}^{(*)}$). Each layer is either a convolution or a linear mapping depending on the test case and the datasets. Details of layers and activation functions are explained in sections 4.1 and 4.2.

The decoding part maps the instant latent vector at t back to the original space, with the output $\hat{\mathbf{Y}}_{:,t}$ approximating the input $\mathbf{Y}_{:,t}$. The autoencoder, comprised of the encoding and decoding parts, is trained with all t to output $\hat{\mathbf{Y}}$, which approximates the input $\mathbf{Y}$. The error between $\hat{\mathbf{Y}}$ and $\mathbf{Y}$ is measured with the mean squared error (MSE)

$$MSE(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N^* \times N_t} \sum_{i=1}^{N^*} \sum_{t=1}^{N_t} (Y_{i,t} - \hat{Y}_{i,t})^2. \quad (4.2)$$

The parameters of the autoencoder, $\omega = \{\omega_{en}^{(1)}, \dots, \omega_{en}^{(n)}, \omega_{de}^{(1)}, \dots, \omega_{de}^{(n)}\}$, are obtained by minimizing the MSE

$$\omega^* = \arg \min_{\omega} MSE(\mathbf{Y}, \hat{\mathbf{Y}}). \quad (4.3)$$

This analysis includes two different autoencoder architectures — the standard autoencoder (AE) (section 4.1) and the mode-decomposing autoencoder (MD-AE) (section 4.2). To compare the results with linear decomposition methods (POD), we use networks with no bias and train on the fluctuating fields, so our results are one-to-one comparable to those obtained with POD. Two types of intermediate layers are employed — convolutional layers for the laminar dataset, and feedforward layers for the wind-tunnel dataset to handle the polar grid.

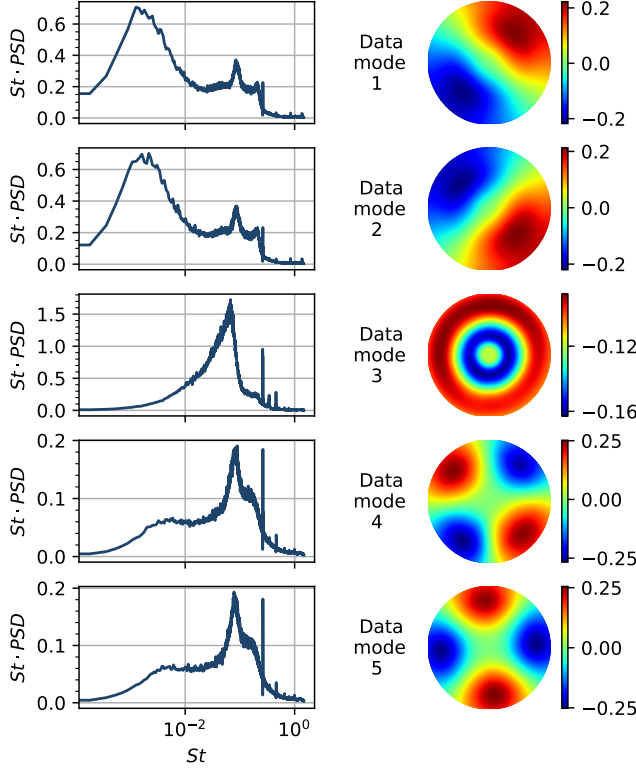


Figure 6: Right: The first five data modes of the wind-tunnel dataset $\Phi_{:,1}^{exp}$ to $\Phi_{:,5}^{exp}$. Left: the premultiplied PSD ($St \cdot PSD$) of their associated time coefficients $A_{1,:}^{exp}$ to $A_{5,:}^{exp}$.

Table 1: Different autoencoder architectures and the training datasets.

Results in section	Autoencoder architecture	Inputs $\mathbf{Y}$	Spatial dimension N^*
6.1	MD-AE (CNN)	$\mathbf{U}$	200×129
6.2	AE (CNN)	$\mathbf{U}$	200×129
7	AE (feedforward)	$\mathbf{P}$	64

In this section, we define the autoencoders for a generic input dataset $\mathbf{Y}$ where the spatial dimension N^* depends on the dataset under investigation (numerical and wind-tunnel datasets) and the autoencoder architectures. The inputs and autoencoder architectures are summarized in Table 1.

4.1. Standard autoencoders

We refer to a standard autoencoder (AE) as an autoencoder with the structure shown in Figure 7, which consists of one encoder F_{en} and one decoder F_{de} . In compact notation, an AE is

$$\begin{aligned} \mathbf{Z}_{:,t} &= F_{en}(\mathbf{Y}_{:,t}; \omega), \\ \hat{\mathbf{Y}}_{:,t} &= F_{de}(\mathbf{Z}_{:,t}; \omega). \end{aligned} \quad (4.4)$$

Details of the layers are given in the Appendix section A (Table 2 to 5). All neural networks that have the convolution encoder and decoder share the same layers and filter size (tables 2 and 3). The feedforward

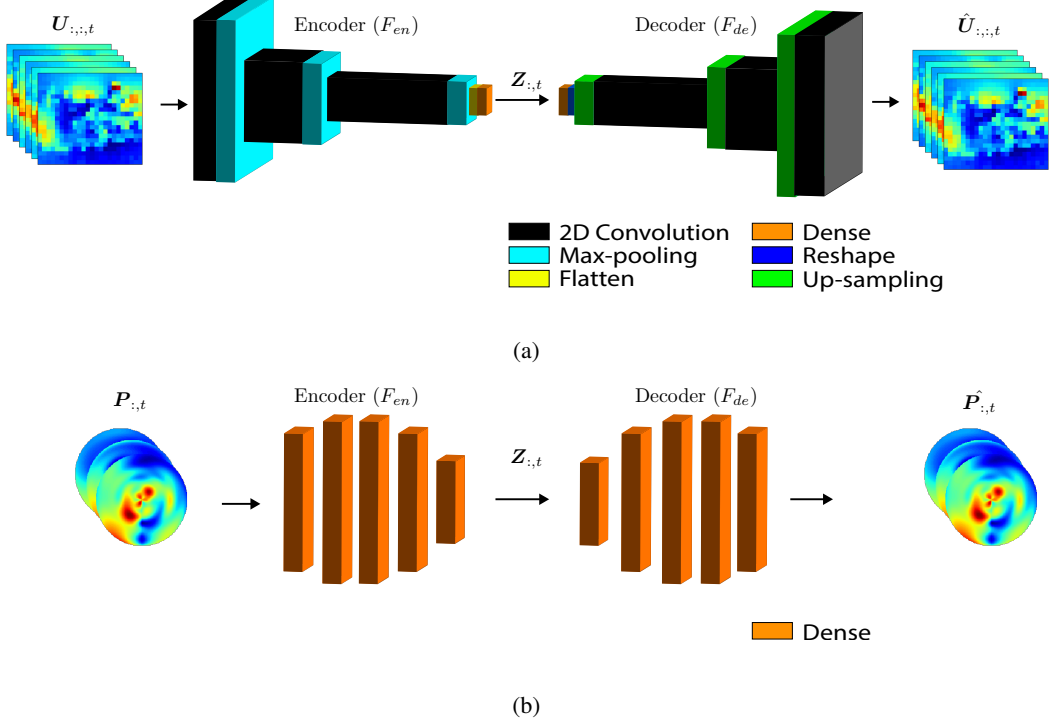


Figure 7: The schematics of the standard autoencoder (AE). Different AEs are employed for different datasets and tests. The AE architecture and dataset for each test are listed in Table 1. a) AE for decomposing the laminar cylinder wake $\mathbf{U}$ with three convolution layers in both the encoder and the decoder. The hyperparameters are in tables 2 and 3 in the Appendix. b) AE for decomposing the wind-tunnel pressure data $\mathbf{P}$ with 5 feedforward layers. The input is a flattened vector of measurement taken from all sensors at time t . The hyperparameters are given in tables 4 and 5 in the Appendix.

AEs have layers given in tables 4 and 5 unless stated otherwise. For all autoencoders, the activation function ‘tanh’ is used. All hyperparameters are given in Table 6.

4.2. Mode-decomposing autoencoders (MD-AE)

An MD-AE associates one decoder to each latent variable (Figure 8), so we can visualize the effect of each latent variable separately, which makes it easier to interpret [24]. The MD-AE is

$$\begin{aligned} \mathbf{Z}_{:,t} &= F_{en}(\mathbf{Y}_{:,t}; \boldsymbol{\omega}), \\ \hat{\mathbf{Y}}_{:,t} &= \sum_{i=1}^{N_z} \underbrace{F_{de}^i(\mathbf{Z}_{:,t}^i; \boldsymbol{\omega})}_{\equiv \mathbf{M}_{:,t}^i} \end{aligned} \quad (4.5)$$

where $\mathbf{M}^i$ denotes the i -th decomposed field of the MD-AE and $\mathbf{M}_{:,t}^i$ is the i -th decomposed field at time step t , corresponding to the output of the i -th decoder. The MD-AE is only applied to the laminar cylinder wake case (the reasoning behind this decision is explained in section 6).

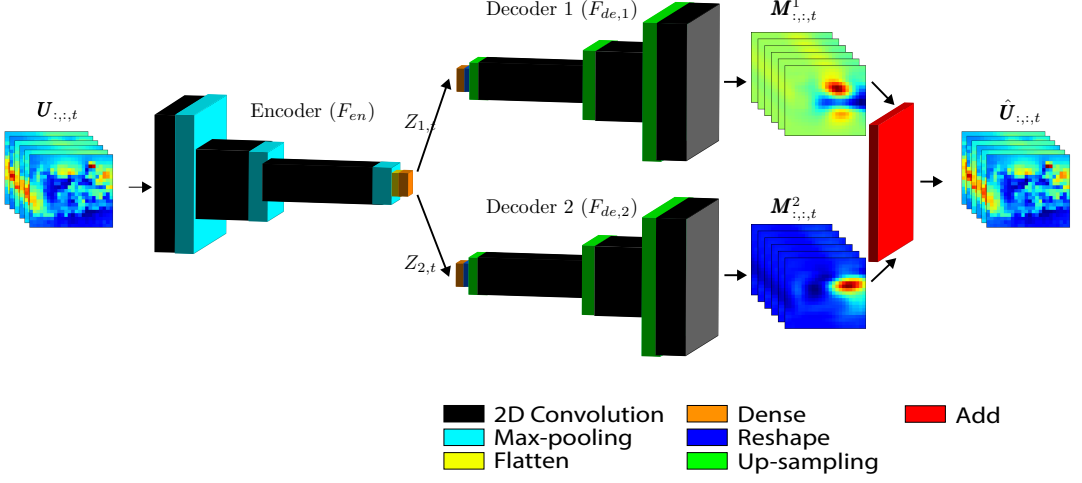


Figure 8: Schematic of the MD-AE [24] with two latent variables as an example. Each latent variable is decoded by a decoder to produce a decomposed field. The sum of the decomposed fields M^1 and M^2 is the output of the MD-AE.

5. The decoder decomposition

The dynamics of POD modes are encapsulated in their time coefficient. To gain physical insight into the decomposed fields, we propose the decoder decomposition to obtain a relationship between the data time coefficients (Eq. (2.2)) and the latent variables trained with the same data. The decoder decomposition is a post-processing method that applies to a trained network. We define the decoder decomposition for AEs in section 5.1 and for MD-AEs in section 5.2.

5.1. The decoder decomposition for standard autoencoders

The data modes Φ^Y (the matrix of POD modes of Y , see Table 1) form a basis of the subspace in which the output $\hat{Y}_{:, t}$ is approximated as

$$\hat{Y}_{:, t} = \sum_{k=1}^{N^*} B_{k,t} \Phi^Y_{:, k} = \Phi^Y B_{:, t}, \quad (5.1)$$

with $B \in \mathbb{R}^{N^* \times N_t}$ being the temporal coefficients (decoder coefficients). The decoder coefficient for the data mode k at time step t is $B_{k,t}$. The matrix form of equation (5.1) is

$$\hat{Y} = \Phi^Y B^T. \quad (5.2)$$

In a trained network, B depends only $\hat{Y}$, which depends only on the latent variables Z . Thus, the gradient of the output of the decoder with respect to the latent variables is

$$\frac{d\hat{Y}}{dZ} = \Phi^Y \frac{dB}{dZ}. \quad (5.3)$$

The gradient of the decoder coefficients is the sensitivity to changes in the latent variables. Each element of the gradient matrix $\left(\frac{d\hat{Y}}{dZ}\right)_{j,i,t}$ is the rate of change of the output at grid point j and time step t ($\hat{Y}_{i,t}$) with respect to the latent variable $Z_{i,t}$. Because a single decoder coefficient corresponds to a single

data mode, which represents a coherent structure, the gradient of the decoder coefficients also reflects the sensitivity of the coherent structure to changes in the latent variables. Physically, the sensitivity quantifies the relative importance of the different latent variables for decoder coefficients and the data modes they represent. We quantify the relative importance of the decoder coefficient $\mathbf{B}_{j,:}$ due to latent variable $\mathbf{Z}_{i,:}$ by defining the average rate of change

$$\epsilon_{i,j} = \frac{\int \left| \frac{\partial \mathbf{B}_{j,:}}{\partial \mathbf{Z}_{i,:}} \right| d\mathbf{Z}_{1,:} \dots d\mathbf{Z}_{N_z}}{\int d\mathbf{Z}_{1,:} \dots d\mathbf{Z}_{N_z}}. \quad (5.4)$$

The average rate of change of a decoder coefficient due to a latent variable i quantifies its contribution to the data mode j in the output associated with that decoder coefficient (Eq. 5.1). The larger the $\epsilon_{i,j}$, the more important the latent variable i is in representing the data mode j in the output. This can be used to rank and select latent variables, as demonstrated in section 7.2. Algorithm 1 shows how the average rate of change is computed.

Algorithm 1: Computing the average rate of change

Input : F_{de} - trained decoder

Δz - constant step size

Output : ϵ - average rate of change

for $i \leftarrow 1, \dots, N_z$ **do**

$\left| \frac{d\hat{\mathbf{Y}}}{d\mathbf{Z}} \right|_{i,:} \leftarrow \frac{1}{\sigma(\mathbf{Z}_{i,:})} \frac{F_{de}(\mathbf{Z}_{i,:} + \Delta z) - F_{de}(\mathbf{Z}_{i,:} - \Delta z)}{2\Delta z}$; // approximate derivatives & normalize

end

Collect $\frac{d\hat{\mathbf{Y}}}{d\mathbf{Z}}$;

$\frac{d\mathbf{B}}{d\mathbf{Z}} \leftarrow \Phi^Y T \frac{d\hat{\mathbf{Y}}}{d\mathbf{Z}}$;

$\epsilon^T \leftarrow \frac{1}{N_t} \sum_{t=1}^{N_t} \left| \frac{d\mathbf{B}}{d\mathbf{Z}} \right|_{:,t}$; // average rate of change

return ϵ

5.2. Decoder decomposition for MD-AEs

In an MD-AE, the decomposed fields $\mathbf{M}^i$ is decomposed into

$$\mathbf{M}^i = \Phi^Y \mathbf{B}^i T, \quad (5.5)$$

where $\mathbf{B}^i$ is the matrix of the i -th decoder coefficients of the i -th decomposed field. Combined with equation (4.5), the output of the MD-AE becomes

$$\hat{\mathbf{Y}} = \Phi^Y \sum_{i=1}^{N_z} \mathbf{B}^i T. \quad (5.6)$$

Equations (5.3) to (5.4) still apply as they are.

6. Decomposition of the unsteady laminar wake

In this section, we analyse the latent space of autoencoders trained with the laminar cylinder wake dataset (section 3.1). We first decompose the dataset with an MD-AE and apply the decoder decomposition in section 6.1, which serves as a benchmark to compare against the literature. We then apply the decoder decomposition to interpret the latent variables of AEs in section 6.2.

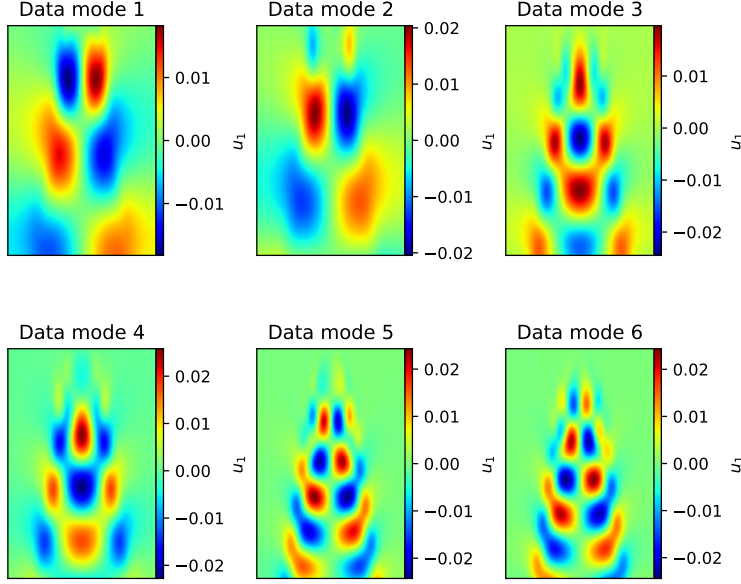


Figure 9: The first six data modes of the unsteady wake behind a cylinder dataset $\Phi_{:,1}^{lam}, \dots, \Phi_{:,6}^{lam}$.

6.1. The latent space of a mode-decomposing autoencoder

We decompose the unsteady wake dataset, described in section 3.1, with an MD-AE with $N_z = 2$ (Figure 10) to verify the decoder decomposition. The reconstruction loss, measured with MSE (Eq. (4.2)), is 2.6×10^{-5} , showing that the output $\hat{\mathbf{U}}$ is an accurate approximation of $\mathbf{U}$. Figure 10a shows that the latent variables have the same periodic behaviour as the data time coefficients, matching the results of [24]. Figure 9 shows the data modes of the laminar dataset. By applying POD to the decomposed fields and by inspection, the decomposed field 1 contains the data modes 1, 3 and 6, and the decomposed field 2 contains the data modes 2, 4 and 5 (Figure 10c). Figure 10b shows that the two latent variables have the same frequency spectrum, with peaks at $St = 0.23$ and 0.69 , corresponding to the vortex shedding frequency and its second harmonic. Because the decomposed field 1 contains a POD mode similar to data mode 1 and the decomposed field 2 contains a similar mode to data mode 2, and the two latent variables have an identical frequency spectrum, are periodic and out of phase, we conclude that the two latent variables form a pair similar to the data time coefficients. However, the latent variables have two frequency peaks instead of a singular peak of the first two data time coefficients, showing that the latent variables contain nonlinear temporal information.

We apply the decoder decomposition to the MD-AE and plot the first four decoder coefficients for the two decomposed fields in Figure 11. None of the first four decoder coefficients are constant in Figure 11, meaning that the first four data modes all contribute toward both decomposed fields. The relative frequencies of the latent variables and the decoder coefficients can also be deduced from Figure 11. As the latent variables change from -1 to 1 , decoder coefficients 1 and 2 in both decomposed fields complete a quarter of a sine wave, but the decoder coefficients 3 and 4 complete half a sine wave, meaning that decoder coefficients 3 and 4 oscillate at double the frequency of decoder coefficients 1 and 2.

For a more quantitative measure of how the data modes are split into the decomposed fields of the MD-AE, we calculate the equivalent energy [16] for each decomposed field separately. The matrix of

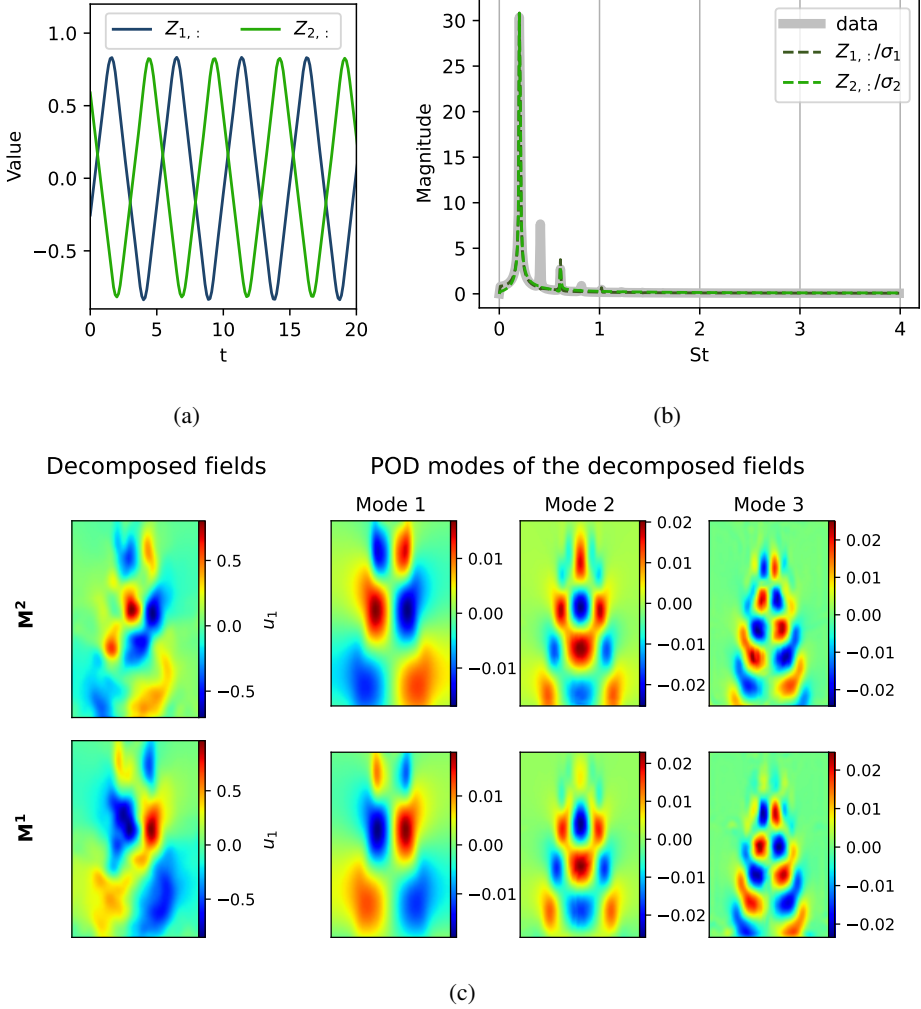


Figure 10: Results of training the MD-AE on the laminar wake dataset with a latent dimension of 2. a) Both latent variables are periodic in time. b) Frequency spectrum of the latent variables, normalised by their standard deviation, compared with the frequency of the data. The latent variables both contain the vortex shedding frequency and the second harmonic of the vortex shedding frequencies. c) POD modes of the decomposed fields. The POD modes of decomposed field 1 are similar to data mode 1,3 and 6 and the POD modes of decomposed field 2 are similar to data mode 2,4 and 5.

equivalent energy for the i -th decomposed field and the data modes of the laminar flow, $\hat{\mathbf{A}}^i$, is given by

$$\hat{\mathbf{A}}^i = \frac{1}{N_t - 1} \mathbf{B}^i \mathbf{B}^{iT}, \quad (6.1)$$

which is interpreted as the variance of the i -th decomposed field, $\mathbf{M}^i$, projected onto the data modes. The equivalent energy of the i -th decomposed field and the k -th data mode is $\hat{\Lambda}_{kk}^i$. As $\mathbf{M}^i$ approaches the first data POD decomposed field ($\Phi_{:,1}^{lam} \mathbf{A}_{1,:}^T$), $\hat{\Lambda}_{11}^i$ approaches Λ_{11} while $\hat{\Lambda}_{kk}^i$, where $k \neq 1$, approaches 0.

Figure 12 shows the equivalent energy of the first six data modes in the two decomposed fields and in the predicted output $\hat{\mathbf{U}}$. The equivalent energies of the predicted output for all data modes are

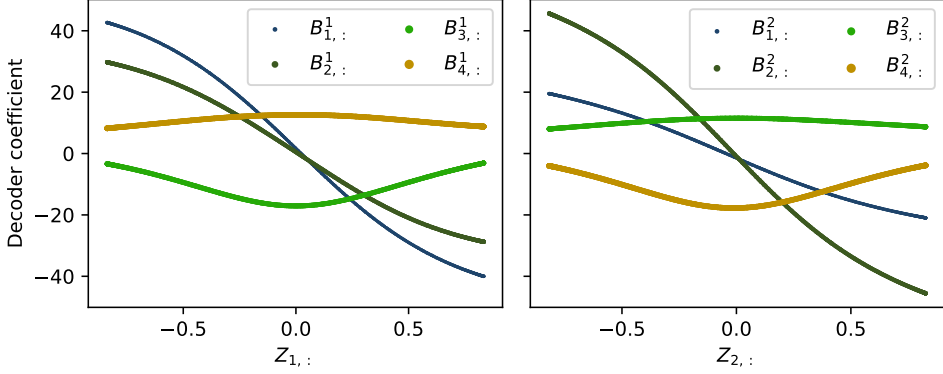


Figure 11: Results from the MD-AE trained with the laminar wake dataset with 2 latent variables. The first four decoder coefficients plotted against the latent variables. Both latent variables affect the magnitude of the first four POD modes of the data in the output of the MD-AE.

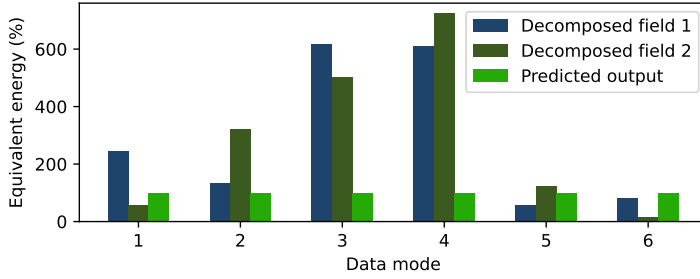


Figure 12: Results from the MD-AE trained with the laminar wake dataset with 2 latent variables. The equivalent energy of the two decomposed fields of the MD-AE, showing the first four POD modes of the data. All POD modes of data are present in both decomposed fields.

approximately 100%, meaning that the output $\hat{\mathbf{U}}$ is an accurate approximation of $\mathbf{U}$. which also shows that the MD-AE does not fully separate data modes into different decomposed fields.

In summary, the decoder decomposition applied to an MD-AE [24] leads to a detailed interpretation of the decomposed fields. By computing the equivalent energy, we also show that, although the MD-AE works well for visualisation, the decomposed fields are entangled.

6.2. The latent space of standard autoencoders

Unlike the MD-AE, an AE has no decomposed fields to visualize. We apply the decoder decomposition to AEs trained with the unsteady wake dataset to connect the data time coefficients to the latent variables. Since the frequency content of the laminar, periodic wake of a cylinder and the data modes are well known [22], we use the interpretation of the data modes and time coefficients to provide an interpretation of the latent variables.

Figure 13 shows the loss of AEs trained with the unsteady wake dataset using 1, 2, 3 and 4 latent variables. The first plateau of loss occurs at $N_z = 2$ before the loss decreases again at $N_z = 4$. Similar behaviour is also observed by [7]. The AEs with two latent variables achieve a significantly smaller loss than AEs with a single latent variable but the difference in loss between using two and three latent variables is small. The laminar wake $\mathbf{U}$ has periodic underlying dynamics and thus can be described by

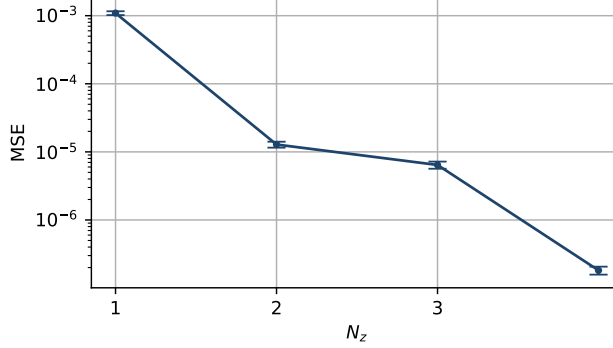


Figure 13: The MSE of AEs trained with the unsteady wake dataset with different numbers of latent variables averaged over five repeats each. The error bars represent the standard deviation of the repeats. At $N_z = 2$ the MSE is approximately 1.3×10^{-5} .

a single variable in polar coordinates (the angle) or two variables in Cartesian coordinates. However, the AEs do not find the best representation when forced to use only one latent variable due to their inability to perform Cartesian to polar transformation. Polar to Cartesian relies on the $\sin$ function, which is a many-to-one function and can be learned by neural networks either through approximation with any activation function or by using $\sin$ as the activation function [38]. However, the Cartesian to polar transformation relies on $\arcsin$, which is not defined for all inputs. The Cartesian to polar transformation is a known difficulty in machine learning, with benchmark problems such as the two-spirals problem built around it [11, 18], this is consistent with our observations (Figure 13). Therefore, we analyse an AE with $N_z = 2$ with decoder decomposition to obtain physical interpretation in the current section, even though $N_z = 1$ should be the theoretical minimum representation of the laminar wake.

The phase plot of the two latent variables is a unit circle (Figure 14a), which indicates that the latent variables of an AE are periodic; the same phase plot also describes the first two data time coefficients. The same periodic behaviour also shows in the time trace of the latent variables (Figure 14b). Figure 14c compares the frequency spectrum of the latent variables and the data. We find that both latent variables oscillate at the vortex shedding frequency, which is the lowest frequency in the dataset (Figure 2, right panel). The higher harmonic frequencies are not included in the latent variables because they are functions of the vortex-shedding frequencies, and the decoders can represent the spatial patterns of multiple data modes. The AE has produced latent variables which have a singular peak in the frequency spectrum, less than the frequency spectrum of the latent variables of the MD-AE, without sacrificing accuracy, showing that AEs are more expressive given the same number of latent variables than MD-AEs.

Figure 15 shows the sensitivity of the decoder coefficients, $\mathbf{B}_{1,:}, \dots, \mathbf{B}_{4,:}$ to the two latent variables. The contours are the values of the decoder coefficients, shown for all values of the latent variables allowed by the nonlinear activation function. The dynamics of the dataset are shown as the grey circle, which shows the values of the latent variables that are observed during training.

The decoder coefficients $\mathbf{B}_{3,:}$ and $\mathbf{B}_{4,:}$ have double the frequency of $\mathbf{B}_{1,:}$ and $\mathbf{B}_{2,:}$, consistent with the characteristics of the first four data modes. At $\mathbf{Z} = \mathbf{0}$, all decoder coefficients have the value 0 because the network has no bias. Figure 15 provides a qualitative overview of the sensitivity of the data modes to the latent variables.

Because the decoder output $\hat{\mathbf{U}}$ depends on multiple latent variables, the equivalent energy (Eq. (6.1)) cannot be used to identify the contribution of each latent variable. We use the average rate of change (Eq. (5.4)) to quantify the contribution of individual latent variables of the AE (Figure 16). Amongst the first four data modes, latent variable $\mathbf{Z}_{1,:}$ contributes most to the presence of the second data mode $\Phi_{:,2}^{lam}$ in the output $\hat{\mathbf{U}}$, while $\mathbf{Z}_{2,:}$ mainly contributes towards the presence of the other three data modes.

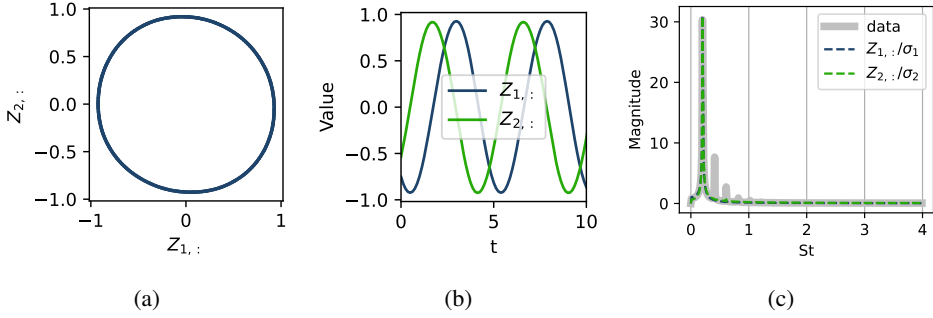


Figure 14: Results from the AE trained with the laminar wake dataset with two latent variables. a) Phase plot of two latent variables. The unit circle indicates harmonic oscillations. b) Time trace of the two latent variables. The latent variables are periodic and 90° out of phase, behaving the same as the first two data time coefficients. c) The frequency spectrum of the latent variables, compared to the frequency in the dataset. The variables are normalized by their standard deviation, σ , before applying the Fourier Transform for visualization.

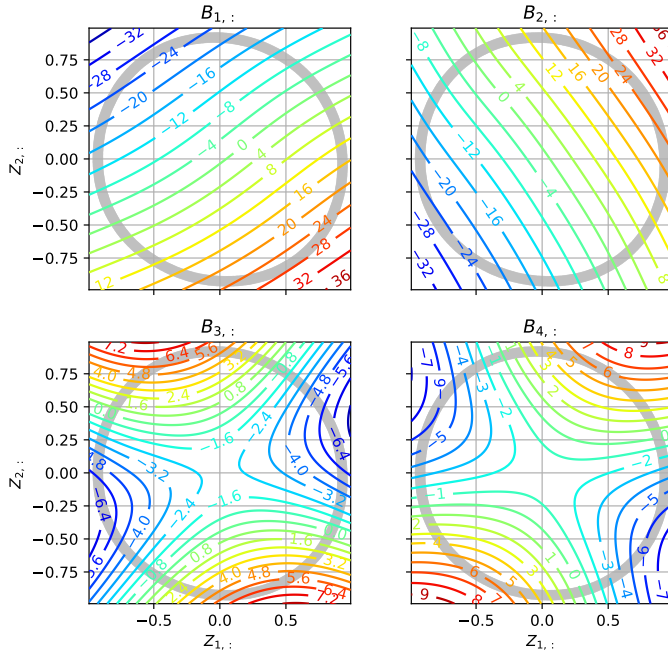


Figure 15: Results from the AE trained with the laminar wake dataset with two latent variables. Contour plot of decoder coefficients of an AE with two latent variables, trained with the unsteady wake dataset. The contours show the values of the decoder coefficients. The grey circle labels the values of the latent variables observed during training, which shows the dynamics of the dataset.

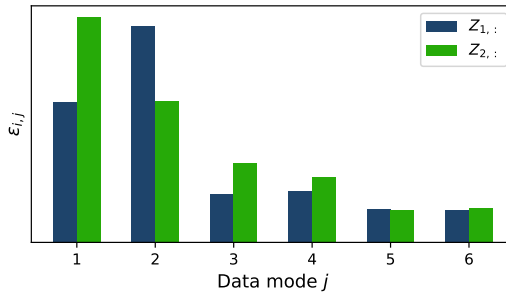


Figure 16: Results from the AE trained with the laminar wake dataset with 2 latent variables. The average rate of change of the decoder coefficients.

The first 6 data modes depend on both latent variables, so no latent variable can be removed without affecting the dynamics of the output. However, the average rate of change suggests that the smaller-scale energy-containing structures between the vortex streets [22], namely the pair comprised of data mode 3 and 4, are mainly represented by $Z_{2,:}$. Therefore, $Z_{2,:}$ should be used if the representation of data modes 3 and 4 in the output is required.

By comparing the frequency spectrum of the latent variables of the AE and the MD-AE (section 6.1), both with $N_z = 2$, we find that the AE has learned a more accurate representation of the data than the MD-AE because the latent variables have the minimum number of frequency peaks needed to represent the data. The frequency of the two latent variables coincides with the lowest frequency in the frequency spectrum of the data, meaning that the data modes with higher frequencies are added by the decoder. By introducing the average rate of change, we show that both latent variables are equally important in representing the vortex shedding, but the first harmonic is contained mainly in one of the latent variables in this case. Because the training of an AE depends on random seeds, the physical meaning of the latent variables may change for different cases. During training, we noticed that the dimension of latent space has a significant impact on the physical interpretation of the latent variables. In-depth comments with example AEs trained with $N_z = 1$ and $N_z = 3$ are found in Appendix C.

7. Autoencoder decomposition of the turbulent wake from wind-tunnel experiments

We analyse the AEs to decompose the wind-tunnel wake dataset $\mathbf{P}$ (section 3.2). We show that, for a turbulent dataset, the size of the network becomes an additional limiting factor for the interpretability of AEs in section 7.1. We demonstrate how to select latent variables to single out a particular flow structure in section 7.2.

7.1. The limiting factor is network size for turbulence

At least two latent variables are needed to represent flows with periodicity (section 6.2). Because the turbulent wake of the axisymmetric body contains periodic behaviours such as vortex shedding [30], we first train an AE with two latent variables on the wind-tunnel dataset (feedforward, see Table 1). Figure 17 (left, centre panels) shows the RMS pressure and the frequency content of the predicted pressure. The frequency peaks of the reconstructed pressure field match the data, but the RMS pressure and the magnitude of the PSD differ from the reference pressure field. We investigate the effect of increasing the latent variables from two to three. The prediction made by the AE with $N_z = 3$ is more accurate in both the RMS pressure and frequency content (Figure 18, left panel, centre panels) compared to the prediction by the AE with $N_z = 2$. By increasing the number of the latent variables from two to three, the prominent frequencies of the prediction remain unchanged, but the magnitude is

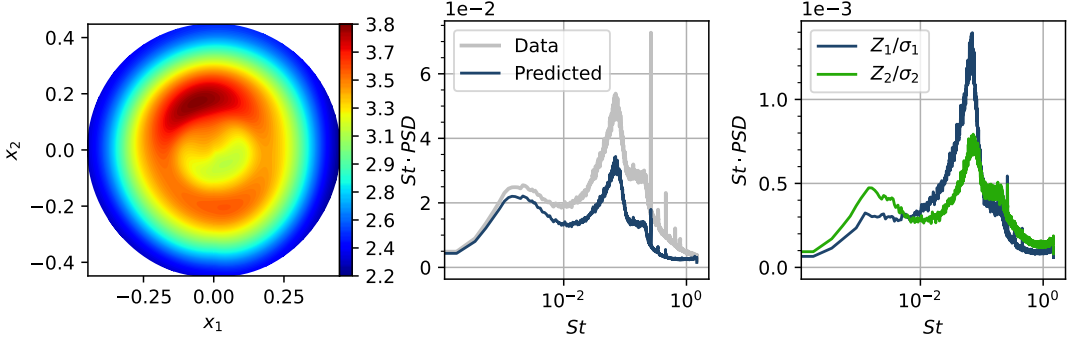


Figure 17: The AE trained with the wind-tunnel dataset with two latent variables. Left: predicted RMS pressure. Centre: the premultiplied overall PSD of the prediction. Right: the premultiplied PSD of the latent variables of the AE trained with two latent variables, normalized by their standard deviation.

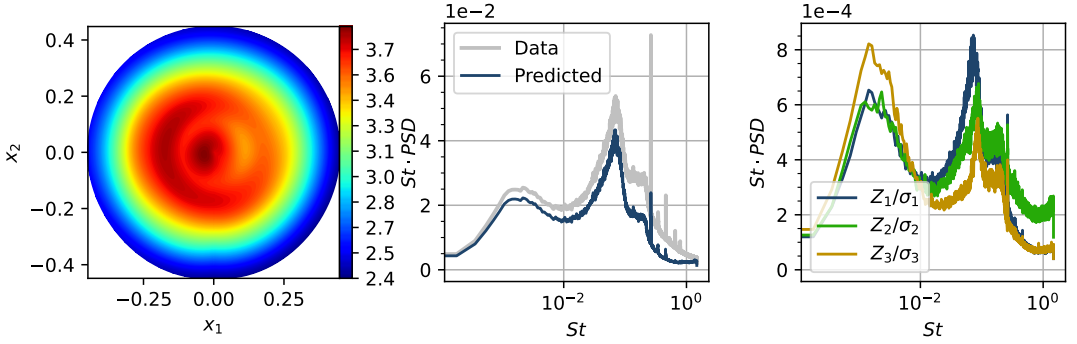


Figure 18: Same as Figure 17 with three latent variables.

closer to the reference data. By increasing the dimension of the latent space from $N_z = 2$ to $N_z = 3$, the prediction has improved without changing any frequency-related behaviour. This suggests that the benefit of increasing the latent dimension comes from increasing the size of the decoder, thus improving the decoder's ability to express spatial patterns.

To further understand the cause of the improved prediction, we train a large decoder-only network (Figure 19) with the latent variables obtained with the previous $N_z = 2$ AE. (The details of the large decoder are given in Table 7.) The large decoder has approximately twice the number of trainable parameters than the decoder in the previous $N_z = 2$ AE. The input to the large decoder is the latent variables obtained from the trained $N_z = 2$ AE, which have already been analysed in Figure 17 (right panel); the output of the large decoder approximates the reference fluctuating pressure $\mathbf{P}$.

The PSD of the large decoder-only network overlaps with the PSD of the $N_z = 3$ smaller-sized AE (Figure 20), both are closer to the PSD of the reference data than the smaller $N_z = 2$ AE. When the latent variables remain unchanged, the large decoder-only $N_z = 2$ network can make predictions that match the predictions made by a smaller dimension $N_z = 3$ AE. By comparing the results of the smaller size $N_z = 2$ and $N_z = 3$ AEs to the large decoder-only network with identical latent variable to the $N_z = 2$ AE, we find that using a larger decoder has the same effect as increasing the number of latent variables. More importantly, the frequency peaks in the latent variables and predictions remain the same for all three networks. Therefore, we conclude that the critical factor for an accurate prediction using the wind-tunnel dataset, which contains multiple spatiotemporal scales, is the size of the decoder. In an AE, the weights are responsible for representing the spatial patterns in the flow, and the latent variables

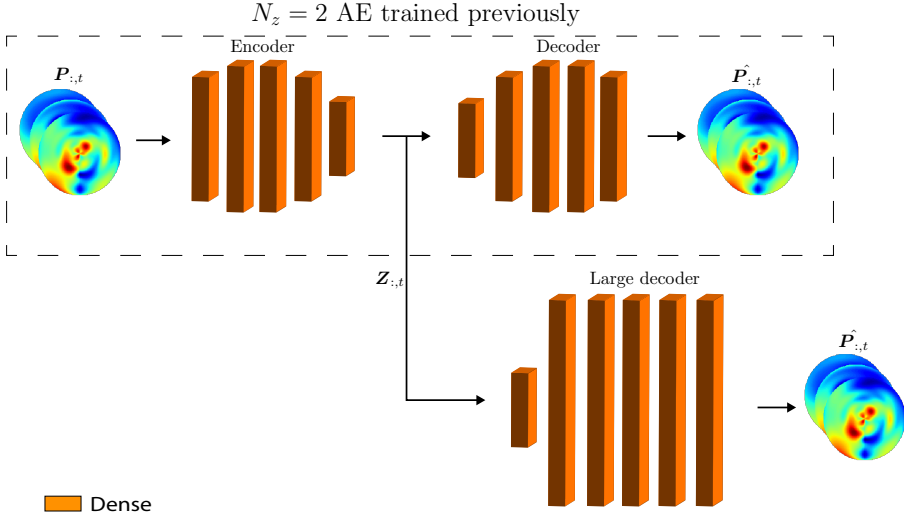


Figure 19: The schematics and training process of the decoder-only network. The decoder has 99% more trainable parameters than the decoder in the $N_z = 2$ AE discussed in section 7.1.

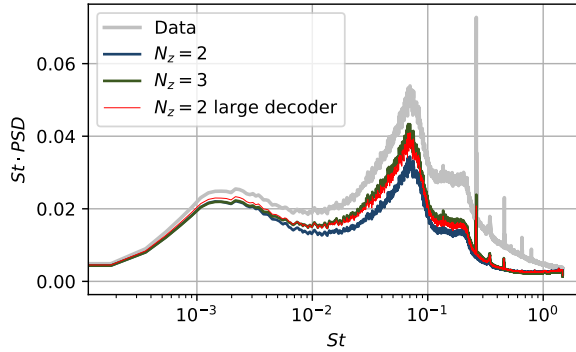


Figure 20: The premultiplied PSD of the predicted pressure from the AE with two and three latent variables, and from the large decoder-only network, trained with the wind-tunnel dataset.

represent the time-dependent behaviours. We found that the optimal dimension of the latent space is a function of both of the underlying manifold's dimension [23, 9, 27] and the size of the decoder.

7.2. Filtering unwanted latent variables

Section 7.1 shows that changing both the dimension of the latent space, N_z , and the size of the AEs have a similar effect for the turbulent datasets. In this section, we apply the proposed decoder decomposition to filter out unwanted latent variables. When training on the turbulent pressure dataset, $\mathbf{P}$, we find that the MSE is sufficiently small at $N_z \approx 28$ (Figure 21). Therefore, we employ the AE with 28 latent variables. We filter the latent variables to enhance or reduce the importance of certain data modes among other data modes in the output of the AEs, similar to selecting data modes for reconstructing only a particular coherent structure. Figure 22 shows the results from the AE with 28 latent variables. The RMS predicted pressure is in agreement with the reference RMS pressure, and the prediction has PSD that approximates the data's PSD in terms of both magnitude and frequency content. To understand

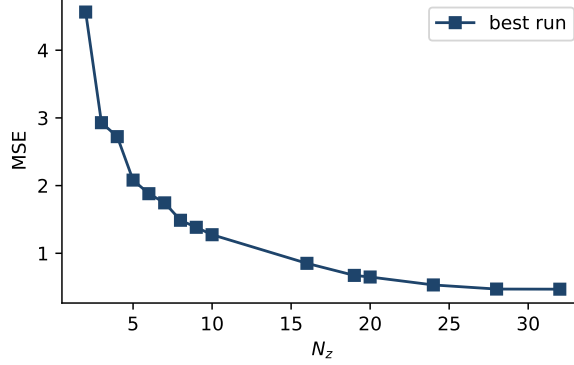


Figure 21: AE with different numbers of latent variables and the same hyperparameters, trained with the wind-tunnel dataset. The loss stops decreasing at $N_z \approx 28$.

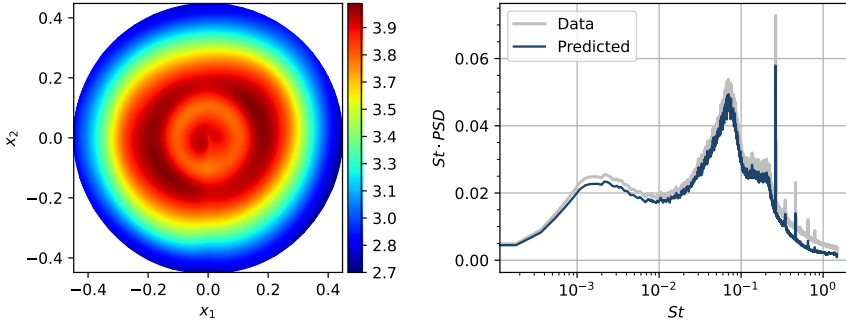


Figure 22: RMS predicted pressure and premultiplied PSD of the prediction of the AE trained with the wind-tunnel dataset with 28 latent variables. The network attains an MSE equivalent to the reconstruction with 30 POD modes.

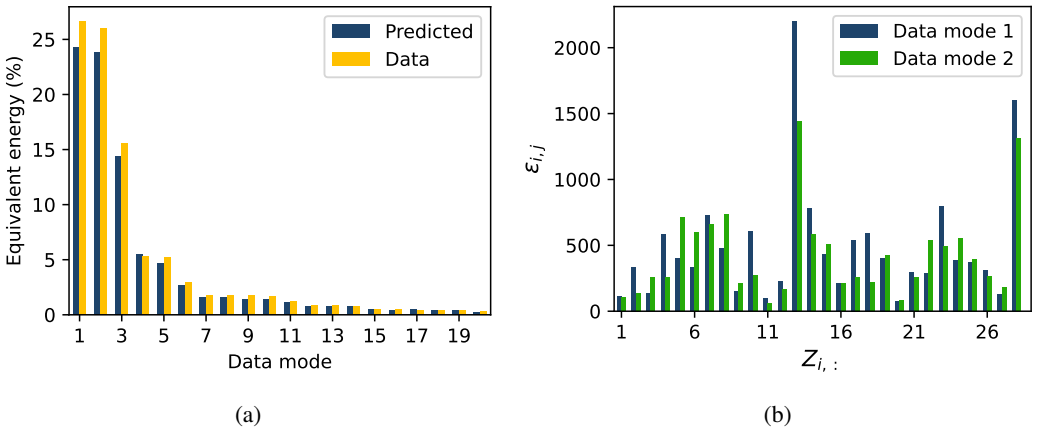


Figure 23: The AE trained with the wind-tunnel dataset with 28 latent variables. a) Equivalent energy of the predicted pressure compared to the POD eigenvalues of the reference data. b) The average rate of change of decoder coefficients 1 and 2 due to latent variables, normalized by the standard deviation of the latent variables. The decoder coefficients 1 and 2 are direct analogies of the data POD time coefficients 1 and 2, which are associated with vortex shedding.

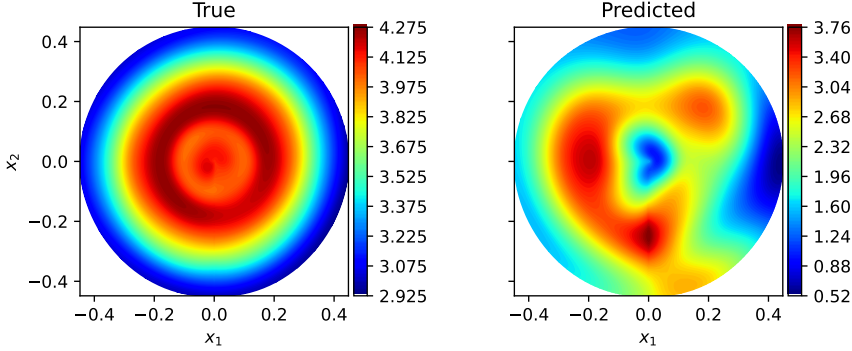


Figure 24: Prediction obtained with the AE with 28 latent variables from the filtered latent variables $\mathbf{Z}_f$ and the reference data.

which data modes are present in the prediction, we define the equivalent energy for an AE with a POD weight matrix, $\mathbf{W}_p$ (Eq. 2.1), as

$$\hat{\Lambda} = \frac{1}{N_t - 1} \Phi^{YT} \left((\mathbf{W}_p \Phi^Y \mathbf{B}^T) (\mathbf{B} \Phi^{YT} \mathbf{W}_p^T) \right) \Phi^Y \quad (7.1)$$

from equation 6.1 [16]. The matrix $\Phi^Y = \Phi^{exp}$ contains the data modes. Figure 23a compares the equivalent energy of the predicted pressure from the AE with $N_z = 28$ with the POD eigenvalues of the reference data. The prediction contains a similar amount of energy to the reference data for the more energetic coherent structures identified by POD. We focus our analysis on data modes 1 and 2, which represent vortex shedding. Figure 23b shows how each latent variable affects the decoder coefficients 1 and 2 in the prediction, which are associated with POD modes 1 and 2 in the prediction. We compute the average rate of change $\epsilon_{i,j}$ (equation (5.4)) and focus the analysis on data modes 1 and 2, which represent vortex shedding. Figure 23b shows how each latent variable affects the decoder coefficients 1 and 2 in the prediction, which are associated with POD modes 1 and 2 in the prediction. For both data modes, the largest contribution comes from the latent variable 13, followed by the latent variable 28. The numbering changes with the seeds and gradient update algorithms, but the conclusions do not. For the output of the AE to represent mainly the data mode 1 and 2, we set all latent variables (except for $\mathbf{Z}_{13,:}$ and $\mathbf{Z}_{28,:}$) to $\mathbf{0}$. This yields the filtered latent space $\mathbf{Z}_f$ such that

$$\hat{\mathbf{P}}_f = F_{de}(\mathbf{Z}_f), \quad (7.2)$$

which contains mostly data modes 1 and 2. The results of decoding the filtered latent variables are shown in Figure 24. The RMS of the filtered prediction shows strong fluctuations only in the outer region. The PSD of the filtered prediction (Figure 25) shows that $\hat{\mathbf{P}}_f$ has a large amount of energy at $St \approx 0.2$ and $St \approx 0.002$. The frequency $St \approx 0.06$ does not appear in the PSD of the filtered prediction, meaning that by keeping only the two most contributing latent variables for data modes 1 and 2, the AE no longer models the pulsation of the vortex core. This means that the filtering has been successful.

The equivalent energy in Figure 26a further shows that the filtered prediction represents only the data modes 1 and 2, which is the objective that we set. Figure 26b shows the two leading POD modes of the filtered prediction. The modes are approximately antisymmetric with frequency peaks at $St \approx 0.2$ and $St \approx 0.002$, which represent vortex shedding. The structures in Figure 26b are not perfectly antisymmetric due to information being lost by removing 26 out of 28 latent variables. The two leading modes contain over 99% of the energy in the filtered prediction, showing that the filtering has significantly compressed the spatial information. Combining the energy, the structures and frequency content, we conclude that the only coherent structures in the filtered prediction are modes associated

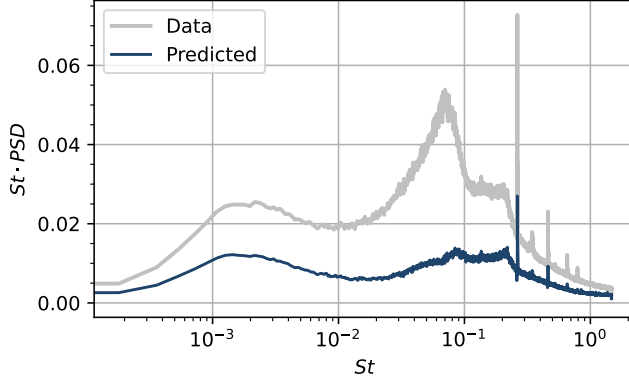


Figure 25: Premultiplied PSD of the prediction obtained with the AE with 28 latent variables from the filtered latent variables. All peaks (except for $St \approx 0.2$) are filtered.

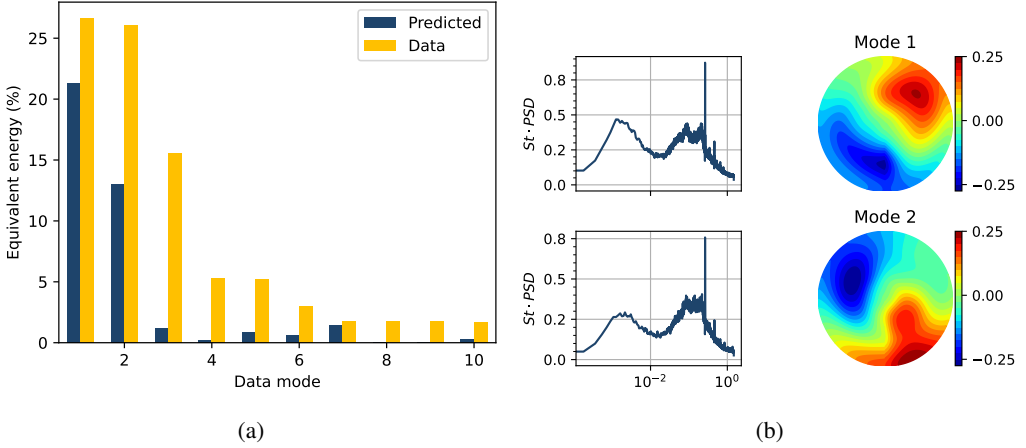


Figure 26: POD on the prediction obtained with the AE with 28 latent variables from the filtered latent variables. a) Equivalent energy of the filtered prediction $\hat{\mathbf{P}}_f$ compared to the POD eigenvalues of the reference data. Amongst the five most energetic POD modes, the filtered prediction shows a large amount of energy only in data modes 1 and 2, which is the goal we set. b) POD modes 1 and 2 of $\hat{\mathbf{P}}_f$, which contain over 99% of the flow energy of $\hat{\mathbf{P}}_f$. These two modes show the same frequency peak at $St \approx 0.2$ and have anti-symmetric spatial structures. These two modes represent vortex shedding.

with vortex shedding. By selecting latent variables based on contribution to the decoder coefficients, we have successfully filtered the coherent structures of the output of the AE. Thus, we have shown decoder decomposition as a viable method for selecting latent variables based on coherent structures of the flow for an AE with a large latent dimension. This is useful for filtering spurious or unwanted latent variables, as well as singling out specific modes of interest.

8. Conclusions

We propose the decoder decomposition to help practitioners design and interpret the latent space of nonlinear autoencoders. The decoder decomposition is a post-processing method, which ranks the latent variables of an autoencoder. Two types of autoencoders are considered: the standard autoencoders

(AE) with one decoder and one encoder, and the mode-decomposing autoencoders (MD-AEs) with one decoder per latent variable. First, we apply the decoder decomposition to analyse the latent space of the two-dimensional unsteady laminar cylinder wake. Both AEs and MD-AEs are applied to the laminar wake and their results are compared. By analysing different dimensions of the latent space, we show that the dimension of the latent space significantly impacts the physical meanings of the latent variables even if the reconstruction errors remains unaffected. Second, we apply the decoder decomposition to AEs trained with wind-tunnel experimental data of a three-dimensional turbulent wake past a bluff body. We show that increasing the size of the decoder has a similar effect to increasing the number of latent variables when the number of latent variables is small. Third, we apply the decoder decomposition to rank and select the latent variables that are associated with the vortex shedding structures. We apply the average rate of change to rank the latent variables based on their contribution to the data modes 1 and 2, which correspond to vortex shedding. Finally, we filter the latent space to eliminate the unwanted modes. The output contains only the two coherent structures associated with vortex shedding, which verifies the method. The decoder decomposition is a simple, yet robust, method for the ranking and selection of latent variables based on the flow physics and coherent structures of interest.

Acknowledgments. We thank G. Rigas and J. Morrison for providing the experimental data in section 3.2.

Funding Statement. We acknowledge funding from the Engineering and Physical Sciences Research Council, UK and financial support from the ERC Starting Grant PhyCo 949388. L.M. is also grateful for the support from the grant EU-PNRR YoungResearcher TWIN ERC-PI_0000005.

Competing Interests. None.

Data Availability Statement. Codes and data are accessible via GitHub: <https://github.com/MagriLab/MD-CNN-AE>.

Ethical Standards. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

Author Contributions. Y. Mo collected the numerical data, performed the analysis and wrote the paper. T. Traverso helped with the analysis and editing of the paper. L. Magri conceived the objectives and methods, and wrote and edited the paper.

References

- [1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- [2] Albuqrek, C. M., Urban, K., Rempfer, D., and Lumley, J. L. (2002). Divergence-free wavelet analysis of turbulent flows. *Journal of scientific computing*, 17(1-4):49–66.
- [3] Alfonsi, G. and Primavera, L. (2007). The structure of turbulent boundary layers in the wall region of plane channel flow. *Proceedings of the Royal Society. A, Mathematical, physical, and engineering sciences*, 463(2078):593–612.
- [4] Baldi, P. and Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2:53–58.
- [5] Bansal, N., Chen, X., and Wang, Z. (2018). Can We Gain More from Orthogonality Regularizations in Training Deep Networks? In *Advances in Neural Information Processing Systems*, volume 31, pages 4261–4271. Curran Associates, Inc.
- [6] Bartholomew, P., slaizet, Schuch, F. N., Frantz, R. A. S., rfj82982, Hamzehloo, D. A., CFLAG, Deskos, G. Y., nbeb, KaySchaefer, nasos94, Jing, H., vc218385, and Monteiro, L. R. (2022). Xcompact3d/Incompact3d:. Zenodo.
- [7] Csala, H., Dawson, S. T. M., and Arzani, A. (2022). Comparing different nonlinear dimensionality reduction techniques for data-driven unsteady fluid flow modeling. *Physics of Fluids*, 34(11):117119.
- [8] Doan, N. A. K., Polifke, W., and Magri, L. (2021). Auto-encoded reservoir computing for turbulence learning. In *Computational Science – ICCS 2021*, pages 344–351, Cham. Springer International Publishing.
- [9] Doan, N. A. K., Racca, A., and Magri, L. (2023). Convolutional Autoencoder for the Spatiotemporal Latent Representation of Turbulence. In Mikyška, J., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V. V., Dongarra, J. J., and Sloot, P. M., editors, *Computational Science – ICCS 2023*, Lecture Notes in Computer Science, pages 328–335, Cham. Springer Nature Switzerland.
- [10] Eivazi, H., Clainche, S. L., Hoyas, S., and Vinuesa, R. (2022). Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows. *Expert Systems with Applications*.

- [11] Fahlman, S. and Lebiere, C. (1989). The Cascade-Correlation Learning Architecture. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.
- [12] Fukami, K., Hasegawa, K., Nakamura, T., Morimoto, M., and Fukagata, K. (2021). Model order reduction with neural networks: Application to laminar and turbulent flows. *SN computer science*, 2(6).
- [13] Fukami, K., Nakamura, T., and Fukagata, K. (2020). Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data. *Physics of Fluids*, 32(9):095110.
- [14] Ganga Prasath, S., Sudharsan, M., Vinodh Kumar, V., Diw akar, S., Sundararajan, T., and Tiwari, S. (2014). Effects of aspect ratio and orientation on the wake characteristics of low Reynolds number flow over a triangular prism. *Journal of Fluids and Structures*, 46:59–76.
- [15] Kevlahan, N. K. R., Hunt, J. C. R., and Vassilicos, J. C. (1994). A comparison of different analytical techniques for identifying structures in turbulence. *Applied scientific research*, 53(3-4):339–355.
- [16] Kneer, S., Sayadi, T., Sipp, D., Schmid, P., and Rigas, G. (2021). Symmetry-aware autoencoders: s-pca and s-nlpca. *ArXiv*.
- [17] Laizet, S. and Lamballais, E. (2009). High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy. *Journal of Computational Physics*, 228(16):5989–6015.
- [18] Liu, R., Lehman, J., Molino, P., Petroski Such, F., Frank, E., Sergeev, A., and Yosinski, J. (2018). An intriguing failing of convolutional neural networks and the CoordConv solution. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- [19] Loiseau, J.-C., Brunton, S. L., and Noack, B. R. (2020). *From the POD-Galerkin method to sparse manifold models*, pages 279–320. Model Order Reduction: Volume 3 Applications. De Gruyter.
- [20] Loshchilov, I. and Hutter, F. (2017). SGDR: Stochastic Gradient Descent with Warm Restarts.
- [21] Lumley, J. L. (1970). Stochastic tools in turbulence. In *Stochastic Tools in Turbulence*, Applied Mathematics and Mechanics, v. 12. Academic Press, New York.
- [22] M. M. Zdravkovich (1997). *Flow Around Circular Cylinders Vol 1: Fundamentals*, volume 1. Oxford University Press, Oxford, 1 edition.
- [23] Magri, L. and Doan, N.A.K (2022). On interpretability and proper latent decomposition of autoencoders. *Center for Turbulence Research Proceedings of the Summer Program 2022*.
- [24] Murata, T., Fukami, K., and Fukagata, K. (2020). Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of fluid mechanics*, 882.
- [25] Nakamura, T., Fukami, K., Hasegawa, K., Nabae, Y., and Fukagata, K. (2020). Cnn-ae/lstm based turbulent flow forecast on low-dimensional latent space.
- [26] Noack, B. R., Morzynski, M., Tadmor, G., and Tadmor, G. (2011). *Reduced-Order Modelling for Flow Control*, volume 528. Springer Wien, Vienna, Austria.
- [27] Racca, A., Doan, N. A. K., and Magri, L. (2023). Predicting turbulent dynamics with the convolutional autoencoder echo state network. *Journal of Fluid Mechanics*, 975:A2.
- [28] Rigas, G. (2021). Turbulent experimental database I. Internal Document.
- [29] Rigas, G., Oxlade, A. R., Morgans, A. S., and Morrison, J. F. (2014). Low-dimensional dynamics of a turbulent axisymmetric wake. *Journal of Fluid Mechanics*, 755.
- [30] Rowan D. Brackston (2017). *Feedback Control of Three-Dimensional Bluff Body Wakes for Efficient Drag Reduction*. PhD thesis, Imperial College London, London.
- [31] Rowley, C. W. and Dawson, S. T. M. (2017). Model Reduction for Flow Analysis and Control. *Annual Review of Fluid Mechanics*, 49(Volume 49, 2017):387–417.
- [32] Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656(August):5–28.
- [33] Schmidt, O. T. and Colonius, T. (2020). Guide to spectral proper orthogonal decomposition. *AIAA Journal*, 58(3):1023–1033.
- [34] Taira, K., Brunton, S. L., Dawson, S. T. M., Rowley, C. W., Colonius, T., McKeon, B. J., Schmidt, O. T., Gordeyev, S., Theofilis, V., and Ukeiley, L. S. (2017). Modal analysis of fluid flows: An overview. *AIAA journal*, 55(12):4013–4041.
- [35] Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., and Kutz, J. N. (2014). On dynamic mode decomposition: Theory and applications. *Journal of computational dynamics*, 1(2):391–421.
- [36] Vinuesa, R. and Sirmacek, B. (2021). Interpretable deep-learning models to help achieve the sustainable development goals. *Nature Machine Intelligence*, 3(11):926.
- [37] Weiss, J. (2019). A tutorial on the proper orthogonal decomposition. In *2019 AIAA Aviation Forum*, Dallas, Texas, United States. American Institute of Aeronautics and Astronautics.
- [38] Wong, J. C., Ooi, C., Gupta, A., and Ong, Y.-S. (2022). Learning in Sinusoidal Spaces with Physics-Informed Neural Networks. *IEEE Transactions on Artificial Intelligence*, pages 1–15.
- [39] Xie, D., Xiong, J., and Pu, S. (2017). All You Need is Beyond a Good Init: Exploring Better Solution for Training Extremely Deep Convolutional Neural Networks with Orthonormality and Modulation.

Appendices

A. Autoencoder layers

The convolutional autoencoders are trained with the laminar wake dataset. All convolutional AEs and MD-AEs are built with encoders described in Table 2 and decoders described in Table 3. The encoders are identical except for the latent dimension N_z . For all convolutional AEs, the decoders are identical except for N_z . For all convolutional MD-AEs, the input to the decoder has shape (1), specified in Table 3. When training on the wind-tunnel dataset, AEs with feedforward layers are used (tables 4 and 5). The large decoder-only network (Table 7) is only used in section 7.1. The hyperparameters used by all networks can be found in Table 6.

Table 2: The encoder used in all convolutional AEs and MD-AEs.

Layers	Output shape	Notes
input	(200,129,1)	
conv2d	(200,129,8)	3-by-3 filter
maxpooling	(100,65,8)	
conv2d	(100,65,16)	3-by-3 filter
maxpooling	(50,33,16)	
conv2d	(50,33,32)	3-by-3 filter
maxpooling	(25,17,32)	
flatten	(13600)	
dense	(N_z)	output shape is (1) if MD-AE

Table 3: The decoder used in all convolutional AEs and MD-AEs.

Layers	Output shape	Notes
input	(N_z)	shape is (1) if MD-AE
dense	(13600)	
reshape	(25,17,32)	
upsampling	(50,33,32)	bi-linear interpolation
conv2d	(50,33,16)	3-by-3 filter
upsampling	(100,65,16)	bi-linear interpolation
conv2d	(100,65,8)	3-by-3 filter
upsampling	(200,129,8)	bi-linear interpolation
conv2d	(200,129,1)	3-by-3 filter

Table 4: The encoder used in feedforward AEs.

Layers	Output shape	Notes
input	(64)	
dense	(128)	
batch_normalisation	(128)	
dropout	(128)	
dense	(256)	
batch_normalisation	(256)	
dropout	(256)	
dense	(256)	
batch_normalisation	(256)	
dropout	(256)	
dense	(128)	
batch_normalisation	(128)	
dropout	(128)	
dense	(64)	
batch_normalisation	(64)	
dropout	(64)	
dense	(N_z)	

Table 5: The decoder used in feedforward AEs.

Layers	Output shape	Notes
input	(N_z)	
dense	(64)	
batch_normalisation	(64)	
dropout	(64)	
dense	(128)	
batch_normalisation	(128)	
dropout	(128)	
dense	(256)	
batch_normalisation	(256)	
dropout	(256)	
dense	(256)	
batch_normalisation	(256)	
dropout	(256)	
dense	(128)	
batch_normalisation	(128)	
dropout	(128)	
dense	(64)	

Table 6: The hyperparameters used in all networks. CosineDecayRestarts is a built-in tensorflow learning rate schedule [1, 20].

Which network (table/figure number(s) of the network(s) using these hyperparameters)	MD-AEs and AEs for the unsteady laminar wake (Table 2, 3)	AEs used to search for suitable latent dimension (Figure 21)	AEs for the wind-tunnel turbulent wake (Table 4, 5 and 7)
Learning rate	0.001	0.004	0.0022
Learning rate schedule	n/a	CosineDecayRestarts	CosineDecayRestarts
Activation function	tanh	tanh	tanh
Regularisation	0.0	0.0	0.00003
Dropout rate	0.0%	0.0%	1.4%

Table 7: Large decoder used in section 7.1

Layers	Output shape	Notes
input	(2)	The large decoder is only used with $N_z = 2$
dense	(64)	
batch_normalisation	(64)	
dropout	(64)	
dense	(256)	
batch_normalisation	(256)	
dropout	(256)	
dense	(256)	
batch_normalisation	(256)	
dropout	(256)	
dense	(256)	
batch_normalisation	(256)	
dropout	(256)	
dense	(256)	
batch_normalisation	(256)	
dropout	(256)	
dense	(64)	

B. POD and linear AE

Performing POD is equivalent to solving the quadratic optimization problem [12]

$$\Phi = \arg \min_{\Phi} \|\mathbf{Q} - \Phi^* \Phi^T \mathbf{Q}\|_2^2. \quad (\text{B.1})$$

Let us consider a linear AE in which the number of latent variables is equal to the number of grid points in the input, i.e. $N_z = N$. A linear AE is an AE with linear activation functions. If $N_z < N$, the linear AE recovers only the N_z most energetic POD modes in an L_2 -norm sense. For a linear AE without biases, we express the training as

$$\omega^* = \arg \min_{\omega} \|\mathbf{Q} - \hat{\mathbf{W}} \mathbf{W} \mathbf{Q}\|_2^2, \quad (\text{B.2})$$

which has the solution

$$\hat{\mathbf{W}} \mathbf{W} = \mathbf{I}, \quad (\text{B.3})$$

where $\mathbf{W}$ and $\hat{\mathbf{W}}$ are the weights of the encoder and the decoder of the linear AE. These matrices are not necessarily orthogonal. To obtain the POD solution, we seek orthogonal $\hat{\mathbf{W}}$ and $\mathbf{W}$. Baldi and Hornik [4] showed that for a linear AE without biases, the only solution is the POD solution, whereas all the remaining critical points are saddle points. However, there are saddle points in the optimisation of (B.2). We apply L_2 -regularization to the weights for the linear AE to converge to the POD solution, which is equivalent to minimizing the Frobenius norm of weights, denoted $\|\mathbf{W}\|_F$. When the autoencoder is linear, applying L_2 -regularization of weights is known as applying a “soft constraint of orthogonality” [5, 39]. Training the linear AE with L_2 -regularization is the solution of

$$\omega^* = \arg \min_{\omega} \|\mathbf{Q} - \hat{\mathbf{Q}}\|_2^2 + \gamma \left(\|\mathbf{W}\|_F^2 + \|\hat{\mathbf{W}}\|_F^2 \right), \quad (\text{B.4})$$

where $\gamma \geq 0$ is the regularization factor. The only solution to the optimization problem (B.2) which minimises (B.4) is

$$\mathbf{W} \mathbf{W}^T = \mathbf{I} \quad (\text{B.5})$$

where $\mathbf{I}$ is the identity matrix. Therefore, $\hat{\mathbf{W}} = \mathbf{W}^T$, i.e., the decoder and encoder matrices are orthogonal. A similar analysis can be applied to a linear MD-AE to show that each decomposed field is a POD mode.

C. Physical interpretation is more difficult when unsuitable latent dimension is used

In section 6.2, we show that the MSE of the AEs converges for $N_z = 2$, and that at least two latent variables are needed to describe the periodic behaviour. However, it is difficult to know the theoretical number of latent variables to use in common machine learning tasks. The common approach in machine learning is to treat the latent dimension as a hyperparameter of the networks. By using the laminar wake as an example, we discuss how the choice of the latent variables affects the physical interpretation of the latent variables. First, we show the results from an AE with $N_z = 1$ in Figure 27. The latent variable is periodic in time, but the changes are not smooth (Figure 27a) and have frequency peaks at high frequencies not present in the data (Figure 27b). When a single variable (the angle) is used to represent the periodic behaviour of the dataset $\mathbf{U}$, there is a discontinuity when the angle moves from 2π to 0. The latent variable is forced to use higher frequencies to try to approximate the discontinuity. These higher frequencies are the results of numerical approximation and are unphysical. In an AE with $N_z = 3$, the latent variables contain the first harmonic in addition to the vortex shedding frequency (Figure 28b), which is present in the data but not in the first two data time coefficients (Figure 2, right panel) or the latent variables from an AE with two latent variables (Figure 14b). We discussed (section 6.2) that the additional frequency peak is not necessary for the accurate reconstruction of the dataset.

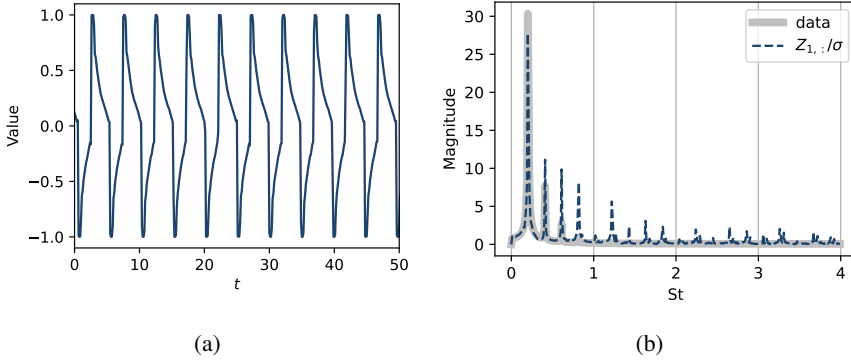


Figure 27: Results from the AE trained with the laminar wake dataset with 1 latent variable. a) The latent variable approximates the discontinuity caused by the angle moving from 2π to 0. b) The frequency spectrum of the latent variable, normalized by its standard deviation, compared with the data frequencies. The latent variables contain frequencies that do not exist in the dataset.

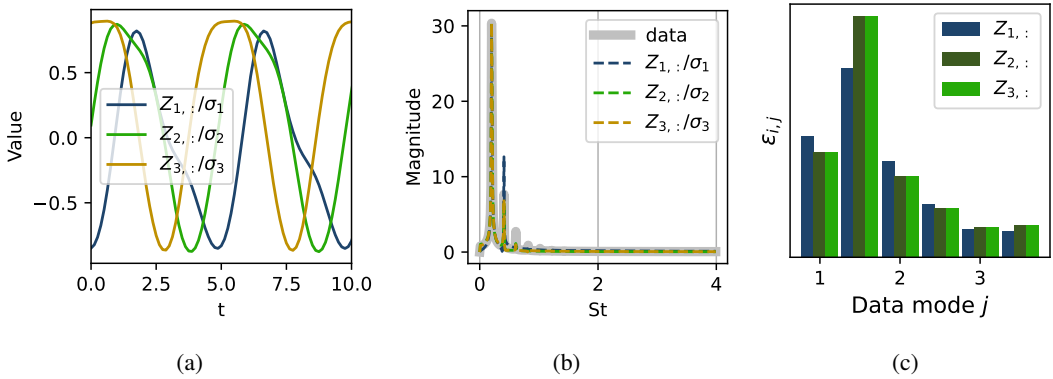


Figure 28: Results from the AE trained with the laminar wake dataset with three latent variables. a) The time trace of the latent variables. b) The frequency spectrum of the latent variables, normalized by their standard deviations. The latent variables contain the vortex shedding and the first harmonic frequency. c) The averaged rate of change of decoder coefficients due to latent variables of an AE with three latent variables. Latent variables $Z_{2,:}$ and $Z_{3,:}$ have the same contributions towards the first six data modes, meaning that both latent variables carry the same information.

The average rate of change shows that latent variables $Z_{2,:}$ and $Z_{3,:}$ contribute equally to the first six data modes (Figure 28c). Equal contribution to all data modes indicates that the two latent variables have the same role in representing the data, which means that the information carried by one of the latent variables is duplicate information contained in the other latent variables. The three latent variables are not suitable because they lead to redundant information. All three latent variables contribute similarly to the first six data modes, making the selection of latent variables more difficult. In the present case, the three latent variables form a group to represent the counter-rotating vortices, even though two latent variables should be, in principle, sufficient.

Both the AE with $N_z = 3$ and the MD-AE with $N_z = 2$ (section 6.1) have latent variables with an additional frequency peak compared to the latent variables of the AE with $N_z = 2$ (section 6.2), even though there is no substantial difference in the reconstruction error. Additionally, unphysical frequencies also arise from numerical approximations when using AE with $N_z = 1$. Therefore, a design consideration

of AEs for nonlinear reduced-order modelling is to use a latent dimension that results in the least number of frequency peaks in the latent variables whilst maintaining the reconstruction accuracy of the AEs.

D. Nomenclature

Acronyms

AE	A standard autoencoder
MD-AE	A mode-decomposing autoencoder
MSE	Mean Square Error
POD	Proper Orthogonal Decomposition
PSD	Power Spectral Density
<i>exp</i>	Experimental
<i>lam</i>	Laminar

Matrix/tensors

$\hat{\Lambda}$	The equivalent energy of the output of an AE.
$\hat{\Lambda}^i$	The equivalent energy of the i -th decomposed field of a MD-AE.
Λ	Diagonal matrix of eigenvalues, sorted from largest to smallest
Φ	POD modes
A	Time coefficients for POD modes
B	Matrix of decoder coefficients
C	Correlation matrix
I	Identity matrix
M^i	The i -th decomposed field of an MD-AE.
P	The wind-tunnel wake dataset, consists of the pressure measurements on a two-dimensional polar grid.
Q	Snapshot matrix
U	The laminar wake dataset, consists of the streamwise velocity of the simulated laminar wake on a two-dimensional grid.
W_P	Weight matrix for the snapshot matrix for POD
Y	An example data matrix used for defining the autoencoders and related methods.
Z	Matrix of latent variables
$\tilde{Q}$	Reconstructed snapshot matrix Q using N_m POD modes.

Non-dimensional group

Re	Reynolds number
St	Strouhal number

Symbols

$\hat{*}$	Dataset * reconstructed by an autoencoder
σ	Standard deviation
ω	parameters of a network
u	Velocities
F_{de}	Function composition that represents the decoder
F_{en}	Function composition that represents the encoder
N	The number of measured variables in the dataset
N_m	Number of truncated POD modes to keep for reconstruction
N_t	Number of time steps
N_u	Number of velocity components
p	Pressure
U_∞	Inlet velocity
D	Diameter of the cylinder/axisymmetric body
f	Frequency

L Domain length
T Period of vortex shedding