Model order reduction with neural networks: Application to laminar and turbulent flows

Kai Fukami[1,*]

Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, CA 90095, USA

Department of Mechanical Engineering, Keio University, Yokohama, 223-8522, Japan

Kazuto Hasegawa

Department of Mechanical Engineering, Keio University, Yokohama, 223-8522, Japan

Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Milano 20156, Italy

Taichi Nakamura, Masaki Morimoto, Koji Fukagata

Department of Mechanical Engineering, Keio University, Yokohama, 223-8522, Japan

Abstract

We investigate the capability of neural network-based model order reduction, i.e., autoencoder (AE), for fluid flows. As an example model, an AE which comprises of a convolutional neural network and multi-layer perceptrons is considered in this study. The AE model is assessed with four canonical fluid flows, namely: (1) two-dimensional cylinder wake, (2) its transient process, (3) NOAA sea surface temperature, and (4) $y-z$ sectional field of turbulent channel flow, in terms of a number of latent modes, a choice of nonlinear activation functions, and a number of weights contained in the AE model. We find that the AE models are sensitive against the choice of the aforementioned parameters depending on the target flows. Finally, we foresee the extensional applications and perspectives of machine learning based order reduction for numerical and experimental studies in fluid dynamics community.

Keywords: reduced order model, autoencoder, wake, turbulence

1. Introduction

Thus far, modal analysis has a significant role to understand and investigate complex fluid flow phenomena. In particular, the combination with linear-theory based data-driven tools, e.g., proper orthogonal decomposition (POD) [1], dynamic mode decomposition [2], and Resolvent analysis [3], has enabled us to examine the fluid flows with interpretable manner [4, 5]. We are now able to see their great abilities in both numerical and experimental studies [6, 7, 8, 9, 10, 11]. In addition to these efforts aided by linear-theory based

*Corresponding author

Email address: `kfukami1@g.ucla.edu` (Kai Fukami)

[1]telephone number : +81-80-1169-4995

methods, the use of neural networks (NNs) has recently been attracting attention as for a promising tool to extract nonlinear dynamics of fluid flow phenomena [12, 13, 14]. In the present paper, we discuss the capability of the NN-based nonlinear model order reduction tool, i.e., autoencoder, while considering canonical fluid flow examples with several assessments.

In recent years, machine learning methods have exemplified their great potential in fluid dynamics, e.g., turbulence modeling [15, 16, 17, 18, 19, 20, 21, 22], and spatio-temporal data estimation [23, 24, 25, 26, 27, 28, 29, 30]. It is not exception for reduced order modeling (ROM), referred to as machine-learning-based ROM (ML-ROM). An extreme learning machine based ROM was presented by San and Maulik [31]. They applied the method for quasi-stationary geophysical turbulence and showed its advantage in terms of stabilization for POD-based ROM. Maulik et al. [32] applied a probabilistic neural network to obtain a temporal evolution of POD coefficients from the parameters of initial condition considering the shallow water equation and the NOAA sea surface temperature. It was exhibited that the proposed method is able to predict the temporal dynamics while showing a confidence interval of its estimation. For more complex turbulence, Srinivasan et al. [33] demonstrated the capability of a long short term memory (LSTM) for a nine-equation shear turbulent flow model. In this way, the ML-ROM efforts combined with the traditional ROM are ongoing now.

Furthermore, the use of autoencoder (AE) has also notable role as an effective model-order reduction tool of fluid dynamics. To the best of our knowledge, the first try of AE application to fluid dynamics is done by Milano and Koumoutsakos [34]. They compared the capability of the POD and a multi-layer perceptron based AE using the Burgers equation and turbulent channel flow. More recently, the population of a convolutional neural network (CNN)-based AE has also been increasing thanks to the concept of filter sharing in CNN, which is suitable to handle high-dimensional fluid data set [35]. Omata and Shirayama [36] used the CNN-AE and POD to reduce a dimension of a wake behind a NACA0012 airfoil. They investigated the difference of temporal behavior on low-dimensional latent space depending on the tool for order reduction. Hasegawa et al. [37, 38] proposed a CNN-LSTM based ROM to predict the temporal evolution of unsteady flows around a bluff body by following only the time series of low-dimensional latent space. The method is also extended to the examination for Reynolds number dependence [39] and turbulent flows [40]. With regard to the perspective on understanding latent modes obtained by AE, Murata et al. [41] suggested a customized AE referred to as mode-decomposing CNN-AE. The proposed method can visualize the contribution of each machine learning based mode. We can also see a wide range of AE studies for fluid flow analyses in [42, 43, 44, 45, 46, 47]. Due to these eagerness for the uses of AE, we here arrive at open questions as follows:

1. How is the dependence on a lot of considerable parameters for AE-based order reduction, e.g., a number of latent modes, activation function, and laminar or turbulence?
2. Where is the limitation of AE-based model order reduction for fluid flows?
3. Can we expect further improvements of AE-based model order reduction with any well-designed methods?

Our aim in the present paper is to demonstrate the example of assessments for AE-based order reduction in fluid dynamics, so as to tackle the aforementioned questions.

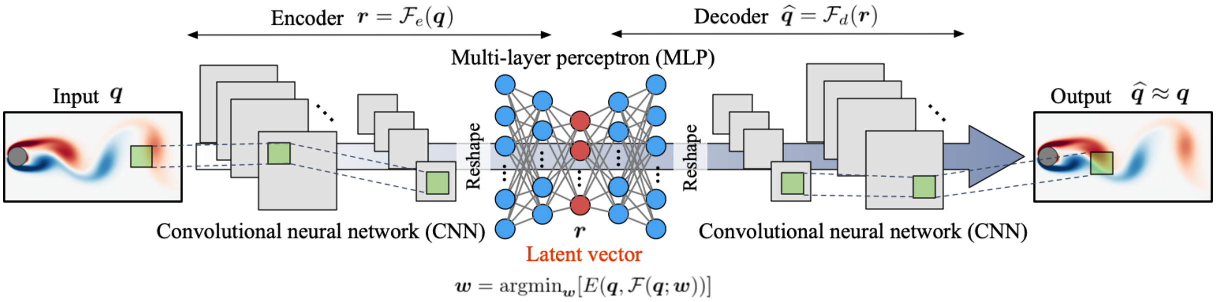


Figure 1: Autoencoder used in the present study. The present model is constructed by a combination of multi-layer perceptron and convolutional neural network.

Especially, we consider the influence on (1) a number of latent modes, (2) choice of activation functions, and (3) the number of weights in an AE with four data sets which cover a wide range of nature from laminar to turbulence. Our presentation is organized as follows: we introduce the present AE models with fundamental theories in section 2. The information for the covered fluid flow data sets is provided in section 3. We then present in section 4 the assessments in terms of various parameters in the AE application to fluid flows. At last, concluding remarks with outlook of AE and fluid flows are stated in section 5.

2. Methods and theories of autoencoder

2.1. Machine learning schemes for construction of autoencoder

In the present study, we use an autoencoder (AE) based on a combination of a convolutional neural network (CNN) and a multi-layer perceptron (MLP), as illustrated in figure 1. Our strategy for order reduction of fluid flows is that the CNN is first used to reduce the dimension into reasonable dimensional space, and the MLP is then employed to map into the latent (bottleneck) space, as explained details later. In this section, let us introduce internal procedures in each machine learning model with mathematical expressions and conceptual figures.

We first utilize a convolutional neural network (CNN)[48] for reducing the original dimension of fluid flows. CNNs have been performed its capability in image recognition thanks to its efficient filtering operation. Recently, the use of CNNs has also been seen in the community of fluid dynamics [49, 50, 51]. CNN is mainly consisted from convolutional layers and pooling layers. As shown in figures 2(a) and (b), the convolutional layer extracts spatial feature from input data by a filter operation,

$$c_{ijm}^{(l)} = \phi\left(\sum_{k=0}^{K-1}\sum_{p=0}^{H-1}\sum_{q=0}^{H-1} c_{i+p-C,j+q-C,k}^{(l-1)} h_{pqkm} + b_m\right), \tag{1}$$

where $C = \mathrm{floor}(H/2)$, $c_{ijm}^{(l-1)}$ and $c_{ijm}^{(l)}$ are the input and output data at layer l and b_m is a bias. A filter is expressed as h_{pqkm} with the size of $(H \times H \times K)$. In the present paper, the size H for the baseline model is set to 3, although it can be changed for the study of dependence on number of weights in section 4.3. The output from the filtering operation

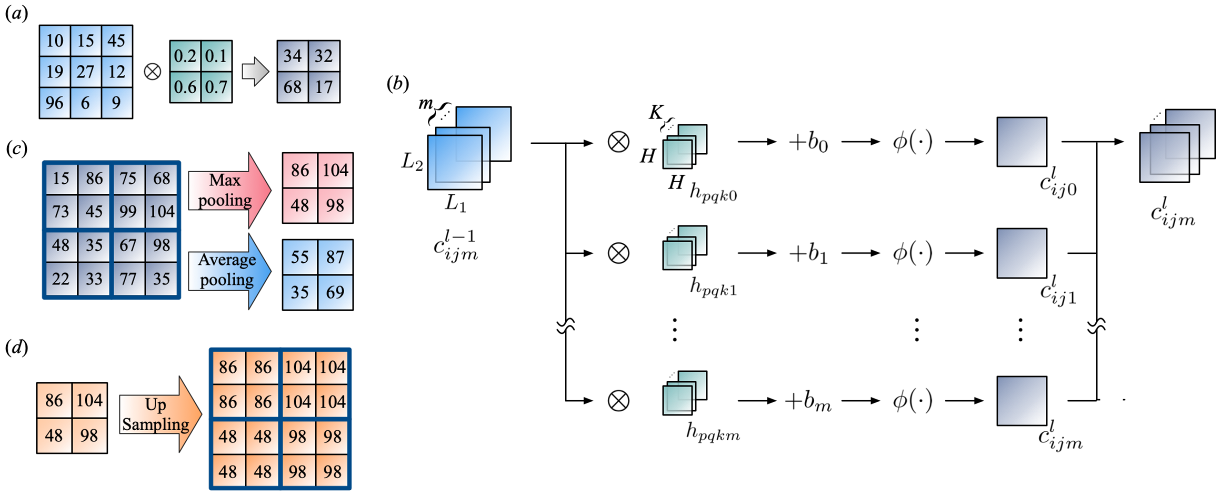


Figure 2: Fundamental operations in convolutional neural network. (a) Convolutional operation with a filter. (b) Computational procedure in the convolution layer. (c) Max and average pooling operations. (d) Upsampling operation.

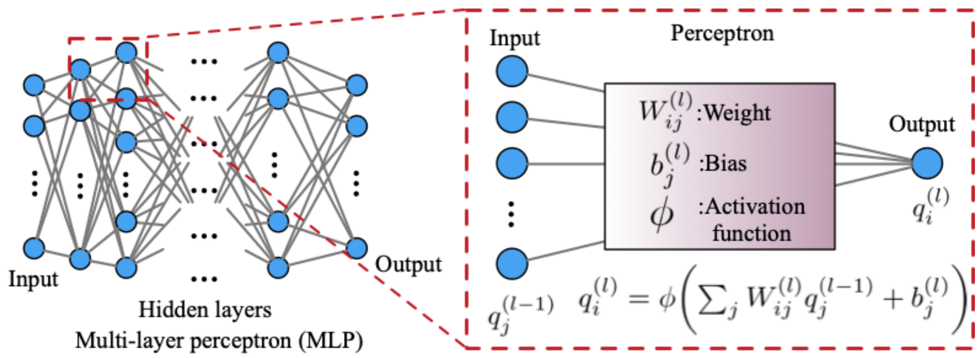


Figure 3: Multi-layer perceptron.

is passed through an activation function ϕ. In the pooling layer illustrated in figure 2(c), representative values, e.g. maximum value or average value, are extracted from arbitrary region by pooling operations, such that the image would be downsampled. It is widely known that the model acquires robustness against input data by pooling operations since spatial sensitivity of the model is decreased [52]. Depending on users' task, the upsampling layer can also be utilized. The upsampling layer copies the value of the low-dimensional images into a high-dimensional space as shown in figure 2(d). With regard to training process, filters in the CNN h are optimized by minimizing a loss function E with back propagation [53] such that $\boldsymbol{w} = \mathrm{argmin}_{\boldsymbol{w}}[E(\boldsymbol{q}_{\mathrm{output}}, \mathcal{F}(\boldsymbol{q}_{\mathrm{input}}; \boldsymbol{w}))]$.

We then apply an MLP [54] for additional order reduction. The MLP has been utilized to a wide range of problems, e.g., classification, regression, and speech recognition [55]. We can also see the applications of MLP to various problems of fluid dynamics such as turbulence modeling [16, 17, 20], reduced order modeling [56], and field estimation [57]. The MLP can be regarded as the aggregate of a perceptron which is a minimum unit, as illustrated in figure 3. The input data from the $(l-1)$th layer are multiplied by a weight

Table 1: The present AE structure for the periodic shedding example at the number of latent space $n_r = 2$.

Encoder		Decoder	
Layer	Output shape	Layer	Output shape
Input	(384, 192, 1)	10th MLP	(4)
1st Conv2D	(384, 192, 16)	11th MLP	(8)
1st max pooling	(192, 96, 16)	12th MLP	(16)
2nd Conv2D	(192, 96, 8)	13th MLP	(32)
2nd max pooling	(96, 48, 8)	14th MLP	(64)
3rd Conv2D	(96, 48, 8)	15th MLP	(128)
3rd max pooling	(48, 24, 8)	16th MLP	(256)
4th Conv2D	(48, 24, 2)	17th MLP	(512)
4th max pooling	(24, 12, 2)	18th MLP	(576)
Flatten	(576)	Reshape	(24, 12, 2)
1st MLP	(512)	1st upsampling	(48, 24, 2)
2nd MLP	(256)	5th Conv2D	(48, 24, 2)
3rd MLP	(128)	2nd upsampling	(96, 48, 2)
4th MLP	(64)	6th Conv2D	(96, 48, 8)
5th MLP	(32)	3rd upsampling	(192, 96, 8)
6th MLP	(16)	7th Conv2D	(192, 96, 8)
7th MLP	(8)	4th upsampling	(384, 192, 8)
8th MLP	(4)	8th Conv2D	(384, 192, 16)
9th MLP (latent space)	(2)	Output	(384, 192, 1)

Table 2: Hyper parameters used in the present AE model.

Parameter	Value	Parameter	Value
Batch size	64	Early stopping patience	50
Number of epochs	5000	Percentage of training data	70%
Learning rate of Adam	0.001	Learning rate decay of Adam	0
β_1 of Adam	0.9	β_2 of Adam	0.999

$\boldsymbol{W}$. These inputs construct a linear superposition $\boldsymbol{W}\boldsymbol{q}$ with biases $\boldsymbol{b}$ and then passed through a nonlinear activation function ϕ,

$$q_i^{(l)} = \phi\left(\sum_j W_{ij}^{(l)} q_j^{(l-1)} + b_j^{(l)}\right). \tag{2}$$

As illustrated in figure 3, these perceptrons of MLP are connected with each other and have a fully-connected structure. As well as the CNN, weights among all connections W_{ij} are optimized with the minimization manner.

By combining two methods introduced above, we here construct the AE as illustrated in figure 1. The AE has a lower dimensional space called latent space at the bottleneck so that a representative feature of high-dimensional data $\boldsymbol{q}$ can be extracted by setting the target data as both input and output. In other words, if we can obtain the similar output to input data, the dimensional reduction can be successfully achieved onto the

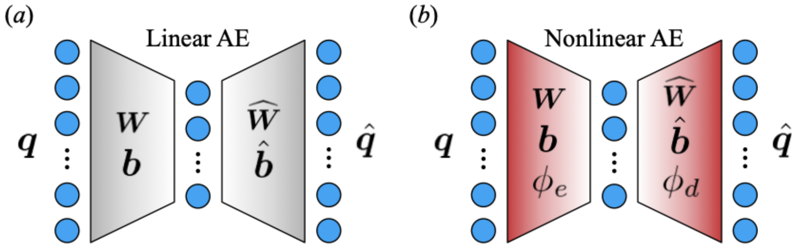


Figure 4: (a) Linear autoencoder. (b) Autoencoder with nonlinear activation functions ϕ_e and ϕ_d.

latent vector $\boldsymbol{r}$. In turn, with over-compression of the original data $\boldsymbol{q}$, of course, the decoded field $\widehat{\boldsymbol{q}}$ cannot be remapped well. In sum, these relations can be formulated as

$$\boldsymbol{r} = \mathcal{F}_e(\boldsymbol{q}), \quad \boldsymbol{q} \approx \widehat{\boldsymbol{q}} = \mathcal{F}_d(\boldsymbol{r}), \tag{3}$$

where $\mathcal{F}_e$ and $\mathcal{F}_d$ are encoder and decoder of autoencoder as shown in figure 1. Mathematically, in the training process for obtaining the autoencoder model $\mathcal{F}$, the weights $\boldsymbol{w}$ is optimized to minimize an error function E such that $\boldsymbol{w} = \mathrm{argmin}_{\boldsymbol{w}}[E(\boldsymbol{q}, \mathcal{F}(\boldsymbol{q}; \boldsymbol{w}))]$. In the present study, we use the L_2 norm error as the cost function E and Adam optimizer [53] is applied for obtaining the weights. Other parameters used for construction of the present AE are summarized in table 2. To avoid an overfitting, we set the early stopping criteria [58]. With regard to the internal steps of the present AE, we first apply the CNN to reduce the spatial dimension of flow fields by $\mathcal{O}(10^2)$. The MLP is then used to obtain the latent modes by feeding the compressed vector by the CNN, as summarized in table 1 which shows an example of the AE structure for the cylinder wake problem at the number of latent vector $n_r = 2$. For a case whose number of latent vector does not reach $\mathcal{O}(10^2)$, e.g., $n_r = 3072$ of turbulent channel flow, the order reduction is conducted by means of only the CNN (without MLP).

2.2. The role of activation functions in autoencoder

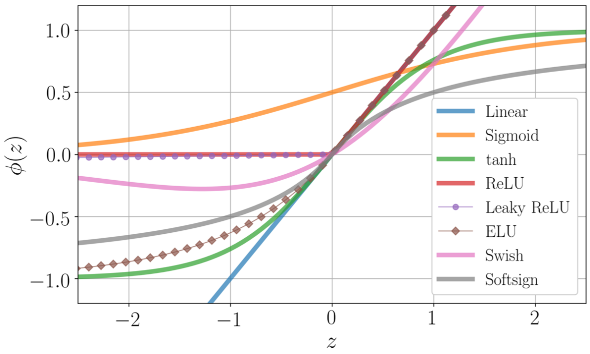
One of key features in the mathematical procedure of AE is the use of activation functions. This is exactly the reason why neural networks can account for nonlinearlity into their estimations. Here, let us mathematically introduce the strength of nonlinear activation functions for model order reduction by comparing to proper orthogonal decomposition (POD) which has been known as equivalent to AE with linear activation function $z = \phi(z)$. The details can be seen in some previous works [59, 60, 61].

We first visit the POD-based order reduction for high-dimensional data matrix $\boldsymbol{q} \in \mathbb{R}^{n_t \times n_g}$ with a reduced basis $\boldsymbol{\Gamma} \in \mathbb{R}^{n_r \times n_g}$, where n_t, n_r and n_g explain numbers of collected snapshots, latent modes and spatial discretized grids, respectively. An optimization procedure of POD can be formulated with an L_2 minimization manner,

$$\boldsymbol{\Gamma} = \mathrm{argmin}_{\boldsymbol{\Gamma}} ||(\boldsymbol{q} - \overline{\boldsymbol{q}}) - \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma}(\boldsymbol{q} - \overline{\boldsymbol{q}})||_2, \tag{4}$$

where $\overline{\boldsymbol{q}}$ is a mean value of high-dimensional data matrix. Next, we consider an AE with linear activation function $z = \phi(z)$, as shown in figure 4(a). Analogous to equation 4, the optimization procedure can be written as

$$\boldsymbol{\omega} = \mathrm{argmin}_{\boldsymbol{w}} ||\boldsymbol{q} - \widehat{\boldsymbol{q}}||_2, \quad \text{where} \quad \widehat{\boldsymbol{q}} = \widehat{\boldsymbol{W}}(\boldsymbol{W}\boldsymbol{q} + \boldsymbol{b}) + \widehat{\boldsymbol{b}}, \tag{5}$$



Activation function	Equation
Linear mapping	$\phi(z) = z$
Sigmoid	$\phi(z) = (1 + e^{-z})^{-1}$
tanh	$\phi(z) = (e^z - e^{-z})/(e^z + e^{-z})$
ReLU	$\phi(z) = \max[0, z]$
Leaky ReLU	$\phi(z) = z$ (if $z > 0$), $0.01z$ (otherwise)
ELU	$\phi(z) = z$ (if $z > 0$), $e^z - 1$ (otherwise)
Swish	$\phi(z) = z/(1 + e^{-z})$
Softsign	$\phi(z) = z/(1 + \lvert z \rvert)$

Figure 5: Activation functions covered in the present study.

with a reconstructed matrix $\widehat{\boldsymbol{q}}$, and weights $\boldsymbol{w} = [\boldsymbol{W}, \boldsymbol{b}, \widehat{\boldsymbol{W}}, \widehat{\boldsymbol{b}}]$ in the AE which contains the weights of encoder $\boldsymbol{W}$ and decoder $\widehat{\boldsymbol{W}}$ and the biases of encoder $\boldsymbol{b}$ and decoder $\widehat{\boldsymbol{b}}$. Substituting these variables with $\boldsymbol{q}$ for equation 4, the argmin formulation can be transformed as

$$\begin{aligned} \boldsymbol{\omega} &= \mathrm{argmin}_{\boldsymbol{w}} ||\boldsymbol{q} - \widehat{\boldsymbol{q}}||_2 \\ &= \mathrm{argmin}_{\boldsymbol{w}} ||\boldsymbol{q} - \widehat{\boldsymbol{W}}(\boldsymbol{W}\boldsymbol{q} + \boldsymbol{b}) - \widehat{\boldsymbol{b}}||_2 \\ &= \mathrm{argmin}_{\boldsymbol{w}} ||(\boldsymbol{q} - \widehat{\boldsymbol{b}}) - (\widehat{\boldsymbol{W}}\boldsymbol{W}\boldsymbol{q} + \widehat{\boldsymbol{W}}\boldsymbol{b})||_2. \end{aligned} \tag{6}$$

Noteworthy here is that equations 4 and 6 are equivalent with each other by replacing the variables as follows,

$$\boldsymbol{W} \to \boldsymbol{\Gamma}, \quad \widehat{\boldsymbol{W}} \to \boldsymbol{\Gamma}^{\mathrm{T}}, \quad \widehat{\boldsymbol{b}} \to \overline{\boldsymbol{q}}, \quad \boldsymbol{b} \to -\overline{\boldsymbol{q}}\boldsymbol{\Gamma}. \tag{7}$$

These transformations tell us that the AE with linear activation is essentially same order reduction tool with POD. In other words, with regard to the AE with nonlinear activation functions as illustrated in figure 4(b), the nonlinear activation functions of encoder ϕ_e and decoder ϕ_d are multiplied to the weights (reduced basis) in equation 4, such that

$$\boldsymbol{\Gamma} = \mathrm{argmin}_{\boldsymbol{\Gamma}} ||(\boldsymbol{q} - \overline{\boldsymbol{q}}) - \phi_d \boldsymbol{\Gamma}^{\mathrm{T}} \phi_e \boldsymbol{\Gamma}(\boldsymbol{q} - \overline{\boldsymbol{q}})||_2. \tag{8}$$

Note that the weights $\boldsymbol{w}$ including biases are optimized in the training procedure of autoencoder.

As discussed above, the choice of nonlinear activation function is crucial to outperform the traditional linear method. In this study, we consider the use of various activation functions as summarized in figure 5. As shown, we cover a wide nature of nonlinear functions which are the well-used candidates, although these are not all of used functions in the machine learning field. We refer you to Nwankpa et al. [62] for details of each nonlinear activation function. For the baseline model of each example, we use Rectified Linear Unit (ReLU) [63] which has been widely known as a good candidate in terms of weight updates. Note in passing that we use same activation function at all layers in each model, e.g., the case of ReLU for 1st and 2nd layers and tanh for 2nd and 3rd layers does not consider. The optimization of choice for these parameters may enable models to improve the compression ability of AE [56, 64, 65, 66]. In the present paper, we perform a three-fold cross validation [67], although the representative flow fields with ensemble error values are reported.

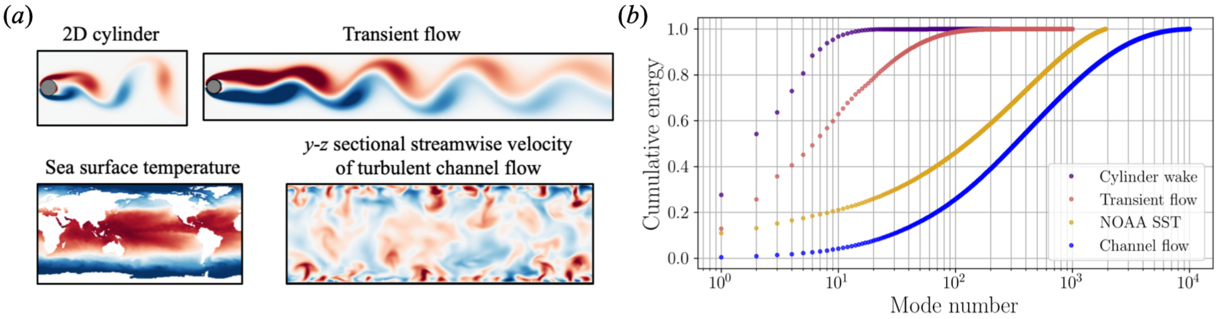


Figure 6: (a) Example of flow fields used in the present study. (b) Normalized cumulative sum of the singular values.

3. Setup for covered examples of fluid flows

In the present study, we cover a wide range of fluid flow nature considering four examples which include laminar and turbulence, as shown in figure 6(a). The broad spread of complexity on the covered example flows can be also seen in a cumulative singular value spectrum presented in figure 6(b).

3.1. Two-dimensional cylinder wake at $Re_D = 100$

The first example is a two-dimensional cylinder wake at $Re_D = 100$. The data set is prepared using a two-dimensional direct numerical simulation (DNS) [68]. The governing equations are the continuity and the incompressible Navier–Stokes equation,

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{9}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla} p + \frac{1}{Re_D} \nabla^2 \boldsymbol{u}, \tag{10}$$

where $\boldsymbol{u} = [u, v]$ and p are the velocity vector and pressure, respectively. All quantities are non-dimensionalized with the fluid density, the free-stream velocity, and the cylinder diameter. The size of the computational domain is (L_x, L_y)=(25.6, 20.0). The cylinder center is located at $(x, y) = (9, 0)$. A Cartesian grid with the grid spacing of $\Delta x = \Delta y = 0.025$ is used for the numerical setup. The number of grid points for the present DNS is $(N_x, N_y) = (1024, 800)$. The flow field around the cylinder is only extracted for data set such that $8.2 \leq x \leq 17.8$ and $-2.4 \leq y \leq 2.4$ with $(N_x^*, N_y^*) = (384, 192)$. We use the vorticity field ω as the data attribute. The time interval of flow data is 0.25 corresponding to approximately 23 snapshots per a period with the Strouhal number equals to 0.172. For training the present AE model, we use 1000 snapshots.

3.2. Transient wake at $Re_D = 100$

We then consider a transient wake behind the cylinder which is a non-periodical manner in time. Since the almost same computational procedure is applied with the case of periodic shedding, let us briefly touch the notable contents. The training data is obtained from the DNS which is almost same set up with the first example. Here, the domain is extended 3 folds in x direction such that $(N_x^*, N_y^*) = (1152, 192)$ with $(L_x^*, L_y^*) = (28.8, 4.8)$. Analogous to the first example, we use the vorticity field ω as the data attribute. For training the present AE model, we use 4000 snapshots whose setting is denser pick up in time than the periodic shedding case to catch the transient nature correctly.

3.3. NOAA sea surface temperature

The third example is the NOAA sea surface temperature data set [69] obtained from satellite and ship observations. The aim in using this data set is to examine the applicability of AE to more practical situations, such as the case without governing equation. We should note that this data set is driven by the influence of seasonal periodicity. The spatial resolution here is 360×180 based on a one degree grid. We use the 20 years of data corresponding to 1040 snapshots spanning 1981 to 2001.

3.4. $y - z$ sectional field of turbulent channel flow at $Re_\tau = 180$

To investigate the capability of AE for chaotic turbulence, we lastly consider the use of a $y - z$ sectional streamwise velocity field of turbulent channel flow at $Re_\tau = 180$ under a constant pressure gradient condition. The data set is prepared by direct numerical simulation developed by Fukagata et al.[70]. The governing equations are the incompressible Navier–Stokes equations, i.e.,

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \tag{11}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla} p + \frac{1}{Re_\tau} \nabla^2 \boldsymbol{u}, \tag{12}$$

where $\boldsymbol{u} = [u \; v \; w]^{\mathrm{T}}$ represents the velocity with u, v and w in the streamwise (x), the wall-normal (y) and the spanwise (z) directions. Also, t is the time, p is the pressure, and $Re_\tau = u_\tau \delta / \nu$ is the friction Reynolds number. The quantities are non-dimensionalized with the half-width δ in the channel and the friction velocity u_τ. The size of the computational domain and the number of grid points are $(L_x, L_y, L_z) = (4\pi\delta, 2\delta, 2\pi\delta)$ and $(N_x, N_y, N_z) = (256, 96, 256)$, respectively. The grid in the x and z directions are arranged to uniform, and non-uniform grid is applied in the y direction. A no-slip boundary condition is applied to the walls and the periodic boundary condition is formed in the x and z directions. We use the fluctuation component of a $y - z$ sectional streamwise velocity u' as the representative data attribute for the present AE.

4. Results

In this section, we examine the influence on the number of latent modes (sec. 4.1), choice of activation function for hidden layers (sec. 4.2), and number of weights contained in an autoencoder (sec. 4.3) with the data sets introduced above.

4.1. Number of latent modes

Let us first investigate the dependence of reconstruction accuracy by autoencoder (AE) on the number of latent space n_r. The expected trend in this assessment for all data sets is that the reconstruction error would increase with reducing n_r since the AE needs to discard the information while extracting dominant features from high-dimensional flow fields into a limited number of latent space.

The relationship between the number of latent space n_r and L_2 error norm $\epsilon = ||\boldsymbol{q}_{\mathrm{Ref}} - \boldsymbol{q}_{\mathrm{ML}}||_2 / ||\boldsymbol{q}_{\mathrm{Ref}}||_2$, where $\boldsymbol{q}_{\mathrm{Ref}}$ and $\boldsymbol{q}_{\mathrm{ML}}$ indicate a reference field and a reconstructed field by an AE, with all data sets is summarized in figure 7. For the periodic shedding case, the structure of whole field can be captured with only two modes as presented in figure 7(a). However, the L_2 error norm at $n_r = 2$ is approximately 0.4 which shows relatively large

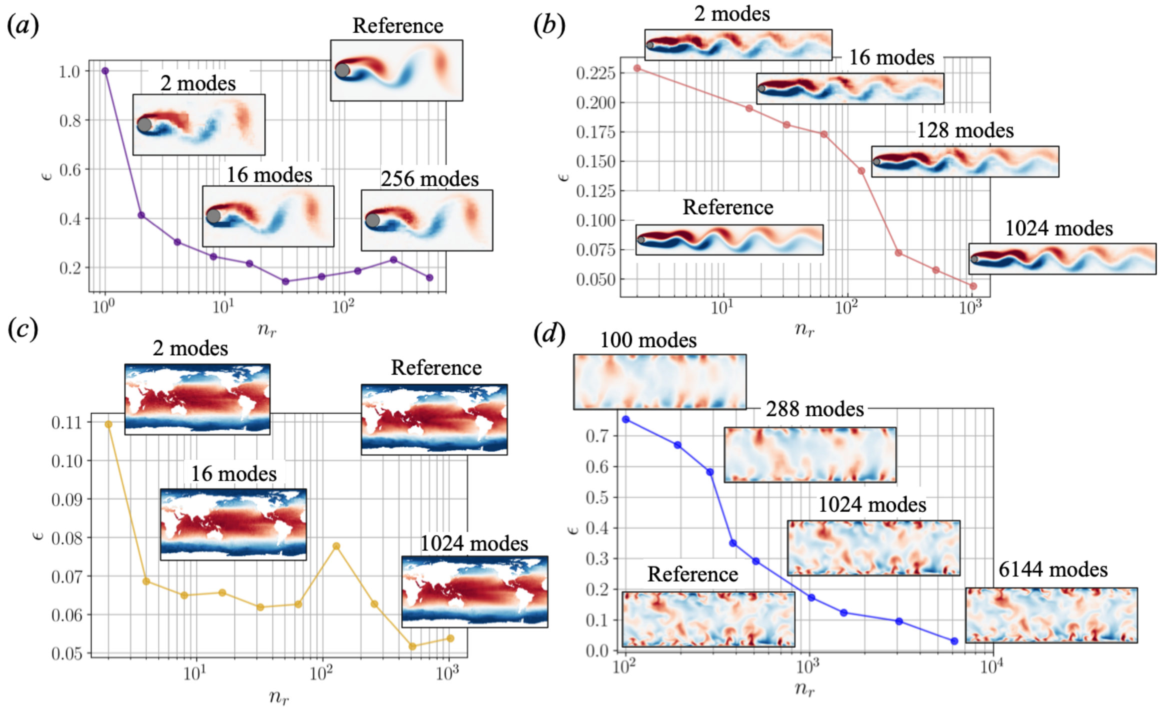


Figure 7: The relationship between the number of latent space n_r and L_2 error norm $\epsilon = ||\boldsymbol{q}_{\rm Ref} - \boldsymbol{q}_{\rm ML}||_2/||\boldsymbol{q}_{\rm Ref}||_2$. (*a*) Two-dimensional cylinder wake. (*b*) Transient flow. (*c*) NOAA sea surface temperature. (*d*) $y-z$ sectional streamwise velocity fluctuation of turbulent channel flow. Three-fold cross validation is performed, although not shown.

value despite the fact that the periodic shedding behind a cylinder at $Re_D = 100$ can be represented by only one scalar value [71]. The error value here is also larger than the reported L_2 error norms in the previous works [41, 72] which handle the cylinder wake at $Re_D = 100$ using AEs with $n_r = 2$. This is likely due to a choice of activation function and we find the significant difference for influence on the choice, which will be explained in section 4.2. It is also striking that the L_2 error at $n_r = 10^1 - 10^2$ somehow increases. This indicates that the number of weights n_w contained in the AE has also non-negligible effect for the mapping ability, since the number of weights is increasing with n_r in our AE setting as it can be seen in table 1, e.g., $n_w(n_r = 128) > n_w(n_r = 256)$. This viewpoint will be investigated in section 4.3. We then apply the AE to the transient flow as shown in figure 7(*b*). Since the transient case is also laminar nature, the whole trend of wake can be acquired with a few number of modes, e.g., 2 and 16 modes. The behavior of AE with the data which has no modelled governing equation can be seen in figure 7(*c*). As shown, the entire trend of the field can be extracted with only two modes since the considered data here is driven with seasonal periodicity as mentioned above. The reason for a rise in error at $n_r = 128$ is likely because of the influence on the number of weights, which is also observed in the cylinder example. We can see the strikingly difference of AE performance against the previous three examples in figure 7(*d*). For the turbulence case, the required number of modes to present finer scales is visibly larger than the others since turbulence has a wider range of scales. The difficulty of compression for turbulence can also be found in terms of L_2 error norms.

To further examine the dependence of reconstruction ability by AEs on number of

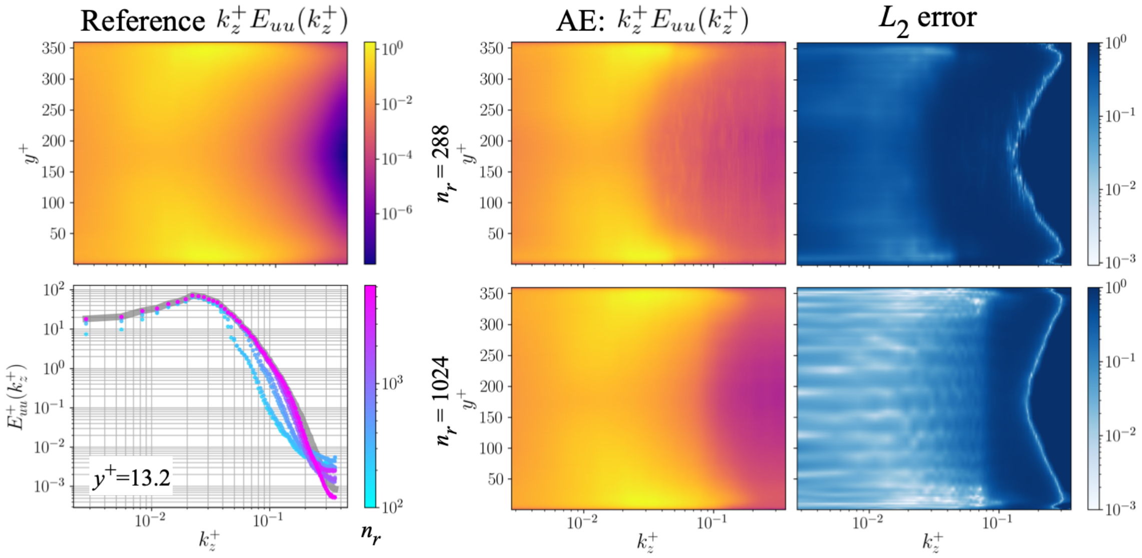


Figure 8: Premultiplied spanwise kinetic energy spectrum at $n_r = 288$ and 1024. The lower left shows $E^+_{uu}(k^+_z)$ at $y^+ = 13.2$. In this plot, the gray line indicates the reference DNS and other colored lines explain the AE models.

latent modes in terms of scales in turbulence, we check the premultiplied spanwise kinetic turbulent energy spectrum $k^+_z E^+_{uu}(k^+_z)$ in figure 8. As shown here, the error level $||k^+_z E^+_{uu}(k^+_z)_{\rm Ref} - k^+_z E^+_{uu}(k^+_z)_{\rm ML}||_2/||k^+_z E^+_{uu}(k^+_z)_{\rm Ref}||_2$ decreases with increasing n_r, especially on the low-wavenumber space. This implies that the AE model extracts the information of low-wavenumber region preferentially. Noteworthy here is that the low error level portion is suddenly appeared on high-wavenumber region, i.e., white lines at $n_r = 288$ and 1024 in the L_2 error map of figure 8. These correspond to the point where the $E^+_{uu}(k^+_z)$ curves of the DNS and AE cross each other, as it can be seen in the lower left of figure 8. The under or overestimation of the kinetic turbulent energy spectrum is caused by a combination of several reasons, e.g., underestimation of u' because of L_2 fitting manner and relationship between a squared velocity and energy such that $\overline{u^2} = \int E^+_{uu}(k^+_z)dk^+_z$, although it is difficult to separate these into each considerable element. Note that the aforementioned observation is also seen in Scherl et al. [73] who takes a robust principal component analysis to the channel flow.

4.2. Choice of activation function

Next, we examine the influence of the AE based low dimensionalization on the choice of activation functions. As presented in figure 5, we cover a wide range of functions: namely linear mapping, sigmoid, hyperbolic tangent (tanh), rectified linear unit (ReLU, baseline), Leaky ReLU, ELU, Swish, and Softsign. Since the use of activation function is a essential key to account for nonlinearity into the AE-based order reduction as discussed in section 2.2, the investigation of the choice here can be regarded as immensely crucial for the construction of AE. For each fluid flow example, we consider the same AE construction and parameters with the model at the first assessment (except for the activation function), which achieved the L_2 error norm of approximately 0.2. Note that it is exception for the example of sea surface temperature, since the highest reported L_2 error over the covered number of latent modes is already lower than 0.2. Following this rule, the numbers

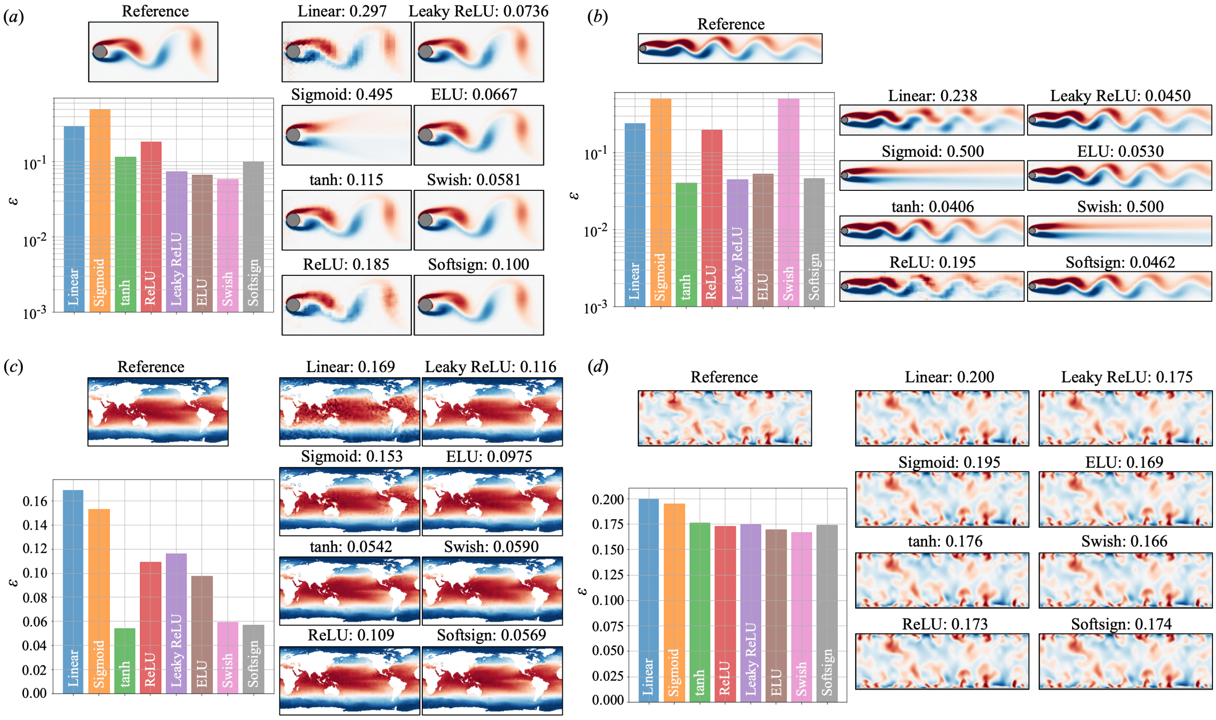


Figure 9: Dependence of mapping ability on activation functions. (a) Two-dimensional cylinder wake. (b) Transient flow. (c) NOAA sea surface temperature. (d) $y-z$ sectional streamwise velocity fluctuation of turbulent channel flow. Three-fold cross validation is performed, although not shown. Values above contours indicate the L_2 error norm.

of latent modes for each example are 16 (cylinder wake), 16 (transient), 2 (sea surface temperature), and 1024 (turbulent channel flow), respectively.

The dependence of the AE reconstruction on activation functions is summarized in figure 9. As compared to the linear mapping $\phi(z) = z$, the use of nonlinear activation functions leads to improve the accuracy with almost all cases. However, as shown in figure 9(a), the field with sigmoid function $\phi(z) = (1 + e^{-z})^{-1}$ is worse than the others including linear mapping, which shows the mode-0 like field. This is likely because of the widely known fact that the use of sigmoid function has often encountered the vanishing gradient which cannot update weights of neural networks accordingly [62]. The same trend can also be found with the transient flow, as presented in figure 9(b). To overcome the aforementioned issue, the swish function $\phi(z) = z/(1 + e^{-z})$ which has a similar form to sigmoid was proposed [74]. This swish can improve the accuracy of the AE based order reduction with the example of periodic shedding, although it is not for the transient. These observations indicate the importance to check the ability of each activation function so as to represent an efficient order reduction. The other notable trend here is the difference of the influence on the selected activation functions depending on the target flows. In our investigation, the appropriate choice of nonlinear activation functions can present significantly improvement for the first three problem settings, i.e., periodic wake, transient, and sea surface temperature. In contrast, we do not see noteworthy differences among the nonlinear functions with the turbulence example. It implies that a range of contained scales in flows has also the effect to the mapping ability of AE with nonlinear activation

Table 3: The number of weights in the baseline AEs for section 4.3.1.

Examples	All n_w	CNN $n_{w,\mathrm{CNN}}$	MLP $n_{w,\mathrm{MLP}}$
Cylinder wake	945721	4137	941584
Transient	270057	5305	264752
Sea surface temperature	59311	37093	22218
Turbulent channel flow	21930	21930	—

functions. Summarizing above, a careful choice of activation functions should be taken to implement the AE with acceptable order reduction since influence and abilities of activation functions highly depend on target flows which users handle.

4.3. Number of weights

As discussed in section 4.1, the number of weights contained in AEs may also have the influence for the mapping ability. Here, let us examine this viewpoint with two investigations as follows:

1. by changing the amount of parameters in the present AE. (section 4.3.1)
2. by pruning weights in the present AE. (section 4.3.2)

4.3.1. Change of parameters

As mentioned above, the present AE comprises of a convolutional neural network (CNN) and multi-layer perceptrons (MLP). For the assessment here, to change a number of weights in the AE, we tune various parameters in both the CNN and the MLP, e.g., the number of hidden layers in both CNN and MLP, the number of units in MLP, the size and number of filters in CNN. Note in passing that we here only focus on the number of weights, although which parameters are tuned may also have effects to AE's ability, e.g., we do not consider difference of cases that the number of weights is same such that $n_{w,1} = n_{w,2}$ but one $n_{w,1}$ is achieved by tuning the number of layers, and other $n_{w,2}$ is by changing the number of units. Also, the same AE construction and parameters at the second assessment with ReLU function are considered, which achieved the L_2 error norm of approximately 0.2 in the first investigation. Analogous to the second investigation for activation functions, the numbers of latent modes n_r for each example are 16 (cylinder wake), 16 (transient), 2 (sea surface temperature), and 1024 (turbulent channel flow), respectively. The dependence of the mapping ability on n_w is investigated as follows,

1. The number of weights in the MLP $n_{w,\mathrm{MLP}}$ is fixed and the number of weights in the CNN $n_{w,\mathrm{CNN}}$ is changed. The L_2 error is assessed comparing the original number of weights in the CNN such that $n_{w,\mathrm{CNN}}/n_{w,\mathrm{CNN:original}}$.
2. The number of weights in the CNN $n_{w,\mathrm{CNN}}$ is fixed and the number of weights in the MLP $n_{w,\mathrm{MLP}}$ is changed. The L_2 error is assessed comparing the original number of weights in the MLP such that $n_{w,\mathrm{MLP}}/n_{w,\mathrm{MLP:original}}$.

The number of weights $n_w = n_{w,\mathrm{CNN}} + n_{w,\mathrm{MLP}}$ in the baseline models is summarized in table 3. Note again that the CNN first works to reduce a dimension of flows by $\mathcal{O}(10^2)$ and the MLP is then utilized to map into the target dimension as stated in section 2.1. Hence, the considered models for the cylinder wake, transient flow, and sea surface temperature

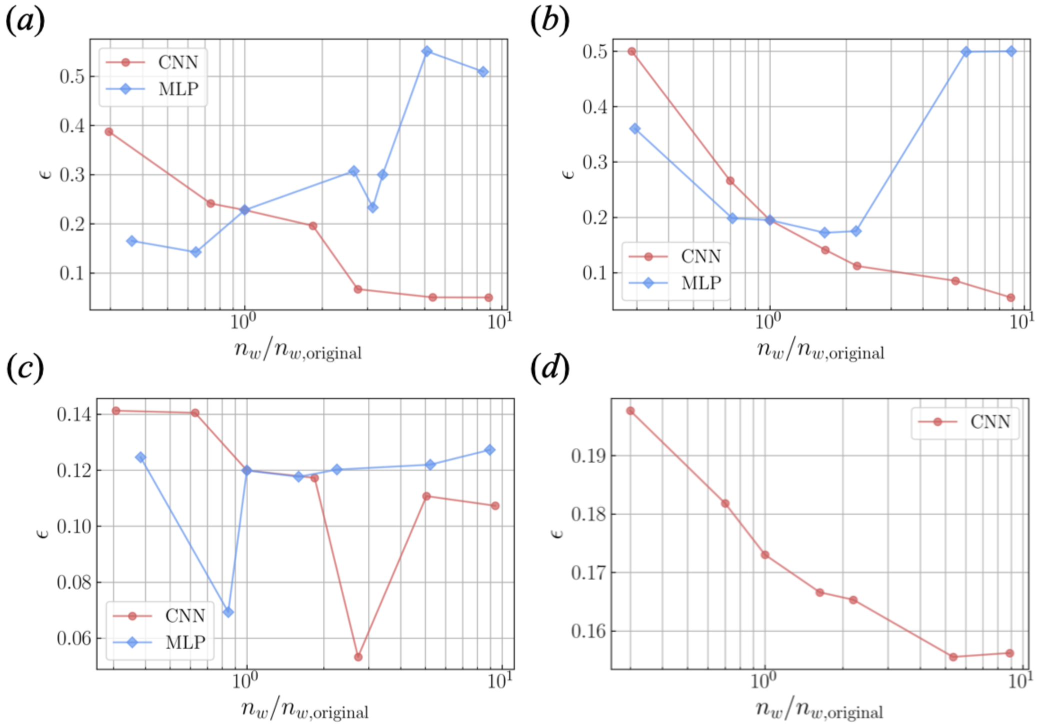


Figure 10: Dependence on the number of weights tuned by modification of parameters in AE. (a) Two-dimensional cylinder wake. (b) Transient flow. (c) NOAA sea surface temperature. (d) $y-z$ sectional streamwise velocity fluctuation of turbulent channel flow.

have both the CNN and the MLP. On the other hand, the MLP is not applied to the model for the example of turbulent channel flow, since $n_r = 1024$. Following this reason, only the influence on the weights of CNN is considered for the turbulent case.

The dependence on the number of weights tuned by changing of parameters is summarized in figure 10. The expected trend here is that an error decreases with increasing the number of weights, which is seen with the example of turbulent channel flow shown in figure 10(d). The same trend can also be observed with the parameter modification in CNN for the examples of periodic shedding and transient flow, as presented in figures 10(a) and (b). The error curve for the parameter modification in MLP, however, shows contrary trend in both cases. This is likely because there are too many numbers of weights in MLP and this deepness leads to difficulty of weight updates, even if we use the ReLU function. With regard to the models of sea surface temperature, the error curves of both CNN and MLP show a valley-like behavior. It implies that these numbers of weights which achieve the lowest errors are appropriate in terms of weight updates. For the channel flow example, the error converges around 0.15 as shown in figure 10(d), but we should note that it already reaches almost $n_w/n_{w,\text{original}} = 10$. These heavy models should be a big computational burden in terms of both time and storage. In sum, users should care for the choice of parameters contained in AE, which directly affect to the number of weights, depending on user's environment, although an increase of number of weights basically leads to acquire a good AE ability.

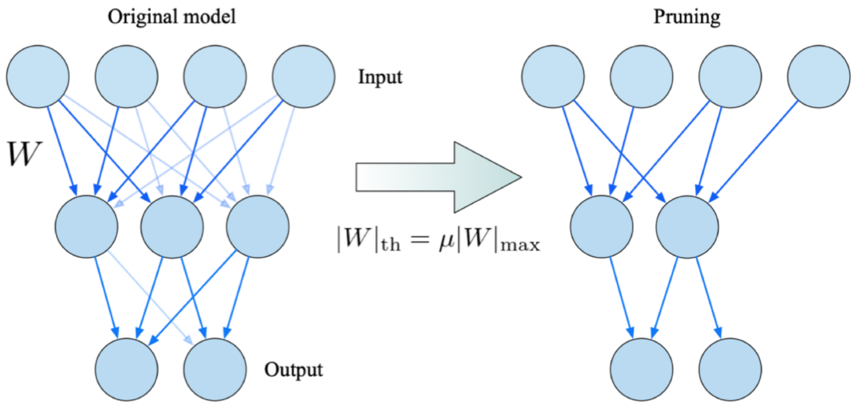


Figure 11: Pruning for multi-layer perceptrons.

Table 4: Models for the pruning study.

Examples	# of latent space n_r	Activation function
Cylinder wake	16	Swish
Transient	16	tanh
Sea surface temperature	2	tanh
Turbulent channel flow	1024	Swish

4.3.2. Pruning operation

Through the investigations above, we found that the appropriate parameter choice enables the AE to improve its mapping ability. Although we may encounter the problem of weight updates as reported above, an increase of number of weights n_w is also basically crucial to achieve a nice AE compression. However, if the AE model could be constructed well with a large number of weights, the increase of n_w also directly corresponds to a computational burden in terms of both time and storage for *a posteriori* manner. Hence, our next interest here is whether an AE model trained with a large number of weights can maintain its mapping ability with a pruning operation or not. The pruning is a method to cut edges of connection in neural networks and has been known as a good candidate to reduce the computational burden while keeping accuracy of both classification and regression tasks [75]. We here assess the possibility of the use of pruning for AEs with fluid field data. As illustrated in figure 11, we do pruning by taking a threshold based on a maximum value of weights per each layer such that $|W|_{\rm th} = \mu|W|_{\rm max}$, where W and μ express weights and a sparsity threshold coefficient, respectively. An original model without pruning corresponds to $\mu = 0$. The pruning for the CNN is also performed as well as MLP. For this assessment, we consider the best models in our investigation for activation functions (section 4.2), as summarized in table 4.

Let us present in figure 12 the results of pruning operation for the AEs with fluid flows. To check the weight distribution of AEs for each problem setting, the relationship between a sparsity threshold coefficient μ and a sparsity factor $\gamma = 1 - n_w/n_{w,\rm original}$ is studied in figure 12(a). With all cases, μ and γ have a proportional relationship as can be expected. Noteworthy here is that the sparsity of the model for the turbulent channel flow is lower than that for the others over the considered μ. This indicates the significant

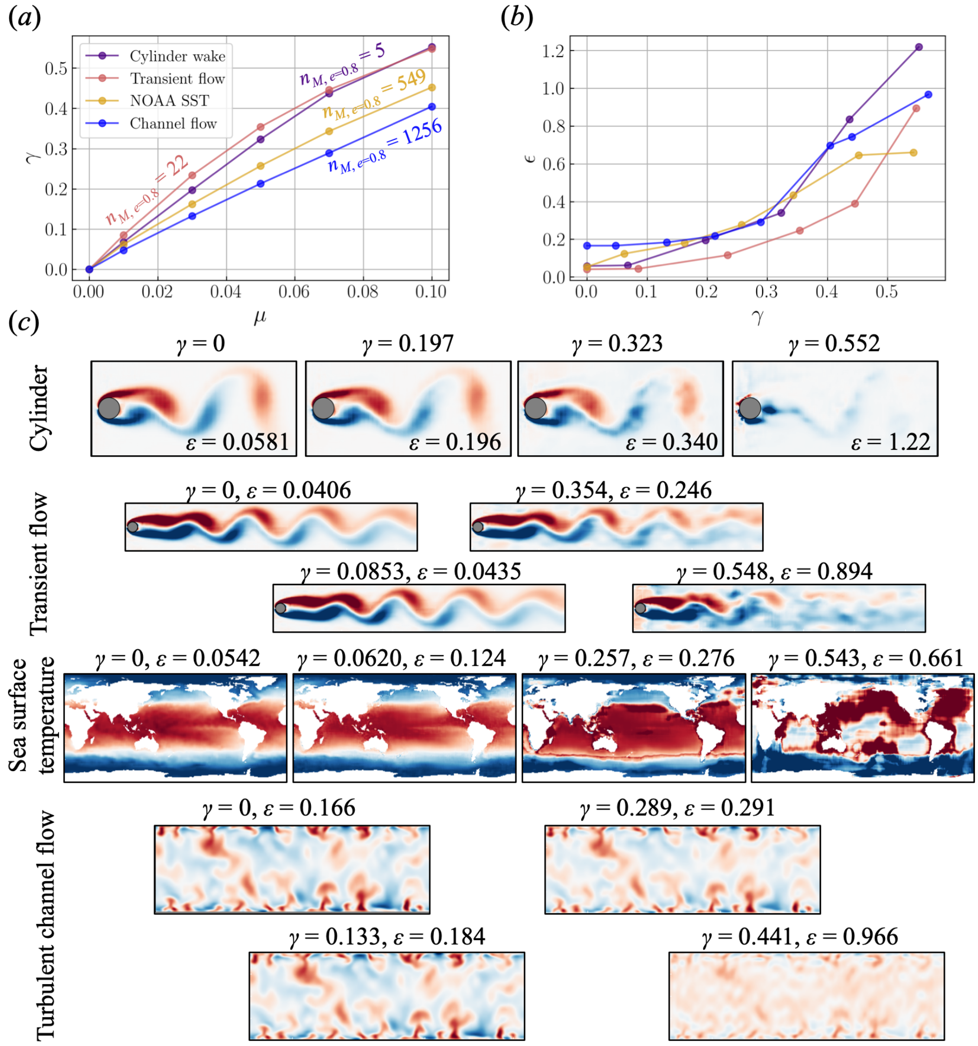


Figure 12: Pruning for AEs with fluid flows. (a) Relationship between a sparsity threshold coefficient μ and a sparsity factor $\gamma = 1-n_w/n_{w,\mathrm{original}}$. The number of singular modes which achieves 80% cumulative energy $n_{M,\ e=0.8}$ is also shown. (b) Relationship between a sparsity factor γ and L_2 error norm ϵ. (c) Representative flow fields of each example.

difference of weight distribution in the AEs that the contribution of the weights in the turbulence example for reconstruction is distributed more equally to the entire system. In contrast, for the laminar cases, i.e., the periodic shedding and transient flow, a fewer number of weights has the higher magnitudes and a contribution for reconstruction than the turbulence case. The curve of sea surface temperature model shows the intermediate behavior between them. It is also striking that the aforementioned trend is analogous to the singular value spectra in figure 6. The relationship between a sparsity factor γ and L_2 error norm with some representative flow fields are shown in figures 12(b) and (c). As presented, the reasonable recovery can be performed up to approximately 10 to 20% pruning. For reader's information, the number of singular modes which achieves 80% cumulative energy $n_{M,\ e=0.8}$ is presented in figure 12(a). Users can decide the coefficient γ to attain a target error value depending on target flows. Summarizing above, the efficient order reduction by AE can be carried out by considering the considerable methods and parameters as introduced through the paper.

5. Concluding remarks

We presented the assessment of neural network based model order reduction method, i.e., autoencoder (AE), for fluid flows. The present AE which comprises of a convolutional neural network and multi-layer perceptrons was applied to four data sets, i.e., two-dimensional cylinder wake, transient process, NOAA sea surface temperature, and turbulent channel flow. The model was evaluated in terms of various considerable parameters in deciding a construction of AE. With regard to the first investigation for the number of latent modes, it was clearly seen that the mapping ability of AE highly depends on the target flows, i.e., complexity. In addition, the first assessment enables us to notice the importance of investigation for the choice of activation function and the effect of number of weights contained in the AE. Motivated by the first assessment, the choice of activation function was then considered. We found that we need to be careful for decision of activation functions at hidden layers of AE so as to achieve the effective order reduction due to the observation that influence of activation functions highly depend on target flows. At last, the dependence of the reconstruction accuracy by the AE on the number of weights contained in the model was assessed. We exhibited that care should be taken to change the amount of parameters of AE models because we may encounter the problem of weight updates by using many numbers of weights. The use of pruning was also examined as one of the considerable candidates to reduce the number of weights. Although it has been known that the ability of neural networks may be improved by re-training of pruned neural networks [76], the possibility of this view will be tackled in future.

Although the strength of the AE could be seen through the investigation, we should also discuss remaining issues of the present form of the AE. One of them is the interpretability of latent space. Since the AE-based modes are not orthogonal with each other, it is still hard to understand the role of each latent vector for reconstruction [36]. To tackle this issue, Murata et al. [41] proposed a customized CNN-AE which can visualize modes obtained by the AE, considering a laminar cylinder wake and its transient process. However, they also reported that the applications to flows, in which requires a lot of spatial modes to present an energetic representation, e.g., turbulence, with their formulation is still challenging. This is because the structure of the proposed AE would be more complicated with increasing the complexity of target problems and this eventually leads to difficulty in terms of interpretability. The toughness of applications to turbulence could be also found in this paper as discussed above. More recently, Fukami et al. [72] attempted to use a concept of hierarchical AE so as to handle turbulent flows efficiently. Although the aforementioned challenges are just example, we can expect more efficient and elegant model order reductions with additional AE designs.

Acknowledgement

This work was supported from Japan Society for the Promotion of Science (KAKENHI grant number: 18H03758).

References

[1] J. L. Lumley, The structure of inhomogeneous turbulent flows, in: A. M. Yaglom, V. I. Tatarski (Eds.), Atmospheric turbulence and radio wave propagation, Nauka, 1967.

[2] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28.

[3] B. J. McKeon, A. S. Sharma, A critical-layer framework for turbulent pipe flow, J. Fluid Mech. 658 (2010) 336–382.

[4] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, L. S. Ukeiley, Modal analysis of fluid flows: An overview, AIAA J. 55 (12) (2017) 4013–4041.

[5] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. Dawson, C.-A. Yeh, Modal analysis of fluid flows: Applications and outlook, AIAA J. 58 (3) (2020) 998–1022.

[6] B. R. Noack, P. Papas, P. A. Monkewitz, The need for a pressure-term representation in empirical galerkin models of incompressible shear flows, J. Fluid Mech. 523 (2005) 339–365.

[7] S. Bagheri, Koopman-mode decomposition of the cylinder wake, J. Fluid Mech. 726 (2013) 596–623.

[8] Q. Liu, B. An, M. Nohmi, M. Obuchi, K. Taira, Core-pressure alleviation for a wall-normal vortex by active flow control, J. Fluid Mech. 853 (2018) R1.

[9] P. J. Schmid, Application of the dynamic mode decomposition to experimental data, Exp. Fluids 50(4) (2011) 1123–1130.

[10] C.-A. Yeh, K. Taira, Resolvent-analysis-based design of airfoil separation control, J. Fluid Mech. 867 (2019) 572–610.

[11] S. Nakashima, K. Fukagata, M. Luhar, Assessment of suboptimal control for turbulent skin friction reduction via resolvent analysis, J. Fluid Mech. 828 (2017) 496–526.

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

[13] M. P. Brenner, J. D. Eldredge, J. B. Freund, Perspective on machine learning for advancing fluid mechanics, Phys. Rev. Fluids 4 (2019) 100501.

[14] J. N. Kutz, Deep learning in fluid dynamics, J. Fluid Mech. 814 (2017) 1–4.

[15] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid. Mech. 51 (2019) 357–377.

[16] J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, J. Fluid Mech. 807 (2016) 155–166.

[17] M. Gamahara, Y. Hattori, Searching for turbulence models by artificial neural network, Phys. Rev. Fluids 2 (5) (2017) 054604.

[18] R. Maulik, O. San, A neural network approach for the blind deconvolution of turbulent flows, J. Fluid Mech. 831 (2017) 151–181.

[19] R. Maulik, O. San, J. D. Jacob, C. Crick, Sub-grid scale model classification and blending through deep learning, J. Fluid Mech. 870 (2019) 784–812.

[20] R. Maulik, O. San, A. Rasheed, P. Vedula, Subgrid modelling for two-dimensional turbulence using neural networks, J. Fluid Mech. 858 (2019) 122–144.

[21] X. Yang, S. Zafar, J.-X. Wang, H. Xiao, Predictive large-eddy-simulation wall modeling via physics-informed neural networks, Phys. Rev. Fluids 4 (3) (2019) 034602.

[22] J.-L. Wu, H. Xiao, E. Paterson, Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework, Phys. Rev. Fluids 3 (7) (2018) 074602.

[23] K. Fukami, Y. Nabae, K. Kawai, K. Fukagata, Synthetic turbulent inflow generator using machine learning, Phys. Rev. Fluids 4 (2019) 064603.

[24] H. Salehipour, W. R. Peltier, Deep learning of mixing by two 'atoms' of stratified turbulence, J. Fluid Mech. 861 (2019) R4.

[25] S. Cai, S. Zhou, C. Xu, Q. Gao, Dense motion estimation of particle images via a convolutional neural network, Exp. Fluids 60 (2019) 60–73.

[26] K. Fukami, K. Fukagata, K. Taira, Super-resolution reconstruction of turbulent flows with machine learning, J. Fluid Mech. 870 (2019) 106–120.

[27] K. Fukami, K. Fukagata, K. Taira, Super-resolution analysis with machine learning for low-resolution flow data, in: 11th International Symposium on Turbulence and Shear Flow Phenomena (TSFP11), Southampton, UK, no. 208, 2019.

[28] J. Huang, H. Liu, W. Cai, Online in situ prediction of 3-D flame evolution from its history 2-D projections via deep learning, J. Fluid Mech. 875.

[29] M. Morimoto, K. Fukami, K. Fukagata, Experimental velocity data estimation for imperfect particle images using machine learning, arXiv:2005.00756.

[30] K. Fukami, K. Fukagata, K. Taira, Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows, J. Fluid Mech., accepted (2021).

[31] O. San, R. Maulik, Extreme learning machine for reduced order modeling of turbulent geophysical flows, Phys. Rev. E 97 (2018) 04322.

[32] R. Maulik, K. Fukami, N. Ramachandra, K. Fukagata, K. Taira, Probabilistic neural networks for fluid flow surrogate modeling and data recovery, Phys. Rev. Fluids 5 (2020) 104401.

[33] P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, R. Vinuesa, Predictions of turbulent shear flows using deep neural networks, Phys. Rev. Fluids 4 (2019) 054603.

[34] M. Milano, P. Koumoutsakos, Neural network modeling for near wall turbulent flow, J. Comput. Phys. 182 (2002) 1–26.

[35] K. Fukami, K. Fukagata, K. Taira, Assessment of supervised machine learning for fluid flows, Theor. Comp. Fluid Dyn. 34 (4) (2020) 497–519.

[36] N. Omata, S. Shirayama, A novel method of low-dimensional representation for temporal behavior of flow fields using deep autoencoder, AIP Adv. 9 (1) (2019) 015006.

[37] K. Hasegawa, K. Fukami, T. Murata, K. Fukagata, Data-driven reduced order modeling of flows around two-dimensional bluff bodies of various shapes, ASME-JSME-KSME Joint Fluids Engineering Conference, San Francisco, USA (Paper 5079).

[38] K. Hasegawa, K. Fukami, T. Murata, K. Fukagata, Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes, Theor. Comp. Fluid Dyn. 34 (4) (2020) 367–388.

[39] K. Hasegawa, K. Fukami, T. Murata, K. Fukagata, Cnn-lstm based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different reynolds numbers, Fluid. Dyn. Res. `doi:10.1088/1873-7005/abb91d`.

[40] T. Nakamura, K. Fukami, K. Hasegawa, Y. Nabae, K. Fukagata, Extension of cnn-lstm based reduced order surrogate for minimal turbulent channel flow, arXiv preprint, arXiv:2010.13351.

[41] T. Murata, K. Fukami, K. Fukagata, Nonlinear mode decomposition with convolutional neural networks for fluid dynamics, J. Fluid Mech. 882 (2020) A13.

[42] N. B. Erichson, M. Muehlebach, M. W. Mahoney, Physics-informed autoencoders for lyapunov-stable fluid flow prediction, arXiv:1905.10866.

[43] K. T. Carlberg, A. Jameson, M. J. Kochenderfer, J. Morton, L. Peng, F. D. Witherden, Recovering missing CFD data for high-order discretizations using deep neural networks and dynamics learning, J. Comput. Phys. 395 (2019) 105–124.

[44] J. Xu, K. Duraisamy, Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics, arXiv:1912.11114.

[45] R. Maulik, B. Lusch, P. Balaprakash, Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders, arXiv:2002.00470.

[46] Y. Liu, C. Ponce, S. L. Brunton, J. N. Kutz, Multiresolution convolutional autoencoders, arXiv:2004.04946.

[47] K. Fukami, T. Murata, K. Fukagata, Sparse identification of nonlinear dynamics with low-dimensionalized flow representations, arXiv preprint, arXiv:2010.12177.

[48] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[49] S. Lee, D. You, Data-driven prediction of unsteady flow fields over a circular cylinder using deep learning, J. Fluid Mech. 879 (2019) 217–254.

[50] J. Kim, C. Lee, Prediction of turbulent heat transfer using convolutional neural networks, J. Fluid Mech. 882 (2020) A18.

[51] S. Lee, D. You, Mechanisms of a convolutional neural network for learning three-dimensional unsteady wake flow, arXiv:1909.06042.

[52] Q. Le, J. Ngiam, Z. Chen, D. Chia, P. Koh, Tiled convolutional neural networks, Adv. Neural Inform. Proc. Syst. 23 (2010) 1279–1287.

[53] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980

[54] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagation errors, Nature 322 (1986) 533—536.

[55] P. Domingos, A few useful things to know about machine learning, Communications of the ACM 55 (10) (2012) 78–87.

[56] H. F. S. Lui, W. R. Wolf, Construction of reduced-order models for fluid flows using deep feedforward neural networks, J. Fluid Mech. 872 (2019) 963–994.

[57] J. Yu, J. S. Hesthaven, Flowfield reconstruction method using artificial neural network, AIAA J. 57(2) (2019) 482–498.

[58] L. Prechelt, Automatic early stopping using cross validation: quantifying the criteria, Neural Netw. 11 (4) (1998) 761–767.

[59] P. Baldi, K. Hornik, Neural networks and principal component analysis: Learning from examples without local minima, Neural Netw. 2 (1) (1989) 53–58.

[60] H. Bourlard, Y. Kamp, Auto-association by multilayer perceptrons and singular value decomposition, Biological cybernetics 59 (4-5) (1988) 291–294.

[61] E. Oja, Simplified neuron model as a principal component analyzer, Journal of mathematical biology 15 (3) (1982) 267–273.

[62] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, Activation functions: Comparison of trends in practice and research for deep learning, arXiv:1811.03378.

[63] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, Proc. Int. Conf. Mach. Learn. (2010) 807–814.

[64] J. Bergstra, D. Yamins, D. Cox, Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, International Conference on Machine Learning (2013) 115–123.

[65] E. Brochu, V. Cora, de N. Freitas, A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, Technical Report TR-2009-023, University of British Columbia.

[66] R. Maulik, A. Mohan, B. Lusch, S. Madireddy, P. Balaprakash, Time-series learning of latent-space dynamics for reduced-order model closure, Physica D: Nonlinear Phenomena 405 (2020) 132368.

[67] S. L. Brunton, J. N. Kutz, Data-driven science and engineering: Machine learning, dynamical systems, and control, Cambridge University Press, 2019.

[68] H. Kor, M. Badri Ghomizad, K. Fukagata, A unified interpolation stencil for ghost-cell immersed boundary method for flow around complex geometries, J. Fluid Sci. Technol. 12 (1) (2017) JFST0011. doi:10.1299/jfst.2017jfst0011.

[69] Available on https://www.esrl.noaa.gov/psd/.

[70] K. Fukagata, N. Kasagi, P. Koumoutsakos, A theoretical prediction of friction drag reduction in turbulent flow by superhydrophobic surfaces, Phys. Fluids 18 (2006) 051703.

[71] K. Taira, H. Nakao, Phase-response analysis of synchronization for periodic flows, J. Fluid Mech. 846 (2018) R2.

[72] K. Fukami, T. Nakamura, K. Fukagata, Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data, Phys. Fluids 32 (2020) 095110.

[73] I. Scherl, B. Storm, J. K. Shang, O. Williams, B. L. Polagye, S. L. Brunton, Robust principal component analysis for modal decomposition of corrupt fluid flows, Phys. Rev. Fluids 5 (2020) 054401.

[74] P. Ramachandran, B. Zoph, Q. V. Le, Searching for activation functions, arXiv:1710.05941.

[75] S. Han, H. Mao, W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, arXiv:1510.00149.

[76] S. Han, J. Pool, J. Tran, W. Dally, Learning both weights and connections for efficient neural network, in: Advances in neural information processing systems, 2015, pp. 1135–1143.