

## Research Article

# Reduced-Order Modeling of Cavity Flow Oscillations across Multi-Mach Numbers Using Deep Learning

Zhe Liu <sup>1</sup>, Fangli Ning <sup>1</sup>, Hui Ding <sup>1</sup>, Qingbo Zhai <sup>1</sup> and Juan Wei <sup>2</sup>

<sup>1</sup>School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China

<sup>2</sup>School of Communication Engineering, Xidian University, Xi'an 710071, China

Correspondence should be addressed to Fangli Ning; [ningfl@nwpu.edu.cn](mailto:ningfl@nwpu.edu.cn)

Received 23 February 2021; Revised 17 June 2021; Accepted 7 July 2021; Published 26 July 2021

Academic Editor: Fehmi Najar

Copyright © 2021 Zhe Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The reduced-order model can accurately and efficiently predict unsteady problems in many aerospace engineering applications. The traditional reduced-order model based on proper orthogonal decomposition (POD) and Galerkin projection has poor robustness and large error in predicting complex problems. In this paper, a reduced-order model combining POD and deep learning is proposed to predict cavity flow oscillations under different flow conditions. Firstly, POD modes and corresponding coefficients are obtained by POD. Then, two deep learning frameworks, including multilayer perceptron (MLP) and long short-term memory (LSTM) neural networks, are used to predict the future POD coefficients, respectively. Finally, the cavity flow oscillations across multi-Mach numbers are predicted by the POD modes and the future coefficients. The results show that both of these frameworks can accurately predict cavity flow oscillations when the flow conditions change, and the time cost is reduced by order of magnitude. In addition, due to the performance of LSTM is better than that of MLP, its calculation speed is faster.

## 1. Introduction

Cavity flow oscillations exist in many aerospace engineering fields [1–4], such as weapon bays [5, 6] and landing gears [7, 8]. The physical mechanism in a cavity is complex. The shear layer above the cavity generates a vortex, which collides with the trailing wall to generate sound waves. The generated sound waves radiate forward and continue to excite the shear layer to generate new vortices. This process causes intense pressure oscillations in the cavity. The study of oscillation characteristics in the cavity is helpful to understand the mechanism of cavity noise and suppress cavity noise. Therefore, the cavity flow oscillation issue has received more and more attention [9–11]. The experimental investigations of cavity flow oscillations are usually carried out in wind tunnels or water tunnels. The maintenance of experimental equipment and the complexity of working conditions require many costs, while computational fluid dynamics (CFD) can effectively solve this problem. It not only fundamentally changes the design process of aerospace

vehicles but also effectively reduces the number of experiments and deeply understands the physical mechanism.

CFD numerical simulation mainly includes direct numerical simulation (DNS) [12–14], Reynolds averaged Navier–Stokes (RANS) [15, 16], and large eddy simulation (LES) [17–19]. DNS is a direct high-fidelity method for solving Navier–Stokes equations, which requires very fine grids. Therefore, when the flow conditions change, the calculation cost will increase sharply.

In order to effectively reduce the calculation cost in engineering practice, the research of reduced-order model (ROM) has been widely concerned. Since the 1990s, researchers [20, 21] have developed a variety of unsteady ROMs. On the one hand, ROM can save expensive calculation costs. On the other hand, it can effectively extract the main characteristics of the flow field and provide a theoretical basis for analyzing the mechanism and oscillation characteristics of complex systems.

Proper orthogonal decomposition (POD) is an efficient method for establishing a ROM. Rowley et al. [12] and

Golerfelt [22] accurately predicted the oscillation in a cavity that has a ratio of  $L/D = 2$  by POD and Galerkin projection. POD modes were obtained by extracting a series of orthogonal bases, and the high-dimensional governing equations were transformed into the low-dimensional ordinary differential equations by the Galerkin projection.

The traditional method applying POD and Galerkin projection can establish the ROM, but the robustness of the method is poor [23–25]. In order to effectively solve nonlinear problems, deep learning is introduced in this work. As an important branch of machine learning, deep learning [26] has been widely recognized and developed in many applications [27–32], such as speech recognition [27, 28] and image processing [29, 30]. The excellent ability to process big data and nonlinear relationships makes deep learning develop rapidly in the fluid mechanics for recent several years. In 2016, Ling et al. [33] realized deep learning of RANS turbulence model by embedding Galileo invariants into the deep neural network structure and predicted the channel flow and separated flow. It is considered the first combination of deep learning and fluid mechanics [34]. Miyanawala and Jaiman [35] realized the prediction of flow characteristics in the wake region of a two-dimensional cylinder by convolution neural network (CNN). Lee and You [36] applied generated antagonistic network (GAN) to predict the laminar vortex shedding over a cylinder.

In this work, ROMs are based on multilayer perceptron (MLP) and long short-term memory (LSTM) neural network for predicting cavity flow oscillations when the Mach number changes are established, respectively. Although there is some literature on establishing the ROM by deep learning, they are mainly applied to simple incompressible flows. For example, Yu and Hesthaven [23] proposed a flow reconstruction method based on POD and artificial neural network (ANN). They validated the efficiency of this approach in two-dimensional viscous nozzle flows, an inviscid M6 wing flow, a viscous hypersonic flow of a complex configuration, and an unsteady two-dimensional Riemann problem. San et al. [25] also proposed an efficient framework based on POD and ANN and accurately predicted a nonlinear wave-propagation problem. The compressible cavity flow is extremely unsteady and complex. There are intense flow oscillations in the cavity. There are complex nonlinear interactions among the shear layer, vortex, sound waves, and cavity walls. Even a small change in Mach number will make the internal oscillation of the cavity change unpredictably. It is meaningful to establish the ROM of cavity flow oscillations across multi-Mach numbers based on deep learning.

In addition to the approach proposed in this paper, nonlinear approaches (such as autoencoders) can be the substitute for POD, but the establishment of the ROM is completely different. The ROM proposed in this paper can obtain the flow field distributions of the multiple physical quantities through the predicted POD coefficients. However, nonlinear approaches can only obtain the flow field distributions of the input physical quantity. We need to readjust the hyperparameters and retrain the neural network if we

want to obtain the flow field distributions of other physical quantities. Furthermore, although POD is a linear approach, the loss of efficiency in the intrusive ROM frameworks can be compensated by nonintrusive reduced-order models (NIROMs) [37–39]. Therefore, we did not try to use nonlinear approaches for the approximation of the solution space.

The paper is organized as follows: Section 2 introduces the basic mathematical process of POD. The theory and characteristics of MLP and LSTM neural networks are described in Section 3 and Section 4. Moreover, Section 5 discusses POD results and the predicted results applied MLP and LSTM neural networks, respectively. Finally, conclusions are summarized in Section 6.

## 2. Proper Orthogonal Decomposition

The snapshots POD method [40] is applied in this work. The snapshots  $\mathbf{q}(\mathbf{x}, t_k) = \{\mathbf{q}_k(\mathbf{x}) \mid k = 1, 2, \dots, N\}$ ,  $\mathbf{q} = [\mathbf{u}, \mathbf{v}, c]$ , where  $c$  is the sound speed. The snapshots are divided into the average quantity and the pulse quantity:

$$\mathbf{q}_k(\mathbf{x}) = \bar{\mathbf{q}}(\mathbf{x}) + \mathbf{q}'_k(\mathbf{x}), \quad (1)$$

where  $\bar{\mathbf{q}}(\mathbf{x}) = (1/N) \sum_{k=1}^N \mathbf{q}_k(\mathbf{x})$ . We hope to obtain a set of optimal orthogonal bases:

$$\boldsymbol{\varphi}(\mathbf{x}) = \{\boldsymbol{\varphi}_i(\mathbf{x}) \mid i = 1, 2, \dots, M, M \leq N\}. \quad (2)$$

The  $k$ th pulse quantity can be approximated as

$$\mathbf{q}'_k(\mathbf{x}) \approx \sum_{i=1}^M a_i(t_k) \boldsymbol{\varphi}_i(\mathbf{x}), \quad (3)$$

where  $a_i(t_k)$  is the coefficient of the  $i$ th POD mode constituting the  $k$ th snapshot.

The snapshot  $\mathbf{q}'_k(\mathbf{x})$  and the basis function  $\boldsymbol{\varphi}_i(\mathbf{x})$  are in the same spatial domain, so  $\boldsymbol{\varphi}_i(\mathbf{x})$  can be represented by a linear combination of all snapshots:

$$\boldsymbol{\varphi}_i(\mathbf{x}) = \sum_{k=1}^N A_i(t_k) \mathbf{q}'_k(\mathbf{x}), \quad (4)$$

where  $A_i(t_k)$  is a complex coefficient. In order to obtain the orthogonal basis  $\boldsymbol{\varphi}_i(\mathbf{x})$ , the above problem can be converted into the eigenvalue problem [41]:

$$\mathbf{C}_{mn} \mathbf{A}_i = \lambda_i \mathbf{A}_i, \quad (5)$$

where  $\mathbf{A}_i$  is the coefficients matrix,  $\mathbf{A}_i = [A_i(t_1), A_i(t_2), \dots, A_i(t_N)]$ ,  $\lambda_i$  is the eigenvalue of the  $i$ th mode, and  $\mathbf{C}_{mn}$  is a  $N \times N$  self-adjoint matrix ( $m, n = 1, 2, \dots, N$ ), which is calculated by

$$\mathbf{C}_{mn} = \frac{1}{N} \langle \mathbf{q}'_m(\mathbf{x}), \mathbf{q}'_n(\mathbf{x}) \rangle. \quad (6)$$

The inner product of energy based on isentropic assumption is applied as follows [42, 43]:

$$\langle \mathbf{q}'_m, \mathbf{q}'_n \rangle = \int_S \left( \mathbf{u}'_m \mathbf{u}'_n + \mathbf{v}'_m \mathbf{v}'_n + \frac{2}{\gamma(\gamma-1)} \mathbf{c}'_m \mathbf{c}'_n \right) dS. \quad (7)$$

The eigenvalue decomposition of the matrix  $\mathbf{C}_{mm}$  is obtained. The eigenvector matrix is  $\mathbf{A}$ , and the eigenvalues  $\lambda_i$  are arranged in descending order.

### 3. Multilayer Perceptron

Artificial neural networks are often referred to as neural networks or multilayer perceptrons. Multilayer perceptron, also known as multilayer perceptron neural network, is a traditional supervised learning method that simulates human neurons [44]. When dealing with the nonlinear regression problem, it can approach the actual mapping relationship between input feature space  $X$  and output tag vector  $Y$  infinitely by learning nonlinear functions. Perceptron is a single neuron model, which is the precursor of the large neural network [45]. The power of neural networks lies in their ability to represent the training data and how to relate it to the output variables you want to predict [46]. Mathematically, they can learn any mapping function and have been proved to be a general approximation algorithm [47]. The prediction ability of a neural network comes from the hierarchical or multilayer structure of the network [48]. Data structures can select (learn to represent) features of different scales or resolutions and combine them into higher-order features [49].

The basic MLP structure includes the input layer, hidden layer, and output layer, in which the number of the hidden layer can be more or less [45]. Each layer is composed of many nodes. The node of each layer is a neuron. Except that the input layer needs to deal with the input characteristics, the other neurons have a nonlinear activation function, and all neurons are fully connected with the next layer. A simple MLP framework with a single hidden layer is shown in Figure 1. There are  $N$  nodes in the hidden layer. The POD coefficient of every POD mode at the current time instant  $a_i(t_k)$  is regarded as the input layer. The POD coefficient of each POD mode at the next time instant  $a_i(t_{k+1})$  is regarded as the output layer. The mathematical formula can be expressed by

$$a_i(t_{k+1}) = f(a_i(t_k); W, c, w, b) = w^T \max\{0, W^T x + c\} + b, \quad (8)$$

where  $W$  and  $w$  is the weight in the mapping from the input layer to the hidden layer and from the hidden layer to the output layer, respectively.  $c$  and  $b$  is the bias in the mapping from the input layer to the hidden layer and from the hidden layer to the output layer, respectively.

The rectified linear unit (ReLU) is selected as the activation function. ReLU is the most common activation function at present, which can effectively solve the gradient disappearance problem of sigmoid and tanh, and its convergence speed is much higher than that of sigmoid and tanh [50]. The superiority of ReLU has been proved in many studies [51, 52]. The training of the neural network is based on the backpropagation algorithm. The parameters of each

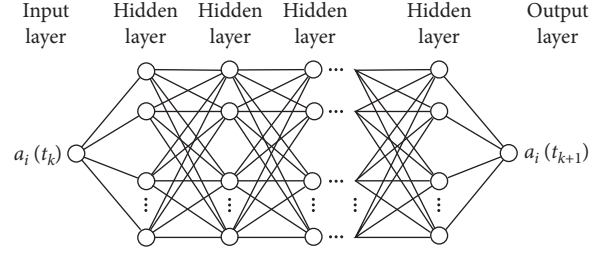


FIGURE 1: MLP architecture.

node are updated layer by layer in real time. In addition, the adaptive moment estimation (Adam) algorithm [53] is used to find the optimal solution of the weight and bias, which is an excellent gradient optimization algorithm. Its main advantage is to use the same learning rate for each parameter and adapt independently as the learning progresses [54]. The loss function is the mean-absolute-error. Comparisons of the activation functions and optimizers are shown in Figure 2. It can be found that the results predicted by the framework with the ReLU activation function and the Adam optimizer are the most accurate. Although the results by the ReLU and tanh are almost the same, the computational speed by ReLU is faster than that by tanh. Therefore, the ReLU activation function and Adam optimizer are used in this work.

### 4. Long Short-Term Memory Neural Network

LSTM is a special kind of recurrent neural network (RNN), which is good at processing time sequence data. The traditional neural network cannot achieve continuous memory, but RNN can solve this problem. When the traditional RNN processes long sequences, RNN may face the problem that the gradient disappears or bursts [55].

LSTM can create a path that allows the gradient to flow sustainably for a long time by introducing controllable self-circulation. The architecture of the LSTM cell is shown in Figure 3. The reason why LSTM can remember long-term information lies in the design of the gate structure, which is a way to allow information to pass selectively. The LSTM cell contains the forget gate, input gate, and output gate. The specific mathematical process [56, 57] is as follows.

The input of the forgetting gate is the output of the above layer  $h_{t-1}$  and the sequence data  $x_t$ . The output  $f_t$  is obtained by a sigmoid activation function. The output value is in the range of  $[0, 1]$ , which indicates the probability that the state of the cell in the previous layer will be forgotten, 1 is completely reserved, and 0 is completely abandoned:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f). \quad (9)$$

The next step is to decide what information we want to keep in the neuron cell, which consists of two parts. First, a sigmoid layer called the input gate layer determines the values we want to update. Then, a tanh layer generates a new candidate value,  $\hat{C}_t$ , which is added to the neuron state:

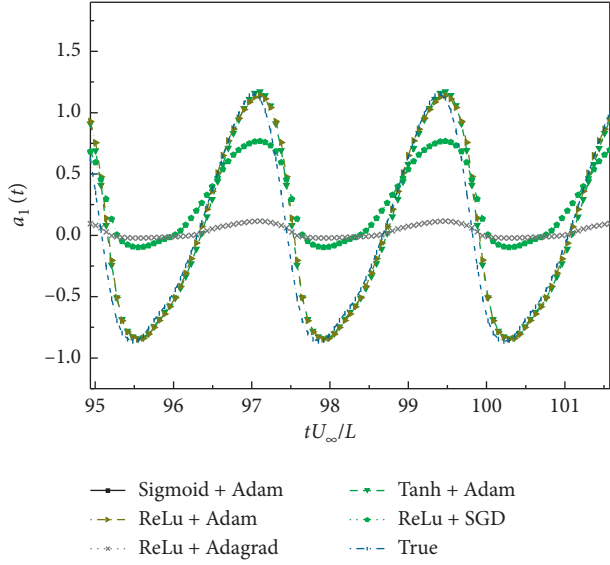


FIGURE 2: Comparisons of the activation functions and optimizers.

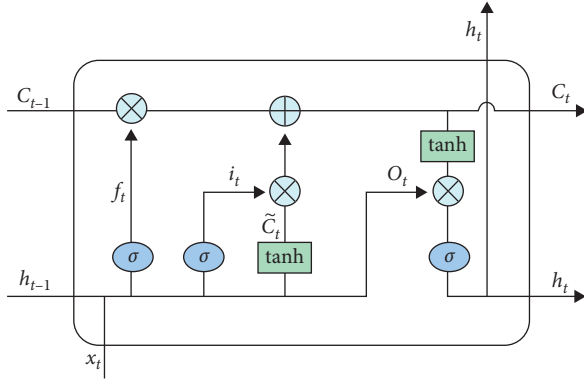


FIGURE 3: Architecture of LSTM cell.

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i), \\ \hat{C}_t &= \tanh(W_C[h_{t-1}, x_t] + b_C). \end{aligned} \quad (10)$$

In order to obtain the current cell state, the forgotten state of the last cell obtained through the forget gate is added to the new information after screening:

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t. \quad (11)$$

The output gate is used to control how much of the cell state of the layer is filtered. Firstly, the sigmoid activation function is used to get the output gate  $o_t$ . Then, the output  $h_t$  in the current layer is obtained by that the cell state  $C_t$  disposed by the tanh activation function multiplies with  $o_t$ :

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), \\ h_t &= o_t * \tanh(C_t). \end{aligned} \quad (12)$$

In the above formulas,  $\sigma$  and  $\tanh$  are the sigmoid and tanh activation functions, respectively, which are shown as

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (13)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (14)$$

## 5. Results and Discussion

**5.1. The Whole Architecture.** In this work, the cavity flow oscillations are numerically simulated by direct numerical simulation (DNS). The flow chart of the ROM for cavity flow oscillations is shown in Figure 4. The goal of this work is to combine POD and deep learning to predict cavity flow oscillations across multi-Mach numbers. The specific process of this method can be divided into five steps:

- (1) Select the cavity velocity fields at  $Ma = 0.51, 0.52$  to be used as the training datasets and velocity fields at  $Ma = 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59, 0.6$  to be used as the test datasets.
- (2) Achieve dominant POD modes and their corresponding POD coefficients of the training and test datasets, respectively. The POD coefficients of the test dataset will be compared with the true simulation results.
- (3) Train MLP/LSTM neural networks for the dominant POD modes.
- (4) Use the test dataset POD coefficients to predict the coefficient at the next time instant. Repeat this step for all the dominant POD modes.
- (5) Use POD modes and the MLP/LSTM predicted POD coefficients to reconstruct the velocity field.

**5.2. DNS Results.** The cavity flow oscillations across multi-Mach numbers are obtained by 2D DNS in the OpenFOAM open-source software. In our previous work [58], the simulation details and grid convergence have been described, and the accuracy of the simulation method has also been verified. The sonicFoam solver is applied, which is based on the PISO (pressure implicit with splitting of operator) algorithm. The time derivative discretization is the Euler scheme, the gradient discretization is the Gauss linear scheme, and the divergence discretization is the Gauss upwind scheme. The baseline rectangular cavity has a length-to-depth ratio of  $L/D = 4$  and  $L/W = 0.5$ ,  $p_\infty = 70422$  Pa,  $T_\infty = 294.5$  K, and  $Ma = 0.5$ ,  $Re_D = 5,000$ . The nonuniform mesh, which is dense near the cavity wall, is used. The number of grid points within the cavity is  $260 \times 200$  and  $312 \times 240$ , as shown in Figure 5. The Strouhal numbers are compared with the theoretical and experimental results, as shown in Table 1. The theoretical results are calculated by the modified Rossiter's formula [59]. The Strouhal numbers obtained by the present DNS are consistent with the theoretical and experimental results.

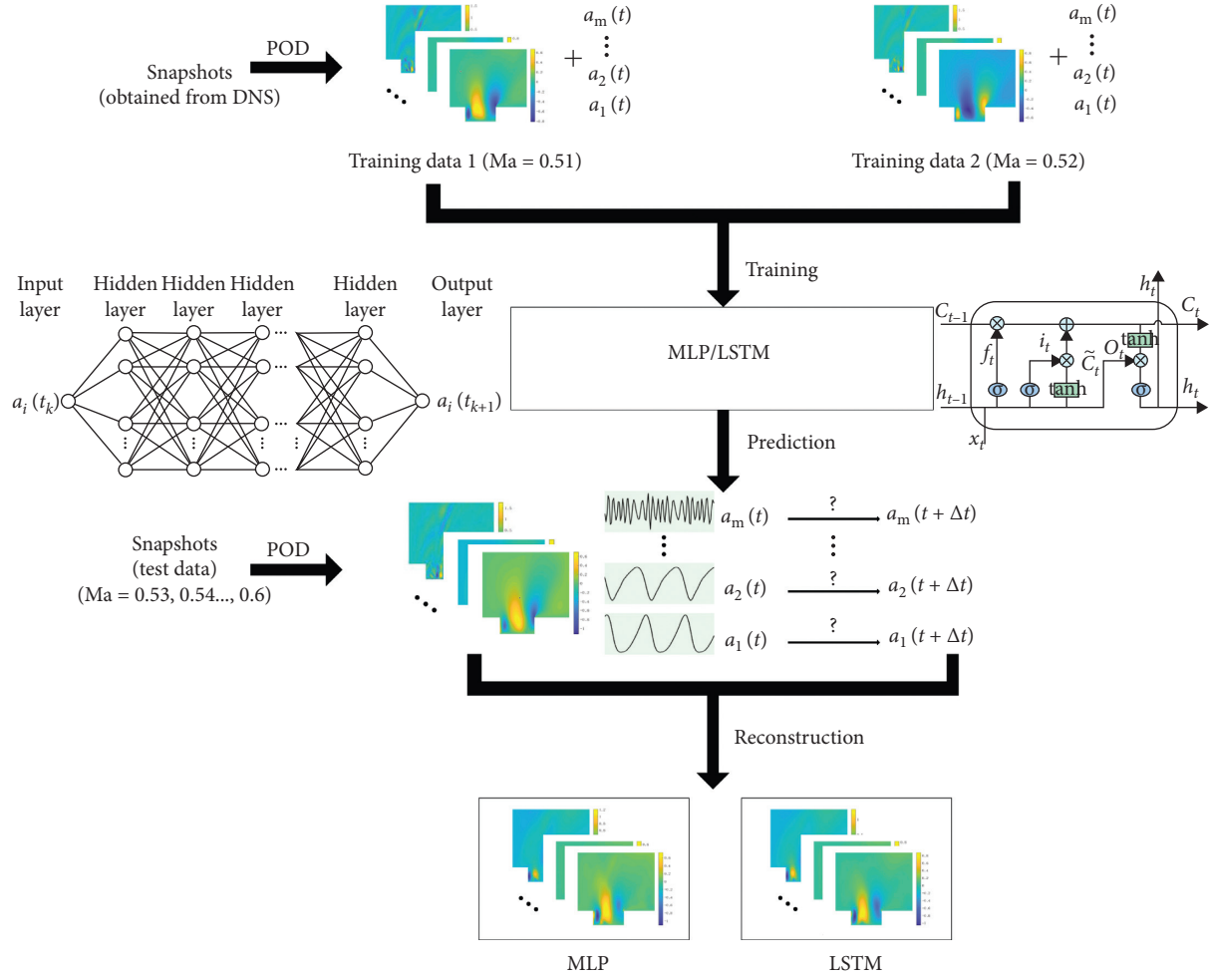


FIGURE 4: Flow chart of the reduced-order model using MLP or LSTM.

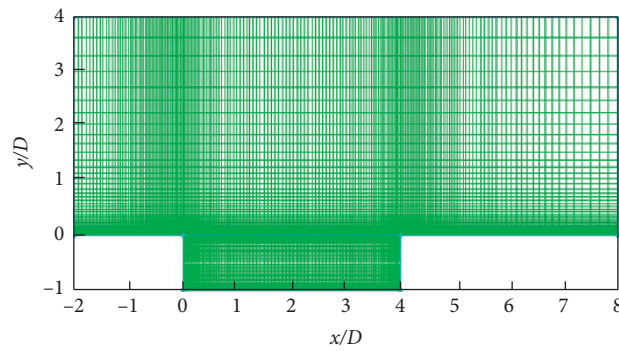


FIGURE 5: Cavity mesh structure [58].

TABLE 1: Comparison of Strouhal numbers with the theoretical and experimental results.

	1st	2nd	3rd	4th
Theory [59]	0.334	0.780	1.226	1.672
Experiment [60]	0.403	0.885	1.367	1.897
Present DNS	0.439	0.878	1.317	1.756



**5.3. POD Analysis.** The POD analysis was conducted on the velocity fields of cavity flow oscillations at different Mach numbers to extract the dominant POD modes and the corresponding coefficients. The eigenvalue  $\lambda_i$  reflects the energy that the POD mode holds. It is now generally accepted that the 99% energy is enough to reconstruct the flow field. The eigenvalues and the cumulative eigenvalues for cavity flow oscillations at  $Ma = 0.51, 0.6$  are shown in Figure 6. The low-order modes occupy the highest energy. The first two modes occupy 78% of the energy at a Mach number of 0.51 and 77% at a Mach number of 0.6. The eigenvalues at the lower orders are almost identical at the two Mach numbers, but there are differences at the higher orders. This is due to the increase of Mach number, the increase of airflow velocity in the cavity, and the more complicated flow structure. Whether  $Ma = 0.51$  or  $0.6$ , 13 POD modes already occupy 99% of the energy, so we extract the first 13 POD modes and their corresponding coefficients.

The most important requirement for deep learning is that the data has some common features. The totally different data cannot be well learned. In order to test the similarity of the POD modes at different Mach numbers, the first two POD modes at  $Ma = 0.51$  and  $Ma = 0.6$  are compared, as shown in Figure 7. POD mode 1 and mode 2 at two Mach numbers have three large-scale structures, which are located near the leading edge, the middle of the cavity, and the trailing edge, respectively. Mode 2 also has a small-scale structure at the trailing edge corner. Although these large-scale structures are different in size and direction, their position and shape are similar. The flow structures at different Mach numbers are qualitatively similar, but there are differences in size and direction. These differences are irregular and unpredictable.

#### 5.4. MLP and LSTM Frameworks

**5.4.1. Selection of Parameters.** In this part, we employ MLP and LSTM frameworks to predict the POD coefficient of the first POD mode. The performance of the two neural networks depends on the selection of parameters such as the number of layers, the number of nodes in the hidden layer, the number of iterations, and the learning rate. In order to achieve better performance, we manually adjust these parameters. The effects of these parameters on the model performance are shown in Tables 2 and 3, respectively. Increasing the number of nodes and layers of the hidden layer will improve the performance of the model, but it will also lead to overfitting of the training data. If the learning rate is too small, the calculation time will be too long. If the learning rate is too large, and the optimal solution may be missed. The number of iterations corresponds to the learning rate. If the learning rate is small, the number of iterations will be more, and if the learning rate is large, the number of iterations will be less. If the batch size is too large, the convergence speed of the model is slow. If the batch size is too small, the model will not converge.

In the MLP1, MLP2, MLP3 framework, when there are 10 nodes, the test loss is the smallest, and there is little

difference in the training loss. Therefore, the number of nodes is fixed at ten. The number of hidden layers is increased to 2 and 3 layers. The results show that the training loss of MLP5 is the smallest, and there is little difference in the test loss. In addition, with the increase or decrease of the learning rate, the test loss does not decrease. With the increase of iterations, the training loss and test loss will decrease slightly, but the calculation cost will increase. The optimal LSTM framework can also be determined by using the same single variable method. Therefore, considering the training loss, test loss, and calculation cost, we choose the network framework MLP5 and LSTM7.

**5.4.2. Predicted POD Coefficients.** The key to establishing the ROM is to predict POD coefficients accurately. Its accuracy directly affects the accuracy of the reconstructed velocity field across multi-Mach numbers. Figure 8 shows the time evolution curves of different modal coefficients achieved by the MLP and LSTM framework. Due to the limited space in the paper, we only list the first four POD modal coefficients across partial test Mach numbers.

The coefficients predicted by MLP and LSTM frameworks are very similar to the real coefficients. The predicted coefficients can well reflect the change of the POD coefficients. No matter when Mach numbers are 0.53, 0.57, 0.6, or even other Mach numbers not shown in this paper, the predicted coefficients are not much different from the real coefficients. However, it is worth noting that, in the case of higher order, the deviation between the predicted coefficient and the real coefficient will increase, which is determined by the high instability and nonlinearity of the higher-order modal coefficients. Because the energy occupied by higher-order POD modes is extremely low, it has little influence on the final velocity field reconstruction results. In addition, we also find that the coefficients predicted by the MLP framework have small fluctuations in some positions, which may be caused by the method only considering the information of the current time instant.

In order to analyze the training results in more detail, the training and test loss obtained by different frameworks are shown in Figure 9. In general, under the same number of iterations, the training loss and test loss of LSTM are smaller than those of MLP (except for the second modal coefficient), and the LSTM framework can achieve convergence with fewer iterations. This shows the superiority of the LSTM framework in dealing with time series problems. The comparison of the two frameworks will be described in Section 5.4.4.

**5.4.3. Velocity Field Reconstruction.** According to the predicted coefficients and POD modes, the reconstructed velocity field can be obtained. Figure 10 and Figure 11 show the velocity field simulated by DNS and predicted by two deep learning frameworks at  $Ma = 0.54$  and  $0.6$ . Both of these two deep learning frameworks can accurately capture large-scale structures in instantaneous velocity fields. The results of contours at other Mach numbers are similar. It shows that

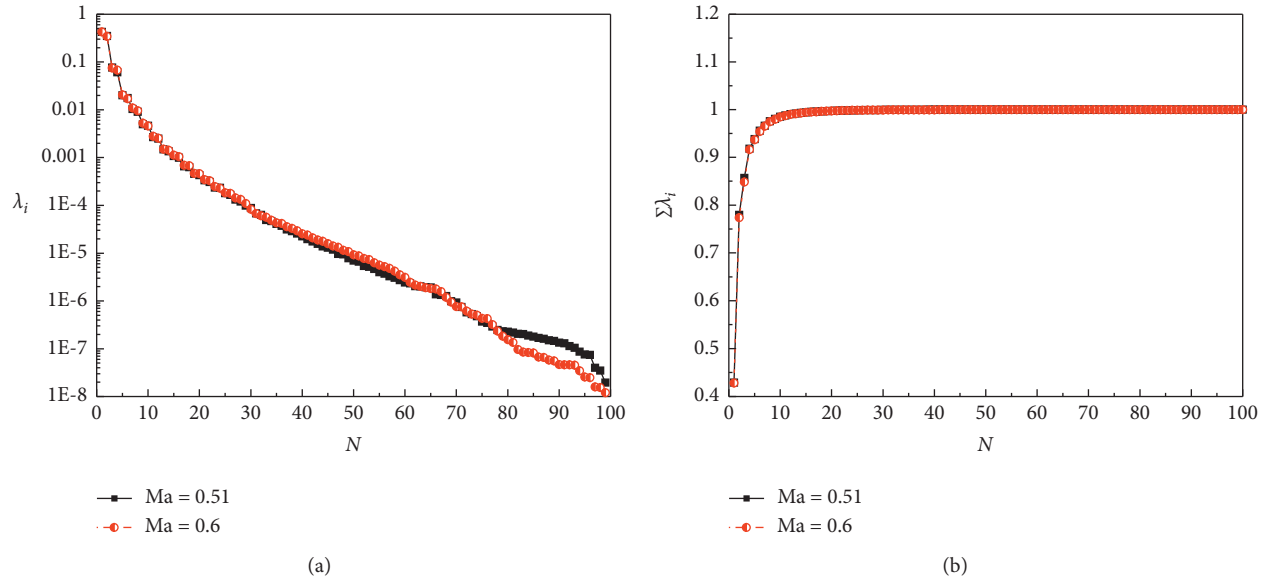


FIGURE 6: (a) Eigenvalues and (b) cumulative eigenvalues for cavity flow oscillations at  $Ma = 0.51, 0.6$ .

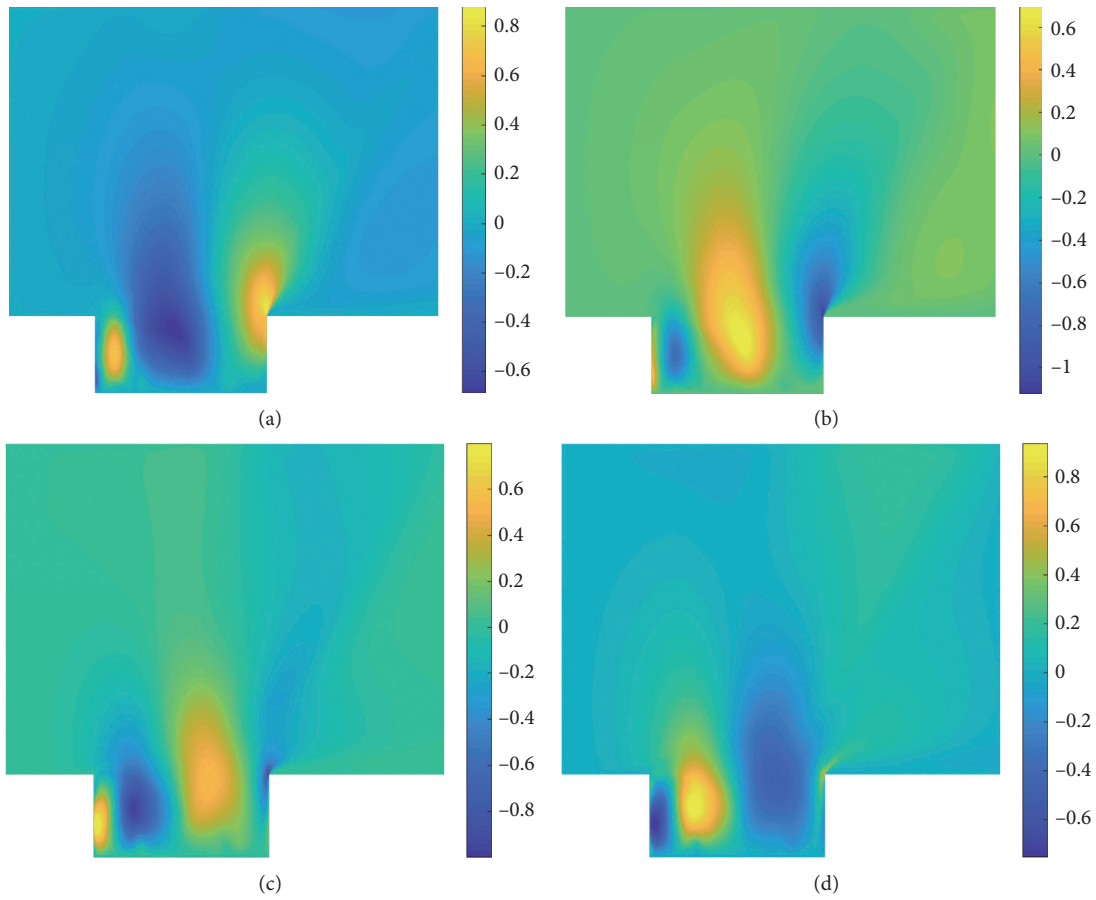


FIGURE 7: The first two POD modes at  $Ma = 0.51$  and  $Ma = 0.6$ . (a) Mode 1 at  $Ma = 0.51$ . (b) Mode 2 at  $Ma = 0.51$ . (c) Mode 1 at  $Ma = 0.6$ . (d) Mode 2 at  $Ma = 0.6$ .

TABLE 2: Training loss and test loss of MLP frameworks with different parameters.

Framework	Number of nodes	Number of hidden layers	Learning rate	Number of iterations	Training loss	Test loss
MLP1	10	1	0.001	5000	0.11768	0.11003
MLP2	20	1	0.001	5000	0.11705	0.11007
MLP3	50	1	0.001	5000	0.11722	0.11050
MLP4	10	2	0.001	5000	0.11069	0.10551
MLP5	10	3	0.001	5000	0.10836	0.10566
MLP6	10	3	0.0001	5000	0.11645	0.10973
MLP7	10	3	0.01	5000	0.10617	0.10939
MLP8	10	3	0.001	10000	0.10665	0.10487
MLP9	10	3	0.001	50000	0.10217	0.10835

TABLE 3: Training loss and test loss of LSTM frameworks with different parameters.

Framework	Number of nodes	Number of epochs	Batch size	Training loss	Test loss
LSTM1	10	150	32	0.11912	0.09644
LSTM2	20	150	32	0.11837	0.09507
LSTM3	50	150	32	0.11900	0.09697
LSTM4	20	50	32	0.13180	0.10413
LSTM5	20	300	32	0.11859	0.09799
LSTM6	20	150	8	0.11930	0.09703
LSTM7	20	150	64	0.12269	0.09310

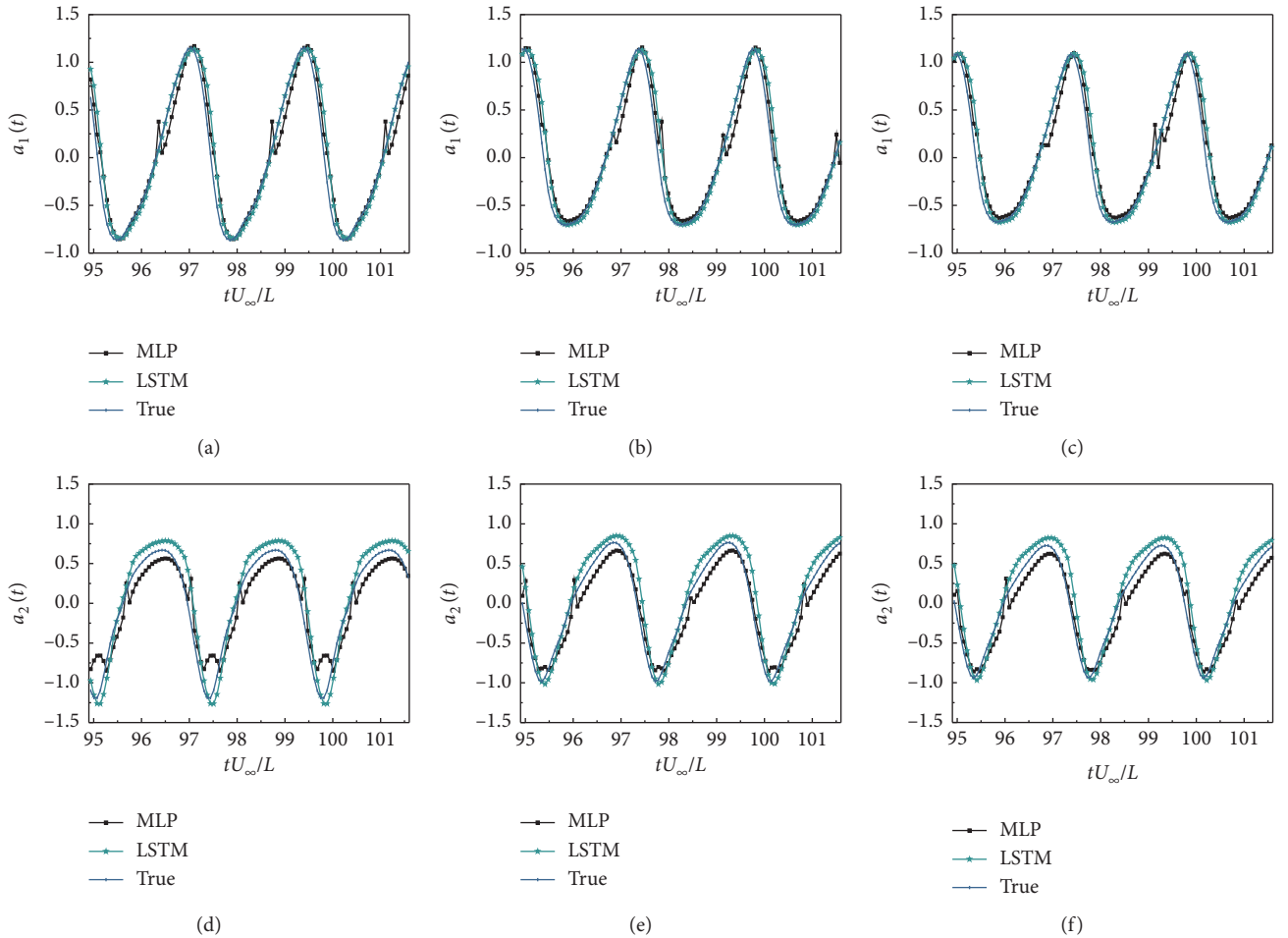


FIGURE 8: Continued.



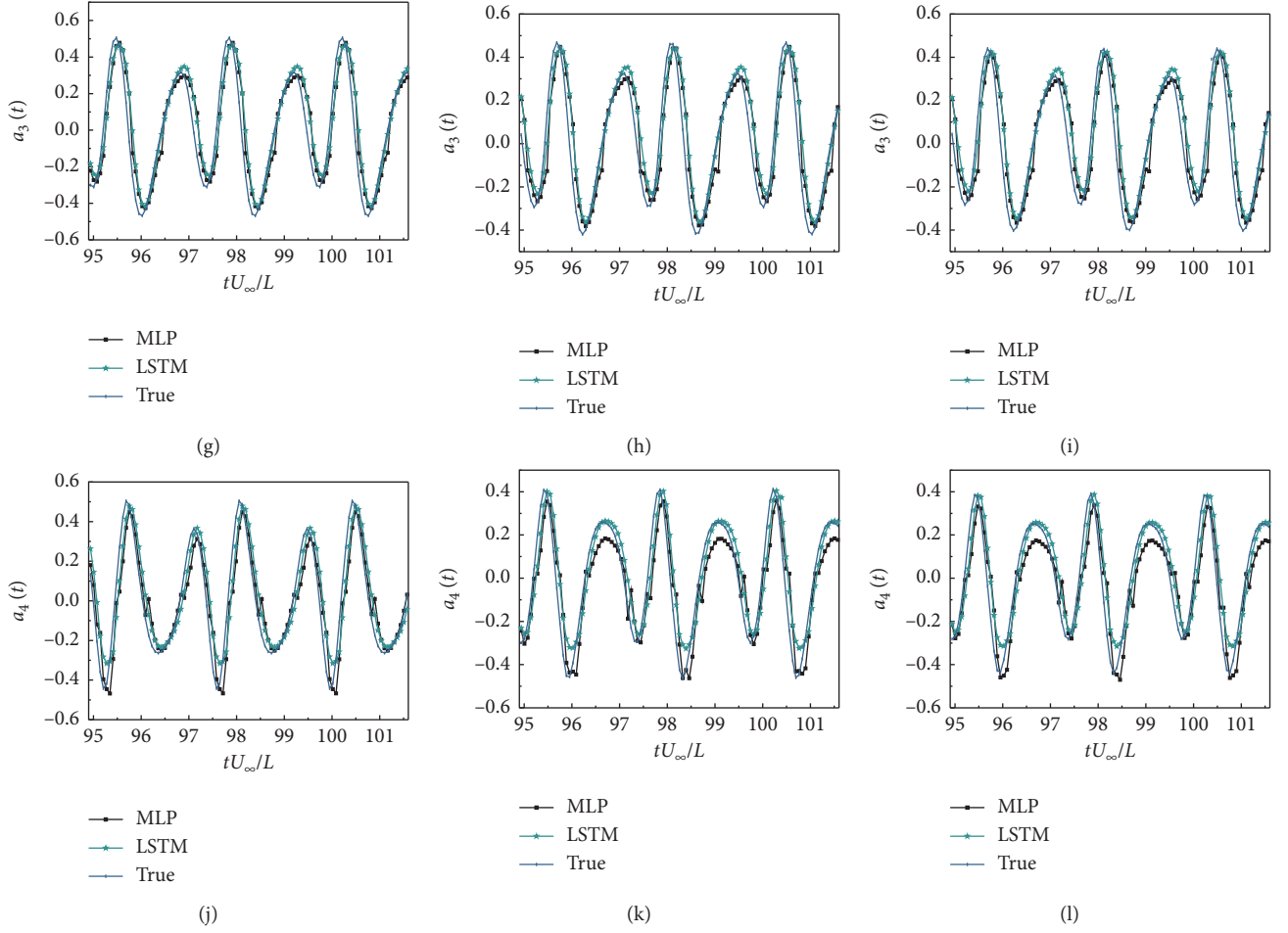


FIGURE 8: The first four POD modal coefficients predicted by MLP and LSTM at  $Ma = 0.53, 0.57$ , and  $0.6$ . (a)  $a_1(t)$  at  $Ma = 0.53$ . (b)  $a_1(t)$  at  $Ma = 0.57$ . (c)  $a_1(t)$  at  $Ma = 0.6$ . (d)  $a_2(t)$  at  $Ma = 0.53$ . (e)  $a_2(t)$  at  $Ma = 0.57$ . (f)  $a_2(t)$  at  $Ma = 0.6$ . (g)  $a_3(t)$  at  $Ma = 0.53$ . (h)  $a_3(t)$  at  $Ma = 0.57$ . (i)  $a_3(t)$  at  $Ma = 0.6$ . (j)  $a_4(t)$  at  $Ma = 0.53$ . (k)  $a_4(t)$  at  $Ma = 0.57$ . (l)  $a_4(t)$  at  $Ma = 0.6$ .

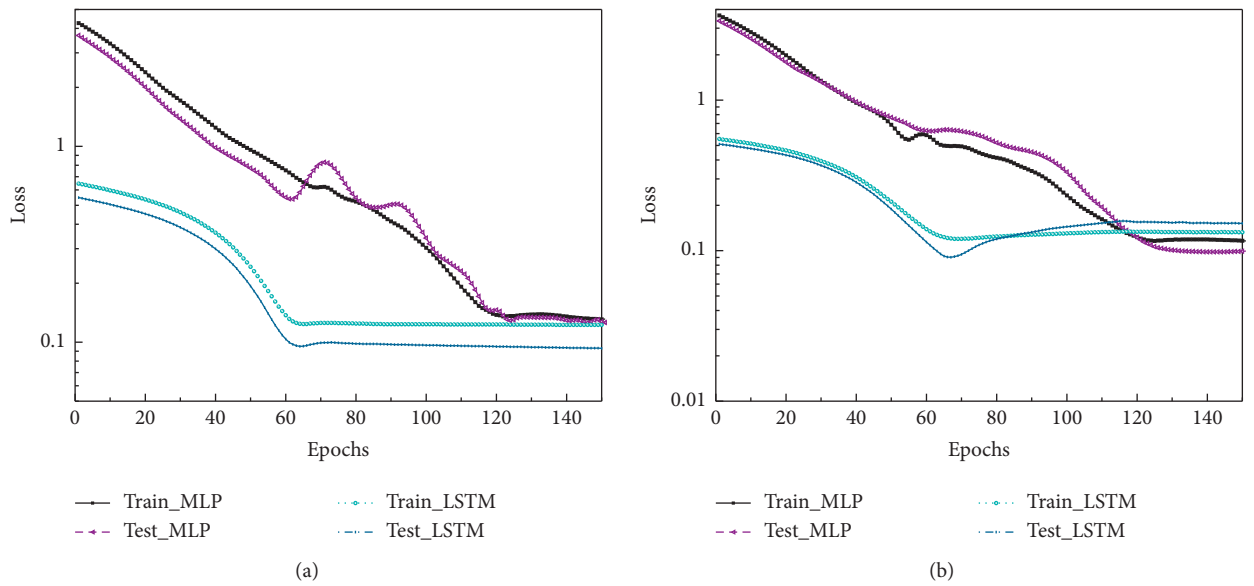


FIGURE 9: Continued.

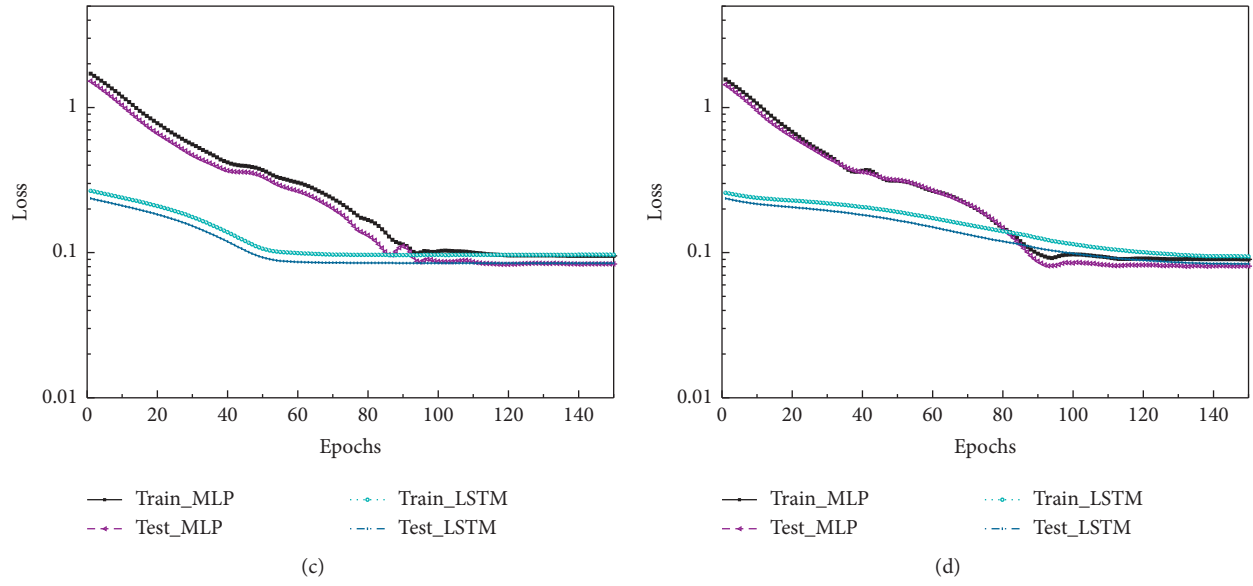


FIGURE 9: The training and test loss predicted by MLP and LSTM. (a)  $a_1(t)$ . (b)  $a_2(t)$ . (c)  $a_3(t)$ . (d)  $a_4(t)$ .

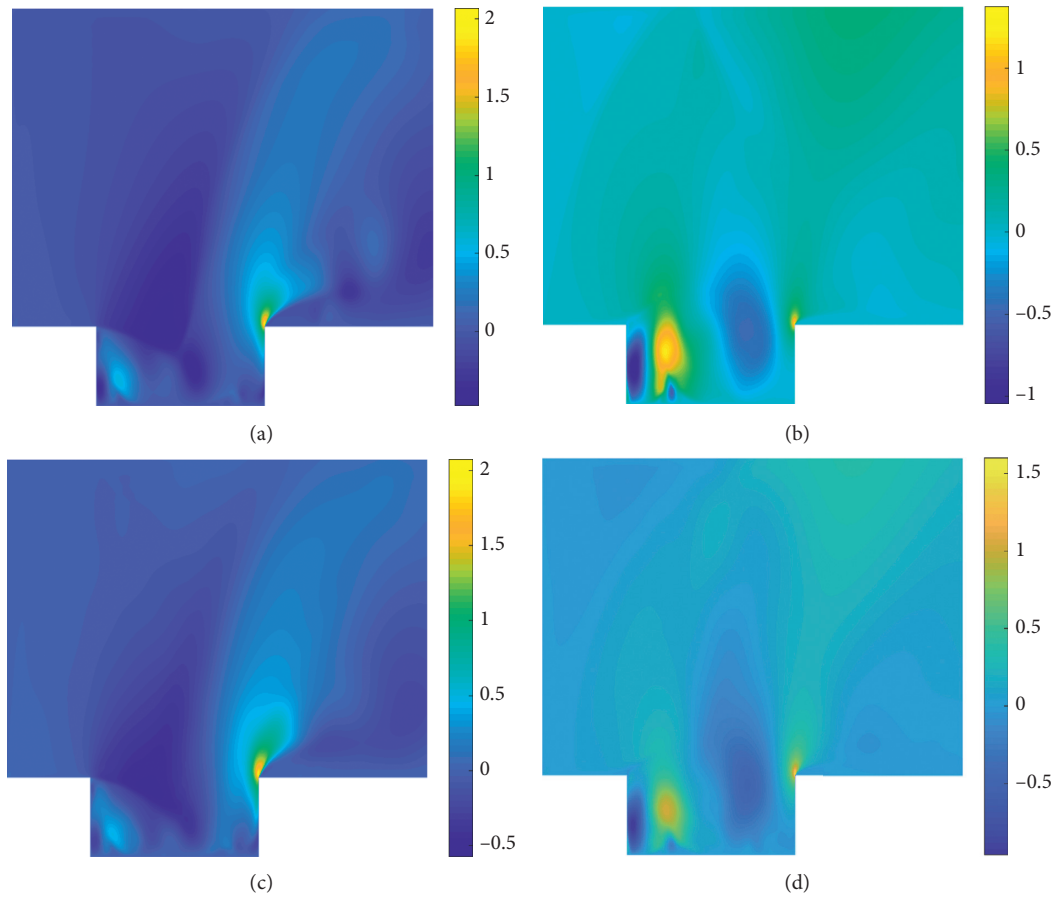


FIGURE 10: Continued.

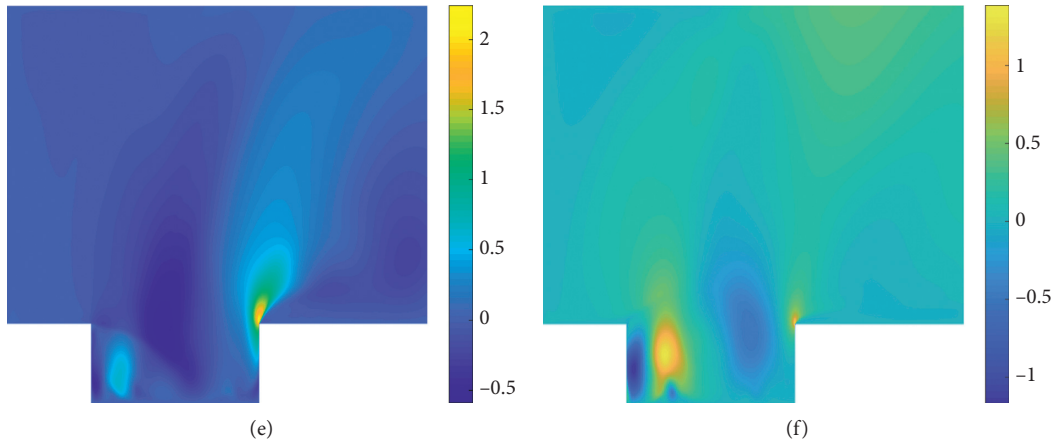


FIGURE 10: The reconstructed velocity field by DNS and deep learning frameworks at  $Ma = 0.54$ . (a) DNS ( $tU_\infty/L = 98.26$ ), (b) DNS ( $tU_\infty/L = 101.57$ ), (c) MLP ( $tU_\infty/L = 98.26$ ), (d) MLP ( $tU_\infty/L = 101.57$ ), (e) LSTM ( $tU_\infty/L = 98.26$ ), and (f) LSTM ( $tU_\infty/L = 101.57$ ).

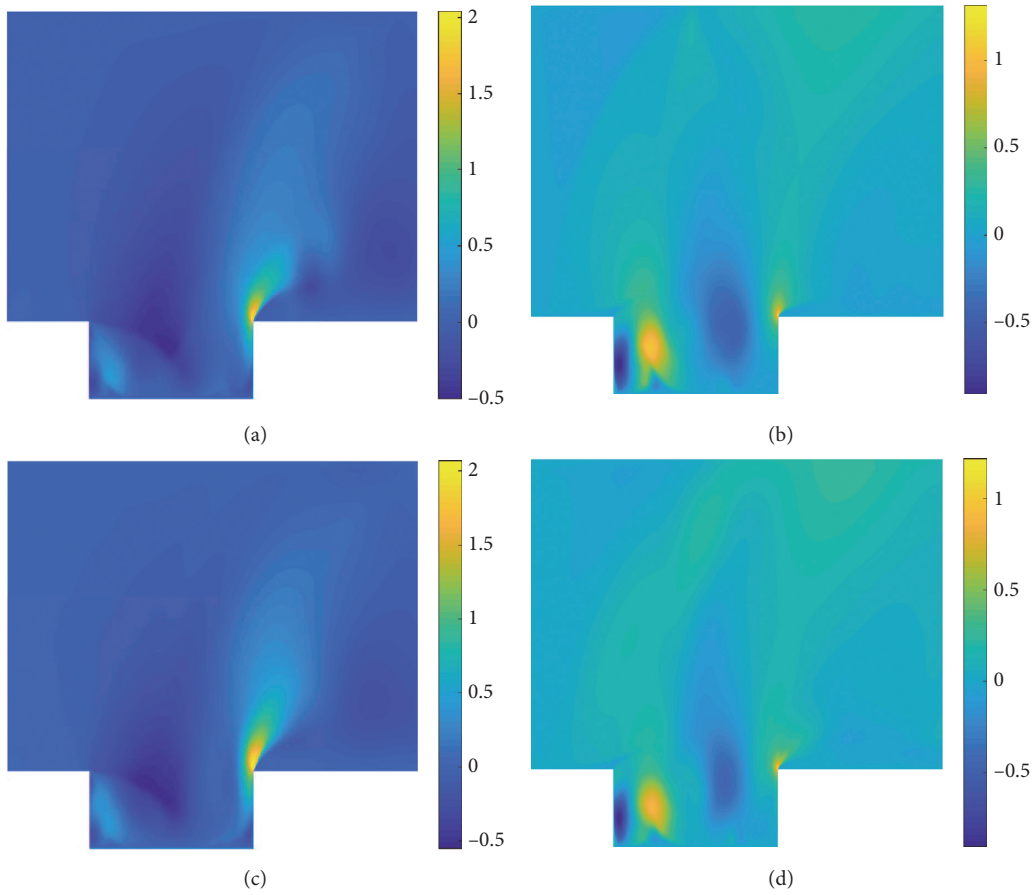


FIGURE 11: Continued.

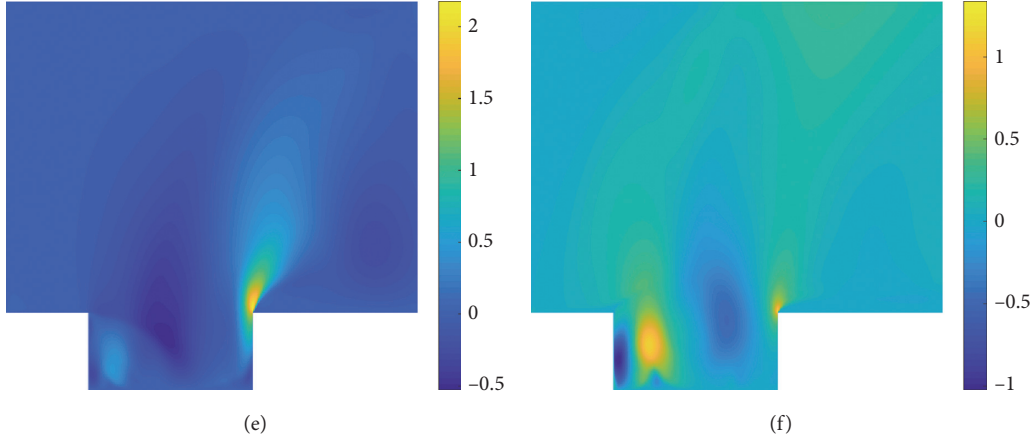


FIGURE 11: The reconstructed velocity field by DNS and deep learning frameworks at  $Ma = 0.6$ . (a) DNS ( $tU_\infty/L = 98.26$ ), (b) DNS ( $tU_\infty/L = 101.57$ ), (c) MLP ( $tU_\infty/L = 98.26$ ), (d) MLP ( $tU_\infty/L = 101.57$ ), (e) LSTM ( $tU_\infty/L = 98.26$ ), and (f) LSTM ( $tU_\infty/L = 101.57$ ).

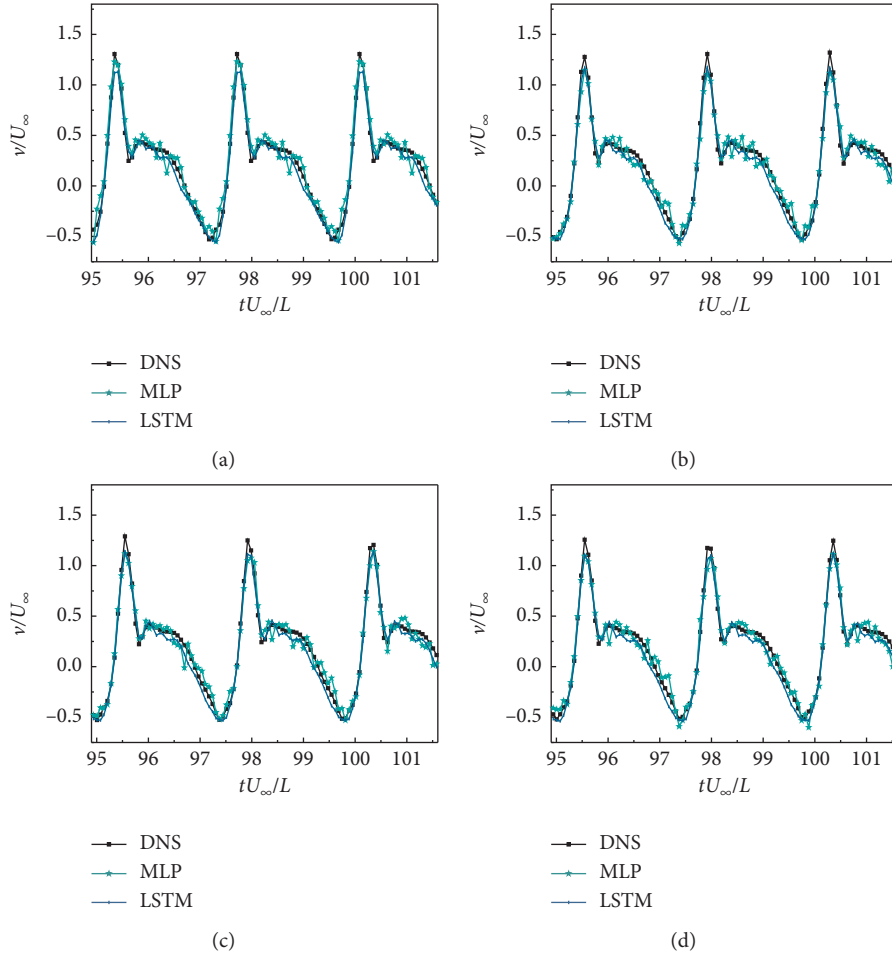


FIGURE 12: Continued.

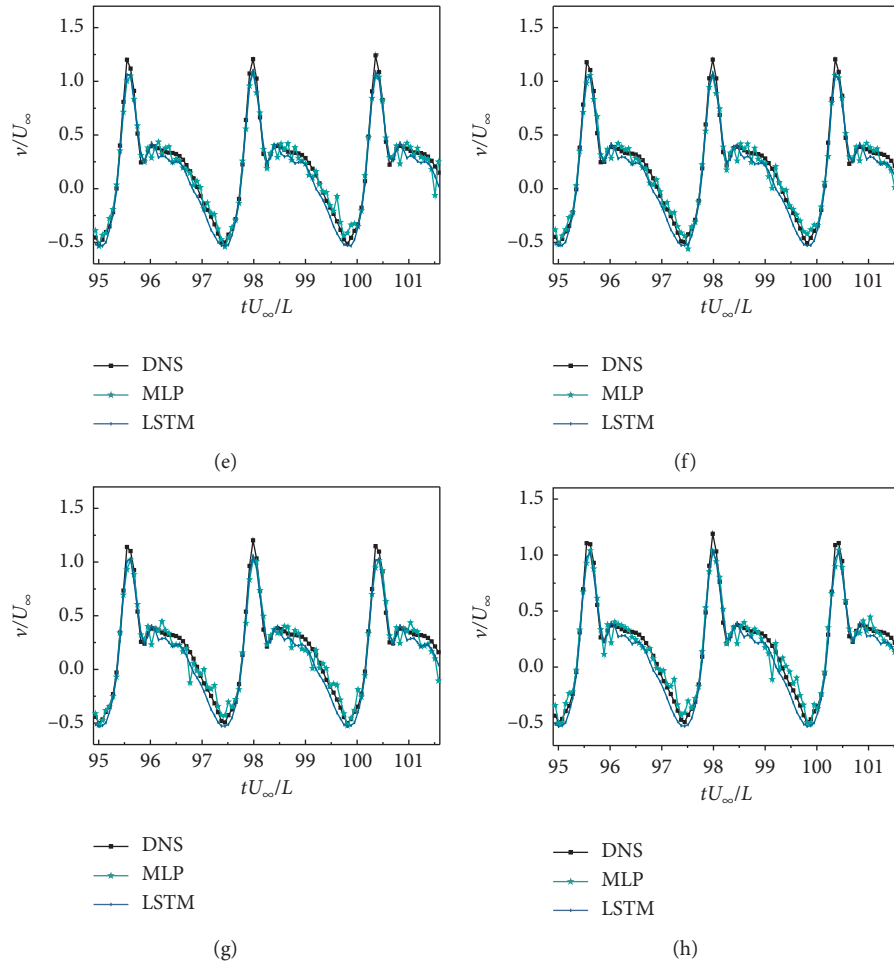


FIGURE 12: Comparison of the time trace of velocity. (a)  $Ma = 0.53$ , (b)  $Ma = 0.54$ , (c)  $Ma = 0.55$ , (d)  $Ma = 0.56$ , (e)  $Ma = 0.57$ , (f)  $Ma = 0.58$ , (g)  $Ma = 0.59$ , and (h)  $Ma = 0.6$ .

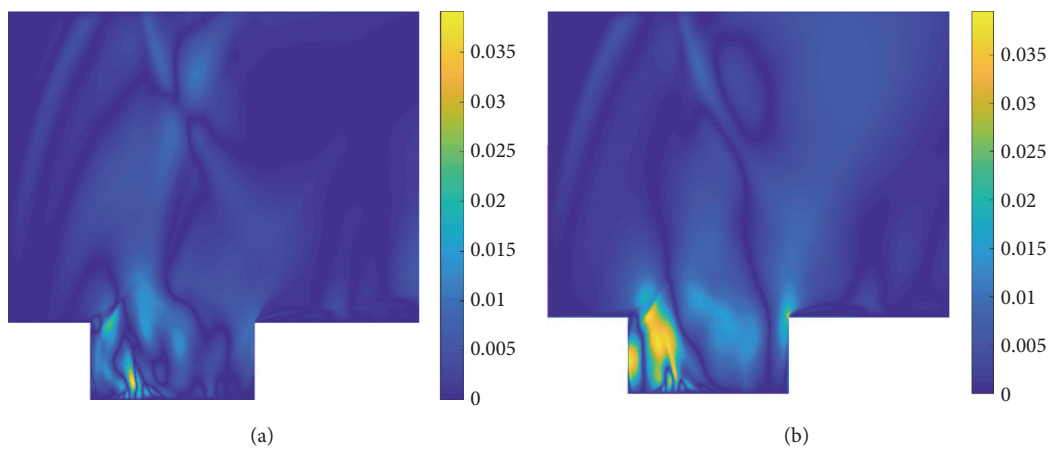


FIGURE 13: Continued.



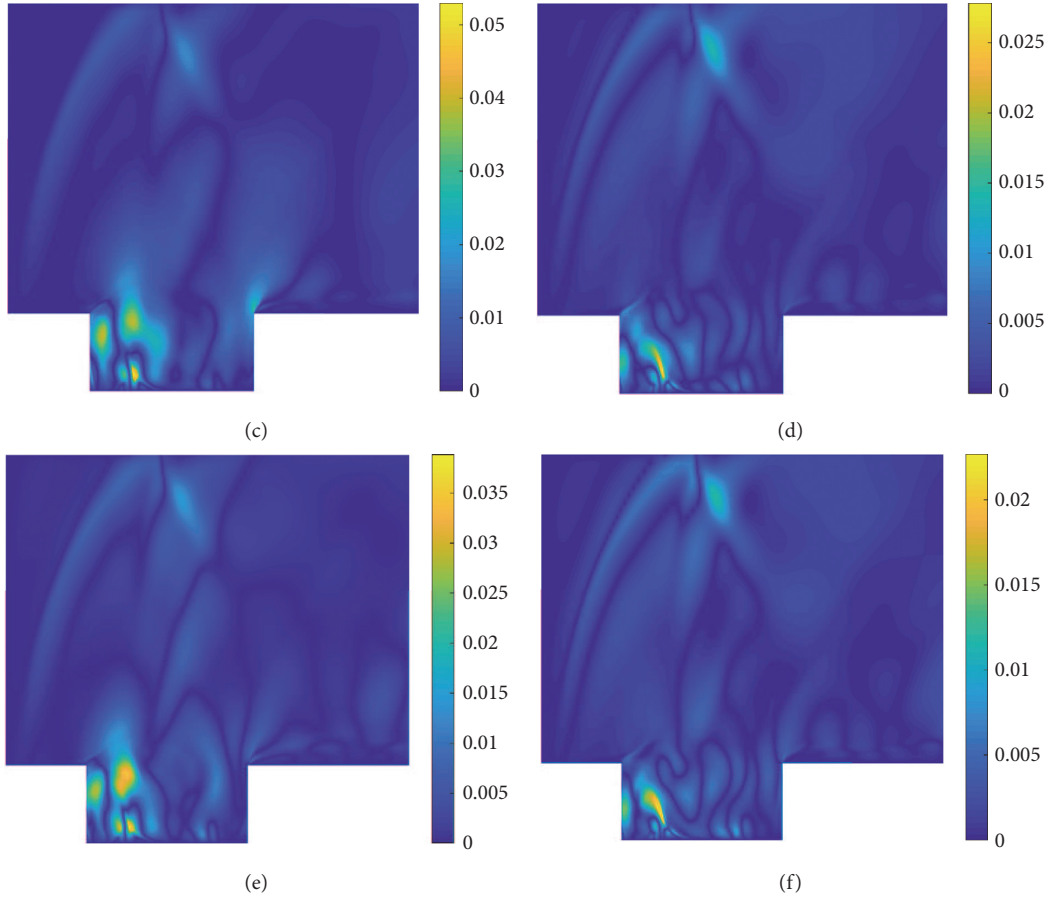


FIGURE 13: Comparison of RMSE predicted by two deep learning frameworks ( $Ma = 0.53, 0.54, 0.55$ ). (a) RMSE (MLP,  $Ma = 0.53$ ), (b) RMSE (LSTM,  $Ma = 0.53$ ), (c) RMSE (MLP,  $Ma = 0.54$ ), (d) RMSE (LSTM,  $Ma = 0.54$ ), (e) RMSE (MLP,  $Ma = 0.55$ ), and (f) RMSE (LSTM,  $Ma = 0.55$ ).

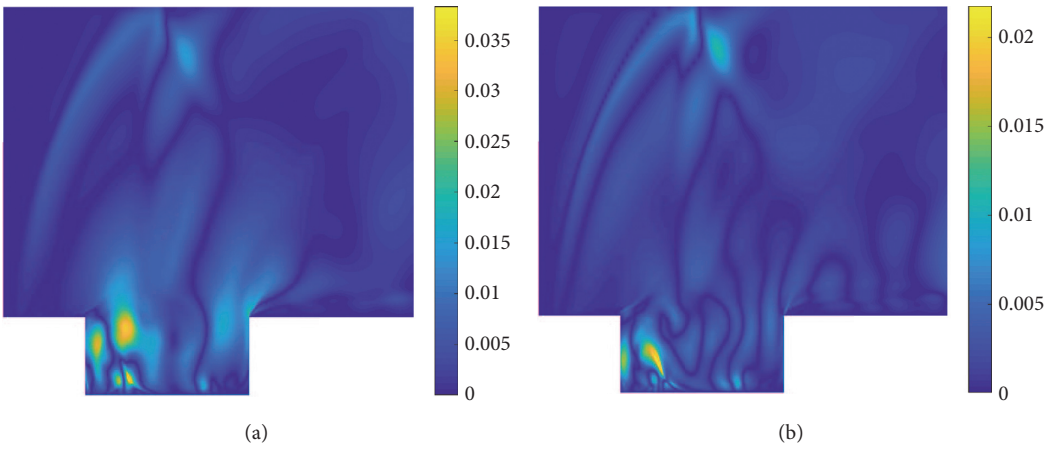


FIGURE 14: Continued.

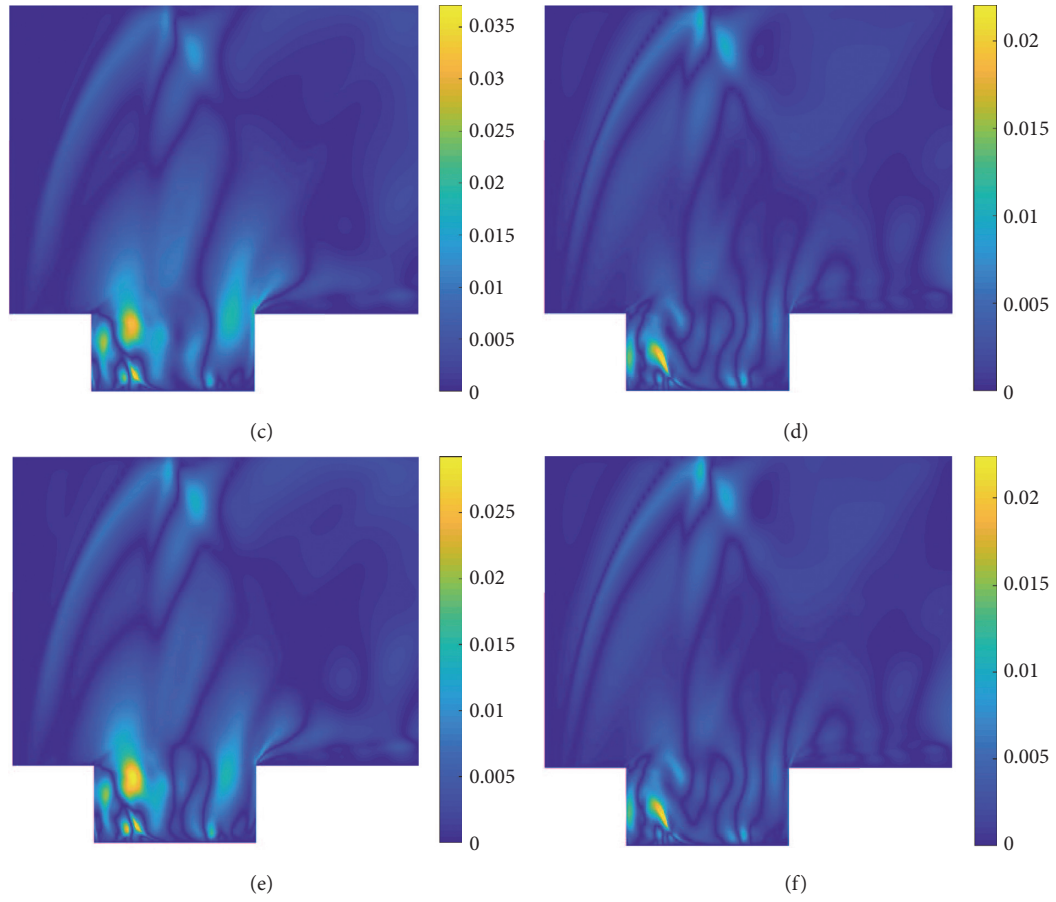


FIGURE 14: Comparison of RMSE predicted by two deep learning frameworks ( $Ma = 0.56, 0.57, 0.58$ ). (a) RMSE (MLP,  $Ma = 0.56$ ), (b) RMSE (LSTM,  $Ma = 0.56$ ), (c) RMSE (MLP,  $Ma = 0.57$ ), (d) RMSE (LSTM,  $Ma = 0.57$ ), (e) RMSE (MLP,  $Ma = 0.58$ ), and (f) RMSE (LSTM,  $Ma = 0.58$ ).

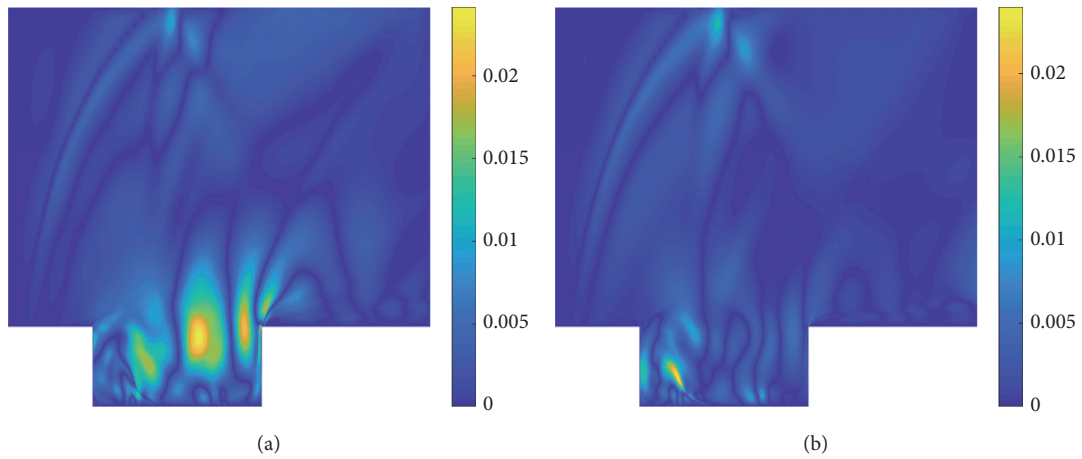


FIGURE 15: Continued.

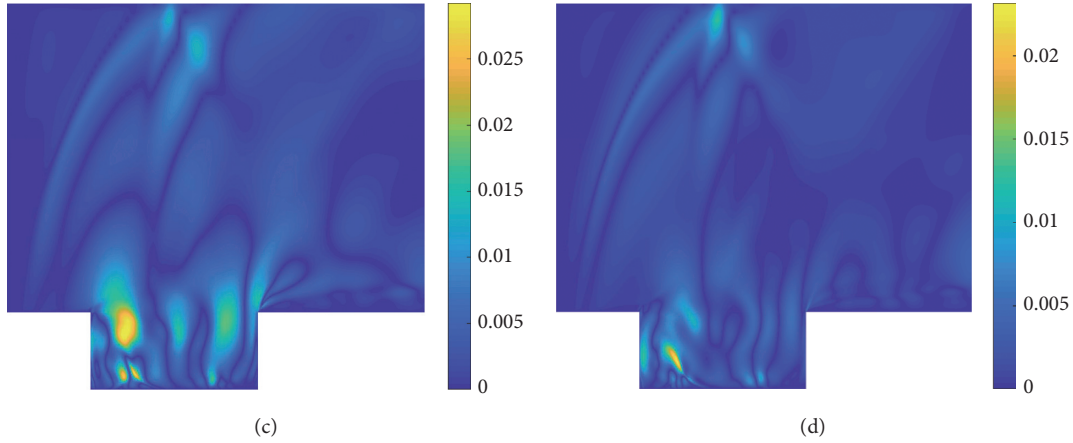


FIGURE 15: Comparison of RMSE predicted by two deep learning frameworks ( $Ma = 0.59, 0.6$ ). (a) RMSE (MLP,  $Ma = 0.59$ ), (b) RMSE (LSTM,  $Ma = 0.59$ ), (c) RMSE (MLP,  $Ma = 0.6$ ), and (d) RMSE (LSTM,  $Ma = 0.6$ ).

TABLE 4: Comparison of the calculation costs between DNS, MLP, and LSTM.

Method	POD time (s)	Training time (s)	Total time (s)	Time reduction (s)
DNS	0	0	2976	-
ROM (MLP)	176	572	748	2228
ROM (LSTM)	176	169	345	2631

the ROMs based on the MLP or LSTM framework are accurate and reliable.

In order to further analyze the reconstruction results in detail, Figure 12 compares the time trace of velocity at the monitor ( $x/L = 0.9, 0$ ). It can be seen from the figure that the shapes of time trace curves of the velocity across different Mach numbers are basically the same, but their magnitudes are different. The two deep learning frameworks can capture the speed value at all Mach numbers. The prediction results by the LSTM framework are more stable and will not oscillate.

**5.4.4. Comparison of MLP and LSTM Frameworks.** From the above analysis of reconstruction results, it can be seen that the MLP and LSTM frameworks can well predict the flow field in the future, but MLP has a certain degree of oscillation. In order to further compare the performance of the two frameworks, Figure 13–15 compare the root mean square error (RMSE) of the reconstruction results. Except at a Mach number of 0.53, the RMSE predicted by the LSTM framework is slightly larger than that by the MLP framework. At all other Mach numbers, the RMSE predicted by the LSTM framework is obviously lower than that by the MLP framework. Especially at Mach number 0.54, the maximum RMSE predicted by the MLP framework is 0.05, while the maximum RMSE predicted by the LSTM framework is 0.025. In addition, all the maximum RMSEs occur near the cavity leading edge, which may be related to the instability of speed caused by inflow passing through the cavity leading edge.

The calculation time of the ROM in this paper mainly includes the time of POD analysis and the training time of

deep learning frameworks. Table 4 compares the calculation time of these two ROMs with the DNS simulation. All the calculations were carried out on a computer with 44 cores (Intel Xeon E5-2699) and 256 GB RAM, and these calculations are based on a single core. As can be seen from the table, both of the two deep learning frameworks can significantly reduce the calculation time by order of magnitude on average. The LSTM framework has the least calculation time.

## 6. Conclusions

In this paper, a reduced-order model based on POD and deep learning was established for predicting cavity flow oscillations across multiple Mach numbers. The specific conclusions are concluded as follows:

- (i) After the POD analysis of the numerical simulation data, the first 13 POD modes were extracted, which occupy 99.9% of the energy. The POD modal structures at  $Ma = 0.51, 0.6$  are qualitatively similar, so the deep learning method can accurately learn their common features. In addition, their size and direction are different, and the variation between different Mach numbers is irregular and unpredictable. Therefore, it is of great significance to establish the reduced-order model of cavity flow oscillations across multi-Mach numbers.
- (ii) By comparing the predicted coefficients with the actual POD coefficients, it is found that the MLP and LSTM frameworks can accurately predict the POD coefficients, but there are some small oscillations in the coefficients predicted by the MLP framework.

- (iii) Both of the frameworks can accurately reconstruct the velocity field at different Mach numbers, but the results reconstructed by the LSTM framework are more accurate, and the root mean square error is smaller than the results reconstructed by the MLP framework. Comparing the total computation time of the two frameworks with DNS, it is found that the time computed by the ROM proposed in this paper is reduced by at least one order of magnitude and that by the LSTM framework is the least.

## Nomenclature

$D$ : Depth of cavity (m)  
 $L$ : Length of cavity (m)  
 $M$ : Number of POD modes  
 $Ma$ : Mach number  
 $N$ : Number of POD snapshots  
 $p$ : Pressure (Pa)  
 $Re$ : Reynolds number  
 $T$ : Temperature (K)  
 $t$ : Time (s)  
 $u, v$ :  $x, y$  Cartesian components of velocity fluctuation (m/s)  
 $U_\infty$ : Free stream velocity (m/s)  
 $W$ : Width of cavity (m).

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 51675425 and 52075441), Shaanxi Key Research Program Project (Grant no. 2020ZDLGY06-09), Science and Technology on Micro-System Laboratory Foundation (Grant no. 6142804200405), and Aeronautical Science Foundation of China (Grant no. 20200015053001).

## References

- [1] D. Bacci, A. J. Saddington, and D. Bray, "Identification of the formation of resonant tones in compressible cavity flows," *Aerospace Science and Technology*, vol. 77, pp. 320–331, 2018.
- [2] S. J. Lawson and G. N. Barakos, "Review of numerical simulations for high-speed, turbulent cavity flows," *Progress in Aerospace Sciences*, vol. 47, no. 3, pp. 186–216, 2011.
- [3] I. Hagiwara, D. W. Wang, Q. Z. Shi, and R. S. Rao, "Reduction of noise inside a cavity by piezoelectric actuators," *Journal of Vibration and Acoustics*, vol. 125, no. 1, pp. 12–17, 2003.
- [4] F. Ning, S. Ning, C. Zhang, and Z. Liu, "Numerical simulation study on aeroacoustic characteristics within deformable cavities," *Shock and Vibration*, vol. 2019, no. 3, 8 pages, Article ID 9573786, 2019.
- [5] S. V. Babu, G. J. M. Loupy, F. Dehaeze, G. N. Barakos, and N. J. Taylor, "Aeroelastic simulations of stores in weapon bays using detached-eddy simulation," *Journal of Fluids and Structures*, vol. 66, pp. 207–228, 2016.
- [6] G. J. M. Loupy, G. N. Barakos, and N. J. Taylor, "Cavity flow over a transonic weapons bay during door operation," *Journal of Aircraft*, vol. 55, no. 1, pp. 339–354, 2018.
- [7] T. R. Ricciardi, W. R. Wolf, and R. Speth, "Acoustic prediction of lagoon landing gear: cavity noise and coherent structures," *AIAA Journal*, vol. 56, no. 11, pp. 4379–4399, 2018.
- [8] E. Neri, J. Kennedy, and G. J. Bennett, "Bay cavity noise for full-scale nose landing gear: a comparison between experimental and numerical results," *Aerospace Science and Technology*, vol. 72, pp. 278–291, 2018.
- [9] K. Luo, W. Zhu, Z. Xiao et al., "Investigation of spectral characteristics by passive control methods past a supersonic cavity," *AIAA Journal*, vol. 56, no. 7, pp. 2669–2686, 2018.
- [10] X. Wang, D. Yang, J. Liu, and F. Zhou, "Control of pressure oscillations induced by supersonic cavity flow," *AIAA Journal*, vol. 58, no. 5, pp. 2070–2077, 2020.
- [11] F. Pish, T. D. Manh, M. B. Gerdroodbary, N. D. Nam, R. Moradi, and H. Babazadeh, "Computational study of the cavity flow over sharp nose cone in supersonic flow," *International Journal of Modern Physics C*, vol. 31, no. 6, Article ID 2050079, 2020.
- [12] C. W. Rowley, T. Colonius, and A. J. Basu, "On self-sustained oscillations in two-dimensional compressible flow over rectangular cavities," *Journal of Fluid Mechanics*, vol. 455, no. 455, pp. 315–346, 2002.
- [13] C. Zhang, Z. Wan, and D. Sun, "Mode transition and oscillation suppression in supersonic cavity flow," *Applied Mathematics and Mechanics*, vol. 37, no. 7, pp. 941–956, 2016.
- [14] Y. Sun, K. Taira, L. N. Cattafesta, and L. S. Ukeiley, "Biglobal instabilities of compressible open-cavity flows," *Journal of Fluid Mechanics*, vol. 826, pp. 270–301, 2017.
- [15] F. Noh-Pat, J. Xamán, G. Álvarez et al., "Unsteady-rans simulation of conjugate heat transfer in a cavity with a vertical semitransparent wall," *Computers & Fluids*, vol. 117, pp. 183–195, 2015.
- [16] R. K. Soni, N. Arya, and A. De, "Numerical simulation of supersonic separating-reattaching flow through rans," *Journal of Physics: Conference Series*, vol. 822, Article ID 012037, 2017.
- [17] G. Rubio, W. D. Roeck, M. Baelmans, and W. Desmet, "Numerical identification of flow-induced oscillation modes in rectangular cavities using large eddy simulation," *International Journal for Numerical Methods in Fluids*, vol. 53, no. 5, pp. 851–866, 2010.
- [18] S. A. Tkachenko, G. E. Lau, V. Timchenko, G. H. Yeoh, and J. Reizes, "Effect of heat loss on turbulent buoyancy-driven flow in a rectangular cavity using the large-eddy simulation," *Numerical Heat Transfer, Part A: Applications*, vol. 70, no. 7, pp. 689–706, 2016.
- [19] S. Feng, X. Zheng, R. Hu, and P. Wang, "Large eddy simulation of high-Reynolds-number atmospheric boundary layer flow with improved near-wall correction," *Applied Mathematics and Mechanics*, vol. 41, no. 1, pp. 33–50, 2020.
- [20] D. J. Lucia, P. S. Beran, and W. A. Silva, "Reduced-order modeling: new approaches for computational physics," *Progress in Aerospace Sciences*, vol. 40, no. 1, pp. 51–117, 2004.
- [21] M. Ghoreyshi, A. Jirásek, and R. M. Cummings, "Reduced order unsteady aerodynamic modeling for stability and control analysis using computational fluid dynamics," *Progress in Aerospace Sciences*, vol. 71, pp. 167–217, 2014.



- [22] X. Gloerfelt, "Compressible pod/galerkin reduced-order model of self-sustained oscillations in a cavity," in *Proceedings of the 12th AIAA/CEAS Aeroacoustics Conference (27th AIAA Aeroacoustics Conference)*, Cambridge, MA, USA, May 2006.
- [23] J. Yu and J. S. Hesthaven, "Flowfield reconstruction method using artificial neural network," *AIAA Journal*, vol. 57, no. 2, pp. 482–498, 2018.
- [24] A. T. Mohan and D. V. Gaitonde, "A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks," 2018, <https://arxiv.org/abs/1804.09269>.
- [25] O. San, R. Maulik, and M. Ahmed, "An artificial neural network framework for reduced order modeling of transient flows," *Communications in Nonlinear Science and Numerical Simulation*, vol. 77, pp. 271–287, 2019.
- [26] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [27] M. Lee, J. Lee, and J.-H. Chang, "Ensemble of jointly trained deep neural network-based acoustic models for reverberant speech recognition," *Digital Signal Processing*, vol. 85, pp. 1–9, 2019.
- [28] Z. Toktam, M. H. Mohammad, and D. Mahmood, "Adaptive windows multiple deep residual networks for speech recognition," *Expert Systems with Applications*, vol. 139, Article ID 112840, 2019.
- [29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, Lake Tahoe, NV, USA, December 2012.
- [30] Y. Lyu, J. Chen, and Z. Song, "Image-based process monitoring using deep learning framework," *Chemometrics and Intelligent Laboratory Systems*, vol. 189, pp. 8–17, 2019.
- [31] J. Yang, T. Gao, S. Jiang, S. Li, and Q. Tang, "Fault diagnosis of rotating machinery based on one-dimensional deep residual shrinkage network with a wide convolution layer," *Shock and Vibration*, vol. 2020, no. 4, 12 pages, Article ID 8880960, 2020.
- [32] J. Zhou, X. Yang, L. Zhang, S. Shao, and G. Bian, "Multisignal vgg19 network with transposed convolution for rotating machinery fault diagnosis based on deep transfer learning," *Shock and Vibration*, vol. 2020, Article ID 8863388, 12 pages, 2020.
- [33] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics*, vol. 807, pp. 155–166, 2016.
- [34] J. N. Kutz, "Deep learning in fluid dynamics," *Journal of Fluid Mechanics*, vol. 814, pp. 1–4, 2017.
- [35] T. P. Miyanawala and R. K. Jaiman, "An efficient deep learning technique for the Navier-Stokes equations: application to unsteady wake flow dynamics," 2017, <https://arxiv.org/abs/1710.09099>.
- [36] S. Lee and D. You, "Prediction of laminar vortex shedding over a cylinder using deep learning," 2017, <https://arxiv.org/abs/1712.07854>.
- [37] Z. Sun, C. Wang, Y. Zheng et al., "Non-intrusive reduced-order model for predicting transonic flow with varying geometries," *Chinese Journal of Aeronautics*, vol. 33, no. 2, pp. 508–519, 2020.
- [38] S. A. Renganathan, R. Maulik, and V. Rao, "Machine learning for nonintrusive model order reduction of the parametric inviscid transonic flow past an airfoil," *Physics of Fluids*, vol. 32, no. 4, Article ID 047110, 2020.
- [39] P. Wu, J. Sun, X. Chang et al., "Data-driven reduced order model with temporal convolutional neural network," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, Article ID 112766, 2020.
- [40] L. Sirovich and M. Kirby, "Turbulence and the dynamics of coherent structures. I. Coherent structures," *Quarterly of Applied Mathematics*, vol. 45, no. 3, pp. 561–571, 1987.
- [41] D. Rehman, I. Akhtar, and M. Naqvi, "Proper orthogonal decomposition of the pressure field in bluff bodies," in *Proceedings of the International Bhurban Conference on Applied Sciences and Technology*, pp. 196–202, Islamabad, Pakistan, January 2013.
- [42] C. W. Rowley, *Modeling, simulation, and control of cavity flow oscillations*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, USA, 2002.
- [43] C. Zhang, Z. Wan, and D. Sun, "Model reduction for supersonic cavity flow using proper orthogonal decomposition (pod) and galerkin projection," *Applied Mathematics and Mechanics*, vol. 38, no. 5, pp. 723–736, 2017.
- [44] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [45] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, no. 14–15, pp. 2627–2636, 1998.
- [46] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," *Journal of Clinical Epidemiology*, vol. 49, no. 11, pp. 1225–1231, 1996.
- [47] H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou, and M. Ettaouil, "Multilayer perceptron: architecture optimization and training," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. 1, pp. 26–30, 2016.
- [48] G. Panchal, A. Ganatra, Y. P. Kosta, and D. Panchal, "Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers," *International Journal of Computer Theory and Engineering*, vol. 3, no. 2, pp. 332–337, 2011.
- [49] Y. Ghanou and G. Bencheikh, "Architecture optimization and training for the multilayer perceptron using ant system," *International Journal of Computer Science*, vol. 43, no. 1, pp. 1–7, 2016.
- [50] A. Jiang and W. Wang, "Research on optimization of relu activation function," *Transducer and Microsystem Technologies*, vol. 37, no. 2, pp. 50–52, 2018.
- [51] F. Ertam and G. Aydın, "Data classification with deep learning using tensorflow," in *Proceedings of the 2017 international conference on computer science and engineering (UBMK)*, pp. 755–758, IEEE, Antalya, Turkey, October 2017.
- [52] D. Pedamonti, "Comparison of non-linear activation functions for deep neural networks on mnist classification task," 2018, <https://arxiv.org/abs/1804.02763>.
- [53] D. P. Kingma, J. Ba, and Adam, "A method for stochastic optimization," in *Proceedings of the International Conference for Learning Representations*, San Diego, CA, USA, January 2015.
- [54] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, <https://arxiv.org/abs/1609.04747>.
- [55] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the International Conference on Machine Learning*, Atlanta, GA, USA, June 2013.



- [56] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [57] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks : The Official Journal of the International Neural Network Society*, vol. 61, pp. 85–117, 2015.
- [58] Z. Liu, F. Ning, Q. Zhai, H. Ding, J. Wei, and B. Li, "Study on oscillation phenomena in morphing cavities with ramps of leading wall," *Journal of Aircraft*, vol. 58, no. 3, pp. 436–447, 2021.
- [59] H. H. Heller, D. G. Holmes, and E. E. Covert, "Flow-induced pressure oscillations in shallow cavities," *Journal of Sound and Vibration*, vol. 18, no. 4, pp. 545–553, 1971.
- [60] M. A. Kegerise, *An experimental investigation of flow-induced cavity oscillations*, Ph.D. thesis, Syracuse University, Syracuse, NY, USA, 1999.