aerospace

MDPI

Article

Reduced-Order Modeling of Steady and Unsteady Flows with Deep Neural Networks †

Bryan Barraza * and Andreas Gross

Mechanical & Aerospace Engineering Department, New Mexico State University, Las Cruces, NM 88003, USA; agross@nmsu.edu

* Correspondence: bryanbar@nmsu.edu

† This paper is an extended version of our Paper published in Barraza, B.; Wei, Z.; Gross, A. Reduced-Order Modeling of Steady and Unsteady Flows with Deep Neural Networks. In proceedings of the AIAA Aviation 2022 Forum, 2022-3978, Chicago, IL, USA, 27 June–1 July 2022.

Abstract: Large-eddy and direct numerical simulations generate vast data sets that are challenging to interpret, even for simple geometries at low Reynolds numbers. This has increased the importance of automatic methods for extracting significant features to understand physical phenomena. Traditional techniques like the proper orthogonal decomposition (POD) have been widely used for this purpose. However, recent advancements in computational power have allowed for the development of data-driven modal reduction approaches. This paper discusses four applications of deep neural networks for aerodynamic applications, including a convolutional neural network autoencoder, to analyze unsteady flow fields around a circular cylinder at Re = 100 and a supersonic boundary layer with Tollmien–Schlichting waves. The autoencoder results are comparable to those obtained with POD and spectral POD. Additionally, it is demonstrated that the autoencoder can compress steady hypersonic boundary-layer profiles into a low-dimensional vector space that is spanned by the pressure gradient and wall-temperature ratio. This paper also proposes a convolutional neural network model to estimate velocity and temperature profiles across different hypersonic flow conditions.

Keywords: reduced-order modeling; machine learning; deep neural networks; data-driven analysis; autoencoder; cylinder wake; hypersonic boundary layers

Citation: Barraza, B.; Gross, A. Reduced-Order Modeling of Steady and Unsteady Flows with Deep Neural Networks. *Aerospace* **2024**, *11*, 506. https://doi.org/10.3390/aerospace11070506

Academic Editor: Pietro Catalano

Received: 22 May 2024
Revised: 18 June 2024
Accepted: 20 June 2024
Published: 24 June 2024

1. Introduction

The understanding of unsteady flows through dimensionality reduction is crucial for gaining physical understanding and often accomplished by modal-decomposition techniques. Such techniques identify effective low-dimensional modal bases, typically orthogonal, that capture dominant flow mechanisms like vortex shedding with high accuracy [1]. The proper orthogonal decomposition (POD) [2] and the dynamic mode decomposition (DMD) [3] are prominent among these techniques, leveraging singular value decomposition (SVD) without necessitating knowledge of the governing dynamics. For instance, the "snapshot" method [4] requires only flow field data at regularly spaced time intervals. POD is particularly noted for its energy optimality, capturing the maximum unsteady energy content with the fewest modes, enhancing its attractiveness despite challenges in highly nonlinear scenarios [5].

Well-established modal decomposition techniques like POD sometimes fail to capture instability modes crucial in flows with instability waves that lead to large-scale fluid motion or breakdown [6]. To address these limitations, some researchers have combined POD modes with linear stability theory (LST) modes, while others have turned to the spectral proper orthogonal decomposition (SPOD). SPOD is adept at identifying coherent structures in data with multiple frequencies and time-dependent structures, making it suitable for scenarios where POD falls short [7]. This method has been effectively applied

to various flows, such as the wake behind curved cylinders [8] and dynamic stall of helicopter blades [9], where it permitted the identification of the dynamics of the different fluid structures.

Moreover, recent advances in machine learning (ML) and deep learning, particularly in autoencoders (AEs) and convolutional neural networks (CNNs), have shown promise in managing these nonlinear challenges [5,10–12]. Autoencoders, especially multi-layer and mode-decomposing CNN AEs, offer a more compact and accurate data representation by leveraging nonlinear activation functions, which often outperform traditional methods like POD in terms of data compression and mode representation [5,13].

In this paper, we employ three different modal decomposition techniques, POD, SPOD and convolutional neural networks to analyze three different cases. First, the unsteady two-dimensional wake flow downstream of a circular cylinder at a Reynolds number of 100 is analyzed, where we compare the coherent structures identified by POD and SPOD with those from a convolutional autoencoder, based on a convolutional CNN ideas by Murata et al. [5], aiming to highlight the respective strengths and applicabilities of each method in capturing essential flow dynamics. The AE architecture is then modified for the decomposition of a supersonic boundary-layer with Tollmien–Schlichting waves. The AE modes obtained are compared with modes from POD analyses. Finally, the study extends to examining steady hypersonic boundary-layer profiles using AEs, relating the latent space to the pressure gradient and wall-temperature ratio, and proposes a CNN for estimating the velocity and temperature profiles.

2. Case Description and Methodology

2.1. Cylinder Wake

A two-dimensional simulation of the flow around a circular cylinder for a Reynolds number based on cylinder diameter of $Re_D = 100$ was carried out with an in-house developed finite-volume code [14]. Length scales were made dimensionless with the cylinder diameter, and velocities were made dimensionless with the freestream velocity. Time was made dimensionless by the ratio of cylinder diameter to freestream velocity. The computational grid had 194 cells in the azimuthal direction and 130 cells in the radial direction (Figure 1), for a total of 25,220 cells. No-slip and no-penetration conditions were employed on the cylinder wall. The wall was assumed to be adiabatic, and non-reflecting boundary conditions were employed at the freestream boundary.

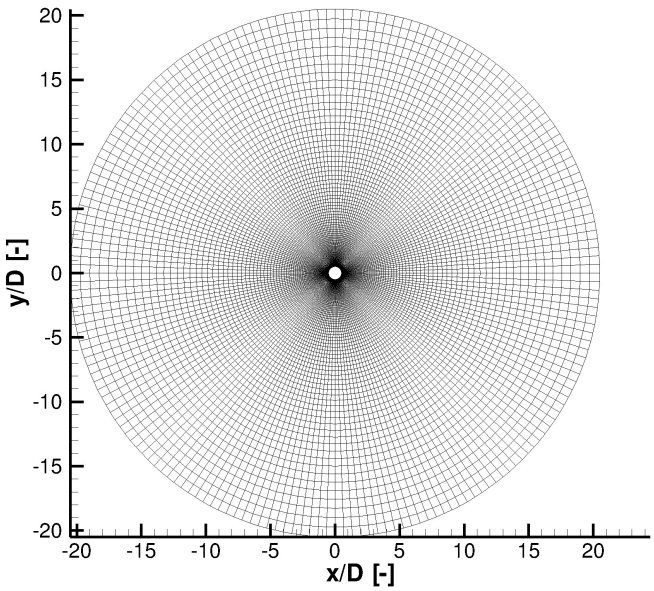

Figure 1. Computational grid for circular cylinder flow. Length scales were made dimensionless with the cylinder diameter, D.

2.1.1. Training Data

A total of 2560 snapshots of the flow field, saved at a constant time interval of $\Delta t = 0.25$, were used to train a CNN-AE. The data set covers roughly 108 periods of the primary shedding mode. The u and v velocity components for the entire computational domain were used as input and target data for the ML approach.

2.1.2. Model Architecture

A network architecture similar to that proposed by Murata et al. [5] was adopted. A summary of the network architecture is provided in Table 1. The architecture is designed to allow for four decomposed fields (i.e., modes). Due to its ease of use and high level of abstraction, all network architectures investigated in this paper were developed using the Keras [15] library with Tensorflow as back-end. The loss function was based on the mean-square error (MSE) and minimized with the Adam [16] optimizer. Early stopping was used with a patience of 20. After 796 epochs, the training and validation MSE were 6.23×10^{-5} and 6.66×10^{-5}, respectively.

Table 1. MD-CNN-AE network architecture for circular cylinder flow.

Encoder	
Layer	Output Shape
Input	(192, 128, 2)
1st Conv(3, 3, 16)	(192, 128, 16)
1st MaxPooling	(96, 64, 16)
2nd Conv(3, 3, 16)	(96, 64, 16)
2nd MaxPooling	(48, 32, 16)
3rd Conv(3, 3, 8)	(48, 32, 8)
3rd MaxPooling	(24, 16, 8)
4th Conv(3, 3, 8)	(24, 16, 8)
4th MaxPooling	(12, 8, 8)
5th Conv(3, 3, 4)	(12, 8, 4)
5th MaxPooling	(6, 4, 4)
6th Conv(3, 3, 2)	(6, 4, 2)
6th MaxPooling	(3, 2, 2)
Flatten	(12, 1)
FCNN (Latent Vector)	(2, 1)
Decoder	
Layer	Output Shape
FCNN output	(12, 1)
Reshape	(3, 2, 2)
1st upsampling	(6, 4, 2)
7th Conv(3, 3, 4)	(6, 4, 4)
2nd upsampling	(12, 8, 4)
8th Conv(3, 3, 8)	(12, 8, 8)
3rd upsampling	(24, 16, 8)
9th Conv(3, 3, 8)	(24, 16, 8)
4th upsampling	(48, 32, 8)
10th Conv(3, 3, 16)	(48, 32, 16)
5th upsampling	(96, 64, 16)
11th Conv(3, 3, 16)	(96, 64, 16)
6th upsampling	(192, 128, 16)
12th Conv(3, 3, 2)	(192, 128, 2)

2.2. Proper Orthogonal Decomposition

The proper orthogonal decomposition (POD),

$$\mathbf{v}(\mathbf{x}, t) = \sum_i a_i(t)\mathbf{q}_i(\mathbf{x}), \tag{1}$$

decomposes an unsteady flow into a series of time-coefficients, a_i, and modes, $\mathbf{q}_i(\mathbf{x})$. For the present analyses, a compressible POD kernel by Rowley et al. [17] was employed,

$$u^2 + v^2 + \frac{2\alpha}{\gamma - 1}c^2, \tag{2}$$

where c is the speed of sound, and α was set to $1/\gamma$ such that the kernel is identical to two times the total internal energy. The POD modes are orthogonal and here sorted according to their eigenvalue magnitudes. The eigenvalue magnitudes correspond to twice the total internal energy contained in the respective modes. For the cylinder wake case, POD analysis was carried out using 512 snapshots from the training database.

2.3. Spectral Proper Orthogonal Decomposition

SPOD is an algorithm that enables the separation of flow phenomena according to their frequencies and energies. Here, a brief introduction to the algorithm is presented. For a deeper review, see the works by Towne et al. [18] and Schmidt and Colonius [7]. An ensemble of M snapshots spaced at a constant time interval of Δt is divided into N_b data blocks with N_f snapshots each and a number of snapshots by which the blocks overlap, N_o. For the nth data block,

$$\mathbf{Q}^{(n)} = \begin{bmatrix} q_1^{(n)} \\ q_2^{(n)} \\ \vdots \\ q_{N_F}^{(n)} \end{bmatrix}, \tag{3}$$

the data are transformed from time space into frequency space with fast Fourier transforms (FFTs). The number of blocks, N_b is a power of 2 due the nature of the FFT algorithm. For each frequency, a matrix of Fourier realizations is assembled,

$$\hat{\mathbf{Q}}_k = \sqrt{\kappa}\begin{bmatrix} \hat{q}_k^{(1)} \\ \hat{q}_k^{(2)} \\ \vdots \\ \hat{q}_k^{(N_b)} \end{bmatrix}, \tag{4}$$

where k is the Fourier mode wavenumber, $\kappa = \Delta t/(sN_b)$ is a normalization factor with $s = 1$ if no data windowing is used (this option was chosen for the results presented in this paper). Using the POD formalism, an orthogonal basis to best approximate the $\hat{q}_k^{(n)}$ is found,

$$\hat{q}_k^{(n)} = \sum_{j=1}^{m} a_{kj}^{(n)}\Phi_{kj} \tag{5}$$

where Φ_{kj} are the orthogonal basis functions for the kth wavenumber, and $a_{kj}^{(n)}$ are the corresponding coefficients of the nth block. A spectral correlation matrix is built,

$$\mathbf{M}_k = \hat{\mathbf{Q}}_k^* \mathbf{W}\hat{\mathbf{Q}}_k, \tag{6}$$

where $\mathbf{W}$ is a weight matrix. The $\mathbf{M}_k$ matrix is Hermitian. The singular value decomposition for $\hat{\mathbf{Q}}_k$ is related to the eigenvalue problem for $\mathbf{M}_k$,

$$\mathbf{M}_k = \hat{\mathbf{Q}}_k^* \mathbf{W} \hat{\mathbf{Q}}_k = \mathbf{\Theta}_k \mathbf{\Lambda}_k \mathbf{\Theta}_k^* . \tag{7}$$

Here, $\mathbf{\Theta}_k$ is the left eigenvector matrix, and $\mathbf{\Lambda}_k$ is a diagonal matrix that contains the eigenvalues. For the cylinder wake, 512 snapshots from the training data base were split into $N_b = 3$ blocks of $N_f = 256$ snapshots, with an overlap of $N_o = 128$ snapshots.

2.4. Unsteady Supersonic Boundary-Layer Flow

A laminar Mach 1.6 boundary layer flow over a flat-plate was simulated with an in-house developed finite-volume code [14]. The inflow momentum thickness Reynolds number was $Re_\vartheta = 113$. Velocities were made dimensionless with the freestream velocity, $U_\infty = 555.65$ m/s, and length scales were made dimensionless with $L_{ref} = 2.862$ mm. The Reynolds number based on U_∞ and L_{ref} was 100,000. The mesh had 200 cells in the streamwise direction and 300 cells in the wall-normal direction (Figure 2), for a total of 60,000 cells. A four cell-wide wall-normal blowing and suction slot near the inflow boundary was employed to introduce Tollmien–Schlichting waves. The forcing amplitude and frequency were 10^{-5} and 1.256. More details about this case can be found in Thumm et al. [19] and Gross and Fasel [20].

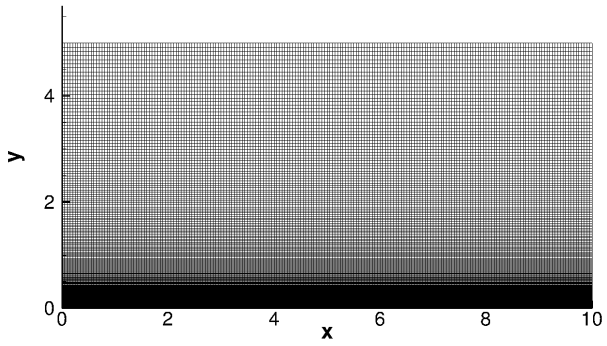

Figure 2. Computational grid for supersonic boundary layer.

2.4.1. Training Data

A database of 1000 snapshots of the u and v-velocity were recorded over a time interval of 0.13 milliseconds at a constant sampling rate. Of the 1000 snapshots, 700 were randomly chosen for AE training and the remaining 300 snapshots were used for validation. Only the near-wall region where the Tollmien–Schlichting waves reside, $0 < y < 0.25$, was analyzed.

2.4.2. Model Architecture

Details of the AE architecture for the decomposition of the supersonic boundary layer flow into two modes are provided in Table 2. The hyperbolic tangent function was chosen as activation function for all layers. The MSE was chosen as loss function. Using the Adam optimizer, early stop, and a patience of 50, training was completed after 1638 epochs. The validation loss at the end of the training was 5.2342×10^{-5}.

Table 2. AE network architecture for supersonic boundary layer.

Encoder	
Layer	Output Shape
Input	(160, 160, 2)
1st Conv(3, 3, 16)	(160, 160, 16)
1st MaxPooling	(80, 80, 16)
2nd Conv(3, 3, 8)	(80, 80, 8)
2nd MaxPooling	(40, 40, 8)
3rd Conv(3, 3, 8)	(40, 40, 8)
3rd MaxPooling	(20, 20, 8)
4th Conv(3, 3, 4)	(20, 20, 4)
4th MaxPooling	(10, 10, 4)
5th Conv(3, 3, 4)	(8, 8, 4)
5th MaxPooling	(4, 4, 4)
Flatten	(64, 1)
FCNN (Latent Vector)	(2, 1)
Decoder	
Layer	Output Shape
FCNN output	(64, 1)
Reshape	(4, 4, 4)
1st upsampling	(8, 8, 4)
Zero Padding (1, 1)	(10, 10, 4)
5th Conv(3, 3, 4)	(10, 10, 4)
2nd upsampling	(20, 20, 4)
6th Conv(3, 3, 8)	(20, 20, 8)
3rd upsampling	(40, 40, 8)
7th Conv(3, 3, 8)	(40, 40, 8)
4th upsampling	(80, 80, 8)
8th Conv(3, 3, 16)	(80, 80, 16)
5th upsampling	(160, 160, 16)
9th Conv(3, 3, 2)	(160, 160, 2)

2.4.3. Proper Orthogonal Decomposition

The unsteady supersonic boundary-layer flow was also analyzed with the POD. For the POD, 512 snapshots from the training database were processed.

2.5. Steady Hypersonic Boundary-Layer Flow

The transformed compressible boundary-layer Equations [21],

$$(Cf'')' + ff'' + \left(\frac{\rho_e}{\rho} - f'^2\right)\beta = 0\,, \tag{8}$$

$$(Cq')' + fq' = 0\,, \tag{9}$$

$$(Cg')' + Prfg' = (1 - Pr)\frac{(\gamma - 1)M^2}{1 + \frac{(\gamma-1)}{2}M^2}[C(f'f''\cos^2\Phi + qq'\sin^2\Phi)]'\,, \tag{10}$$

with Φ as the angle between the velocity vector outside the boundary-layer and the streamwise direction, were solved to obtain velocity and temperature profiles for a range of pressure gradients, β = ($-0.005, -0.004, -0.003, -0.002, -0.001, 0, 0.001$), and temperature ratios, $T_w/T_{t,\infty}$ = (0.2, 0.4, 0.6, 0.8, 1). The cross-flow velocity component was set to zero, $q = 0$. Additional details about the boundary layer equations are provided in Barraza et al. [22]. The freestream conditions, Mach number, Reynolds number, Prandtl number and isentropic exponent for the boundary layer calculations are summarized in Table 3.

Table 3. Conditions for hypersonic boundary layer calculations.

M	6
Re	10,822,430
T_∞	51.219 K
P_∞	611.362 Pa
Pr	0.71
γ	1.4

2.5.1. Training Data

A total of 40 velocity and temperature profiles, evenly distributed over a streamwise interval of 0.0065 m $\leq x \leq$ 1 m, were used to build the training database. The u and T profiles were made dimensionless with the freestream velocity and temperature and sampled at constant intervals from the wall to the boundary layer edge (80 samples each). The total number of profiles in the training database was $40 \times 7 \times 5 = 1400$.

2.5.2. Model Architecture

Two different approaches were employed for decomposing and reconstructing the hypersonic boundary layer profiles. The first approach is based on an AE. The network architecture is summarized in Table 4. The AE encodes the u and T profiles into a latent vector, $\mathbf{r}$, with three components. The entire latent vector is then decoded to reconstruct the profiles (with 80 points across the boundary layer thickness; i.e., the same resolution as for the input).

Table 4. Network structure for AE-based analysis of hypersonic boundary layer profiles.

Encoder	
Layer	Output Shape
Input	(80, 1)
1st 1D-Conv(3, 16)	(80, 16)
1st MaxPooling	(40, 16)
2nd 1D-Conv(3, 8)	(40, 8)
2nd MaxPooling	(20, 8)
3rd 1D-Conv(3, 4)	(20, 4)
3rd MaxPooling	(10, 4)
4th 1D-Conv(3, 2)	(10, 2)
4th MaxPooling	(5, 2)
Flatten	(10, 1)
FCNN (Latent Vector)	(3, 1)
Decoder	
Layer	Output Shape
FCNN output	(10, 1)
Reshape	(5, 2)
1st upsampling	(10, 2)
5th 1D-Conv(3, 4)	(10, 4)
2nd upsampling	(20, 2)
6th 1D-Conv(3, 8)	(20, 8)
3rd upsampling	(40, 4)
7th 1D-Conv(3, 16)	(40, 16)
4th upsampling	(80, 8)
8th 1D-Conv(3, 1)	(80, 1)

The second approach is based on three convolutional neural networks (CNNs) that are trained with u-velocity and temperature profiles (80 data points across boundary layer), and y values for specific Re_x, β, edge Mach numbers, M_e, edge temperatures, T_e, edge densities, ρ_e, edge viscosities, μ_e, edge pressures, P_e, and temperature ratios, $T_w/T_{t,\infty}$ (i.e.,

wall temperature to freestream total temperature). The wall-temperature ratios, pressure gradients, β, and x range are the same as for the first approach (Sections 2, 2.5 and 2.5.1). The range for the remaining parameters is

$$\begin{aligned} 54{,}058 \leq \quad & Re_x & \leq 10{,}822{,}430 \\ 5.17 \leq \quad & M_e & \leq 6.188 \\ 505.26\ \text{Pa} \leq \quad & P_e & \leq 1486\ \text{Pa} \\ 3.1 \times 10^{-6}\ \text{kg/(m} \cdot \text{s)} \leq \quad & \mu_e & \leq 4.433 \times 10^{-6}\ \text{kg/(m} \cdot \text{s)} \\ 48.5\ \text{K} \leq \quad & T_e & \leq 66.02\ \text{K} \\ 0.036\ \text{kg/m}^3 \leq \quad & \rho_e & \leq 0.0784\ \text{kg/m}^3 \end{aligned} \tag{11}$$

The NN architecture for the three CNNs is shown in Table 5.

Table 5. Network structure for CNN-based analysis of hypersonic boundary layer profiles.

Convolutional Neural Network	
Layer	Output Shape
Input	(8, 1)
Fully Connected Layer	(20, 1)
Reshape	(5, 4)
1st upsampling	(10, 4)
1st 1D-Conv(3, 8)	(10, 8)
2nd upsampling	(20, 8)
2nd 1D-Conv(3, 16)	(20, 16)
3rd upsampling	(40, 16)
3rd 1D-Conv(3, 32)	(40, 32)
3rd upsampling	(80, 32)
4th 1D-Conv(3, 1)	(80, 1)

3. Results and Discussion

3.1. Cylinder Wake Flow

An instantaneous visualization of the spanwise vorticity (Figure 3) reveals a von Kármán vortex street. The two modes obtained from the MD-CNN-AE are visualized in Figure 4.

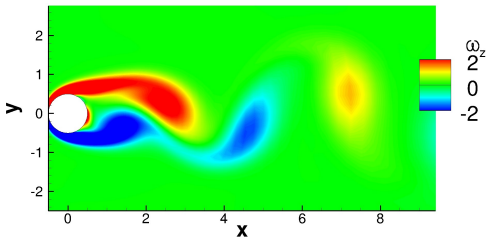

Figure 3. Instantaneous visualization of spanwise vorticity, $-2 < \omega_z < 2$.

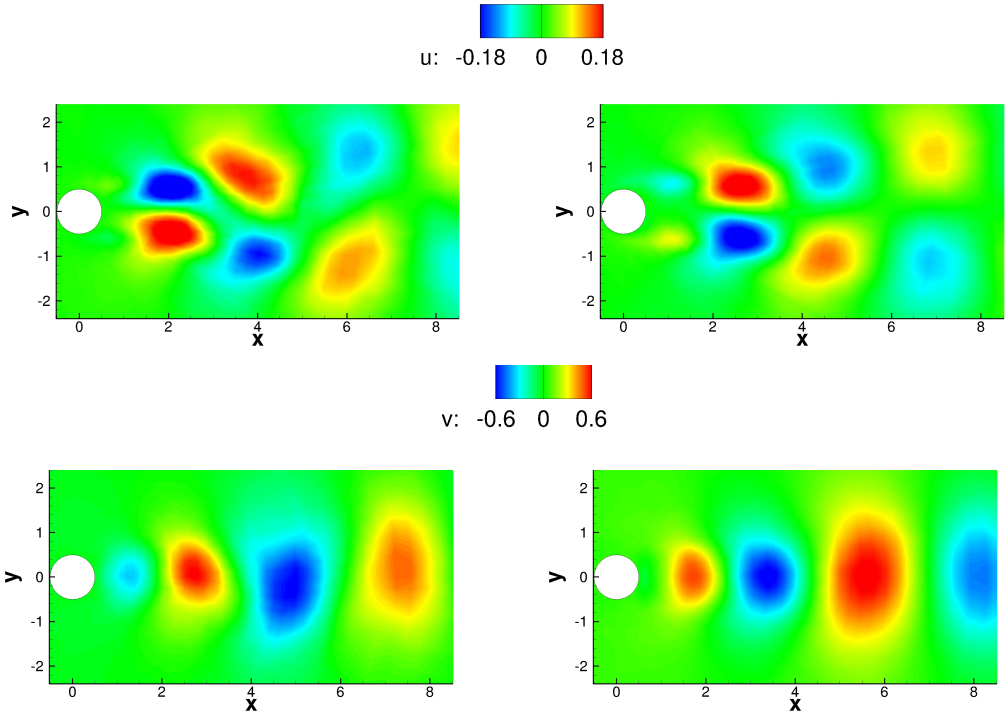

Figure 4. Autoencoder modes: u-velocity (**top**) and v-velocity (**bottom**) contours.

The two leading modes obtained from the POD are presented in Figure 5. As mentioned earlier, AE decomposition was performed using 2560 snapshots. The AE and POD modes share many similarities: The u-contours are asymmetric with respect to $y = 0$ and the v-contours are symmetric with respect to $y = 0$. For both mode pairs (AE and POD), the second mode is shifted by a quarter wavelength with respect to the first mode. For the POD, mode pairs are known to capture traveling waves [1], and the same appears to be the case for the AE modes.

Reconstructions of the flow field from the leading AE and POD modes are presented in Figure 6. Qualitatively, the AE reconstruction appears closer to the original flow field. This observation is in agreement with Murata et al. [5], even though the present results are based on an O-grid with lower resolution and fewer snapshots.

The root-mean-squared error (RMSE) and the absolute error were computed for both reconstructions. The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{l \times m} \sum_{i=1}^{l} \sum_{j=1}^{m} (\phi_{i,j} - \hat{\phi}_{i,j})^2} \tag{12}$$

where ϕ and $\hat{\phi}$ are the u or v velocity component of the original and reconstructed flow field. The grid point indices are denoted by i and j. The absolute error is defined as $\phi_{\text{error}} = |\phi - \hat{\phi}|$. Figure 7a,b show the absolute error for the AE and POD reconstructions, where the left column is the u error and the right column is the v error. The absolute error for the POD reconstruction is larger than for the AE reconstruction, in particular for the wake region, which is dominated by unsteady flow structures. The RMSE values are compared in Table 6. Compared to the POD reconstruction, the RMSE for the AE reconstruction is 47% lower for the u velocity and 39% lower for the v-velocity. This indicates that for a highly truncated modal basis (mean flow plus two modes), the AE modes represent the unsteady flow field better than the POD modes. It is believed that this can be attributed to the ability of the AE to capture nonlinearities. A drawback of the AE approach is however that it requires more snapshots for network training.

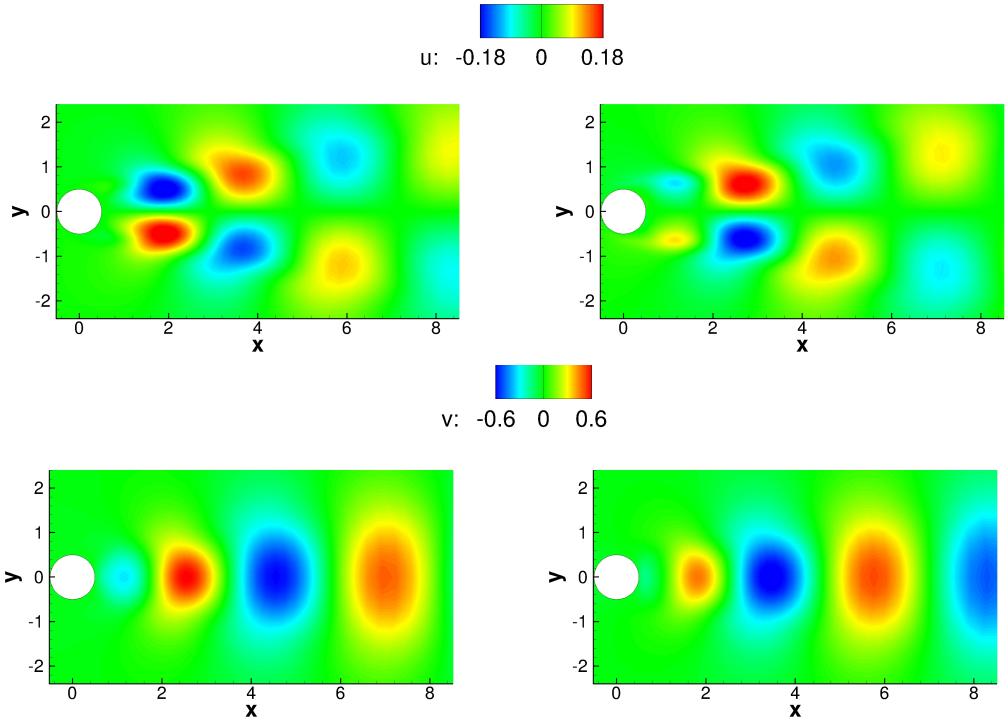

Figure 5. Leading POD modes: u-velocity (**top**) and v-velocity (**bottom**) contours.

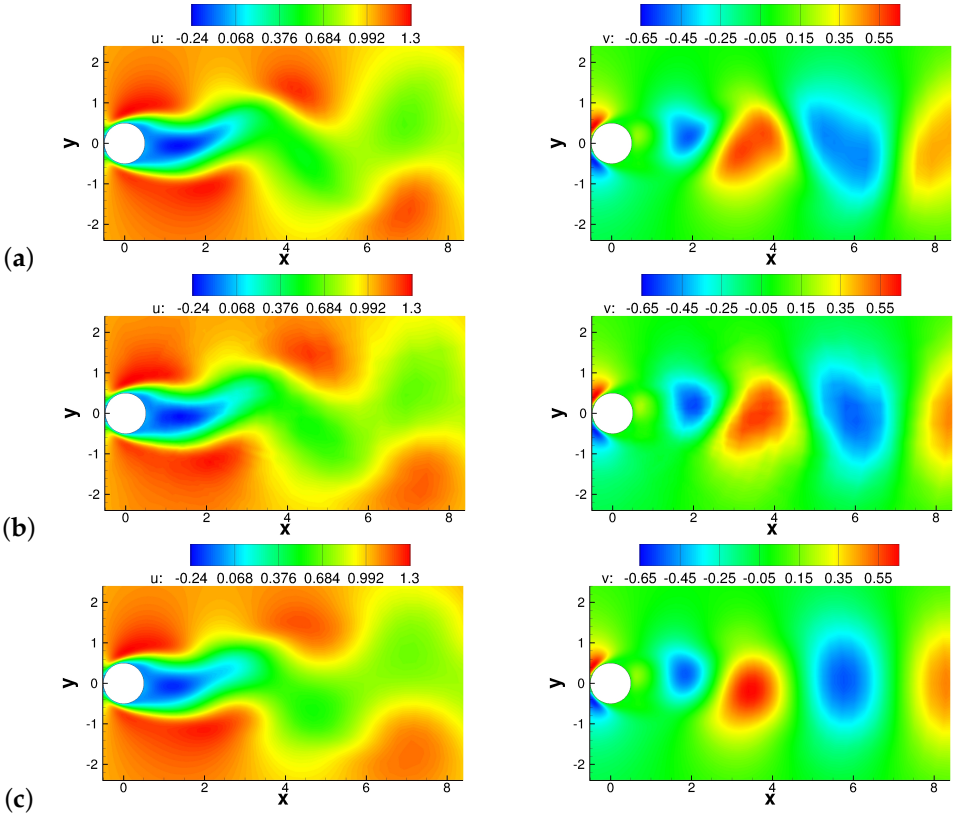

Figure 6. Contours of u-velocity (**left**) and v-velocity (**right**). (**a**) Original flow field, (**b**) AE reconstruction and (**c**) POD reconstruction.

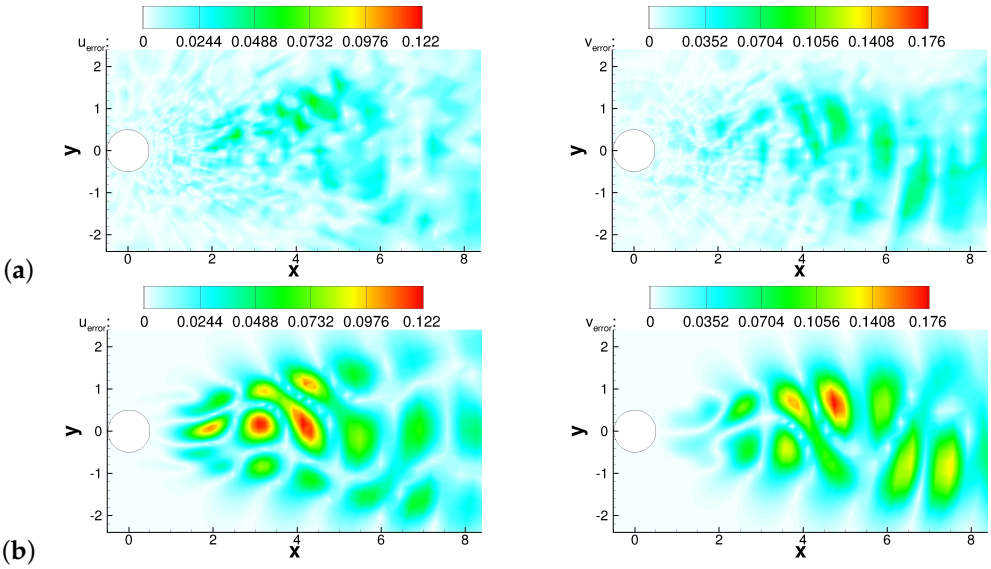

Figure 7. Absolute error between original flow and (**a**) AE reconstruction and (**b**) POD reconstruction. Left and right column correspond to u and v velocity error.

Table 6. Comparison of RMSE for u and v velocity component.

	u RMSE	v RMSE
AE	0.00476	0.00648
POD	0.00894	0.01058

Figure 8 displays the SPOD spectrum as a function of the Strouhal number $St = fD/U_\infty$. The eigenvalues for the first three modes demonstrate a clear dominance of the first mode, especially at lower Strouhal numbers, indicating the presence of a coherent low-frequency behavior in the flow. Figures 9 and 10 present the the u-velocity and v-velocity component of the SPOD modes for two different Strouhal numbers: $St = 0.171$ (left column) and $St = 0.328$ (right column). The SPOD modes for $St = 0.171$ exhibit a strong spatial coherence and are well separated from the higher modes, which is consistent with the observation that the flow is dominated by a primary shedding frequency. This also explains why the MD-CNN-AE can represent the instantaneous flow field so well with only two AE modes plus the mean flow, as shown in Figure 6.

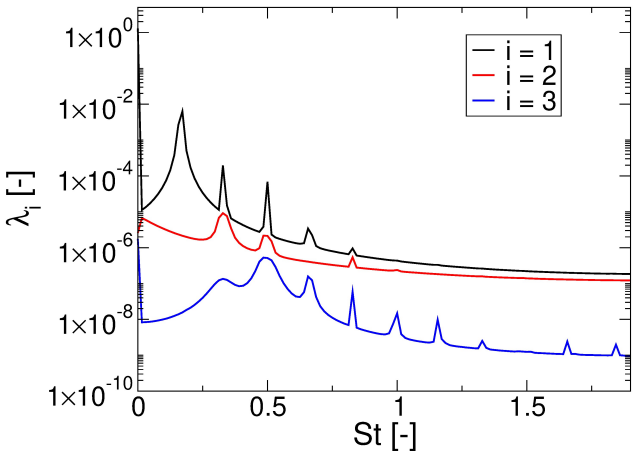

Figure 8. SPOD spectrum as a function of $St = fD/U_\infty$.

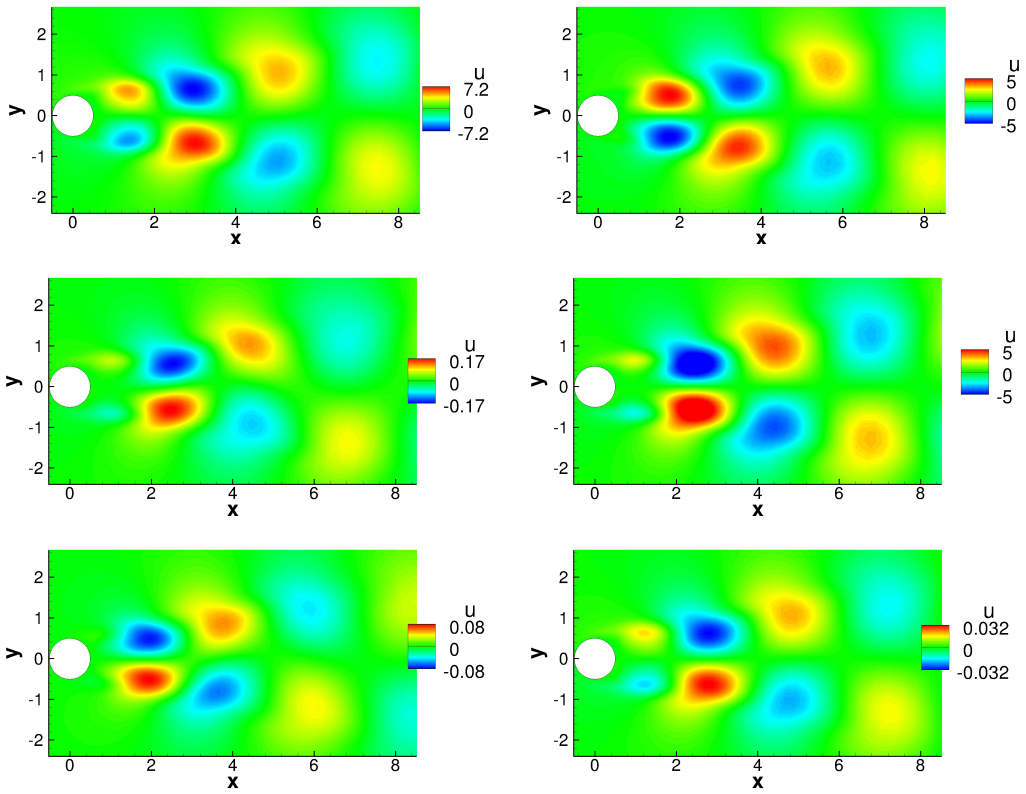

Figure 9. SPOD modes for u-velocity. (**Left column**): $St = fD/U_\infty = 0.171$. (**Right column**): $St = fD/U_\infty = 0.328$.

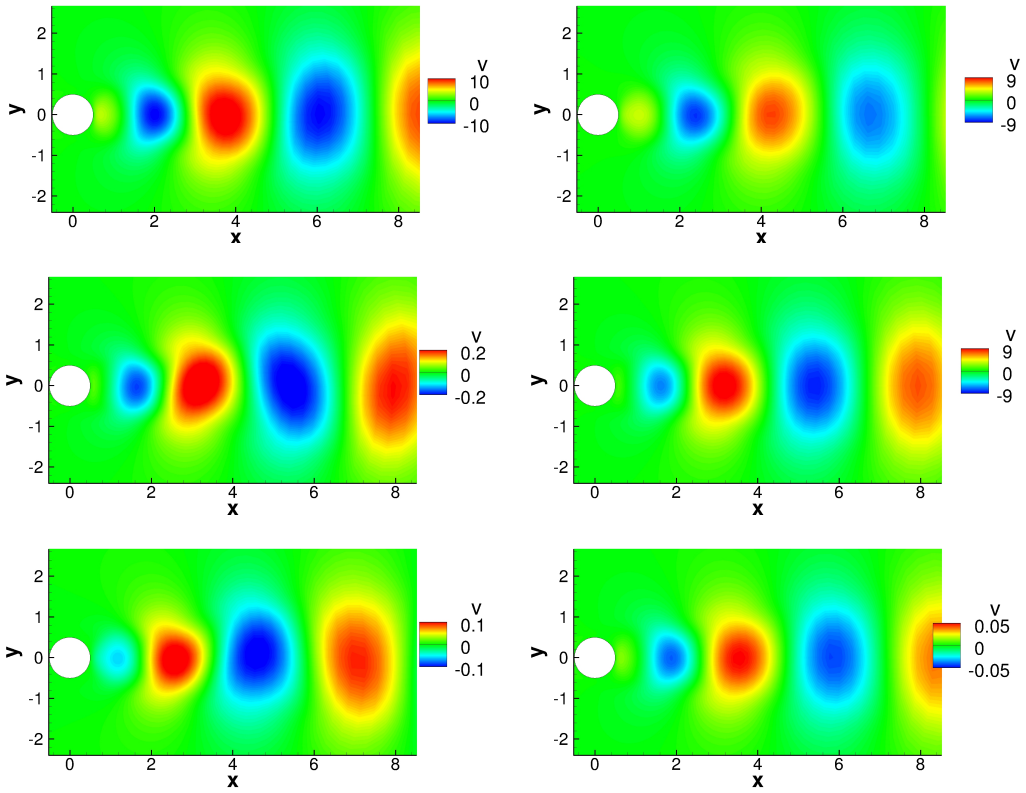

Figure 10. SPOD modes for v-velocity. (**Left column**): $St = fD/U_\infty = 0.171$. (**Right column**): $St = fD/U_\infty = 0.328$.

3.2. Unsteady Supersonic Boundary-Layer Flow

Contours of the base flow u-velocity and the u-disturbance velocity obtained directly from the simulation are provided in Figure 11. The figure of the disturbance velocity reveals Tollmien–Schlichting waves that first grow and then decay in the streamwise direction. As the disturbance input is harmonic and the flow is time-periodic, SPOD is not required and POD provides time-coefficients that are harmonic in time. The AE modes and dominant two POD modes are compared in Figures 12 and 13. Overall, the AE modes are very similar to the POD modes. As for the cylinder flow, the modes are phase-shifted by a quarter wavelength in the streamwise direction. This behavior can be explained by the fact that the Tollmien–Schlichting waves are traveling in the streamwise direction.

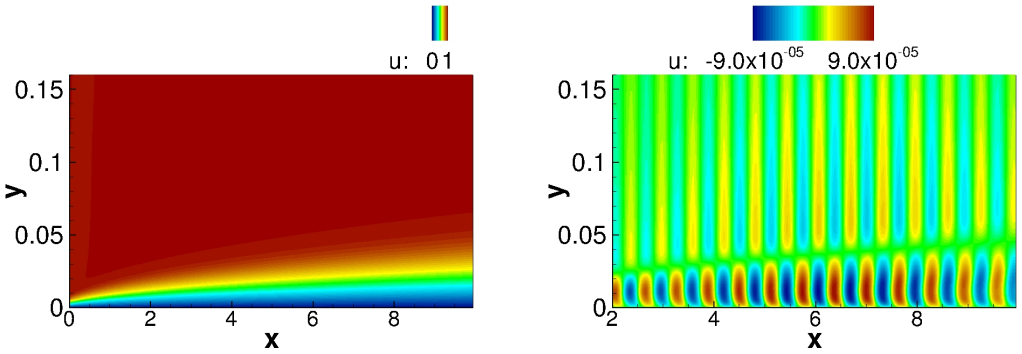

Figure 11. Base flow u-velocity and u-disturbance velocity.

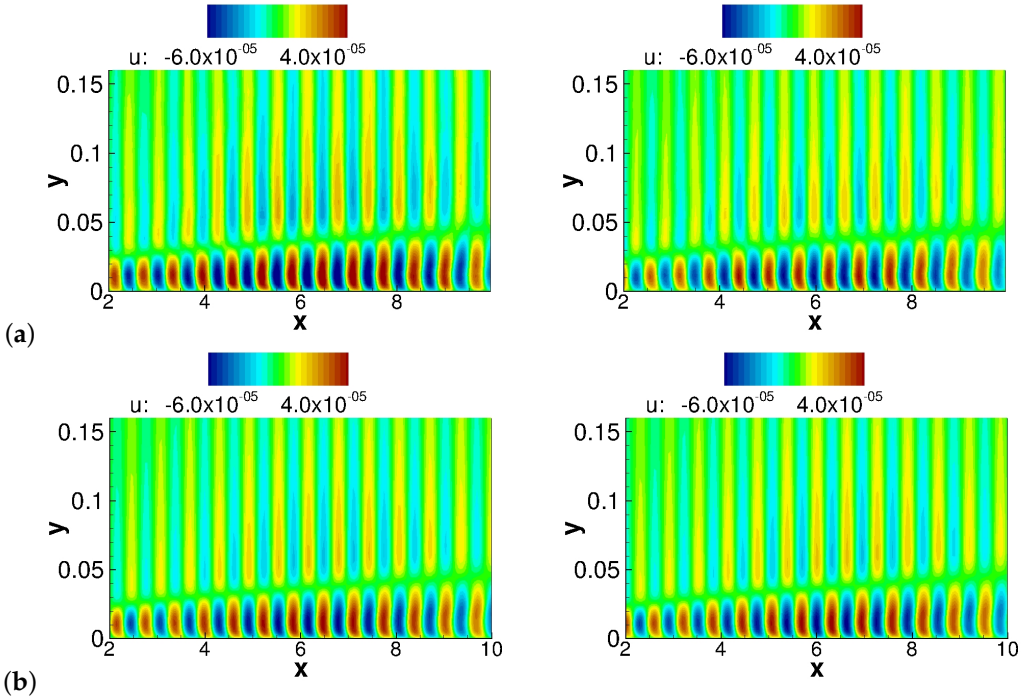

Figure 12. Modes obtained from (**a**) autoencoder and (**b**) POD: u-velocity contours.

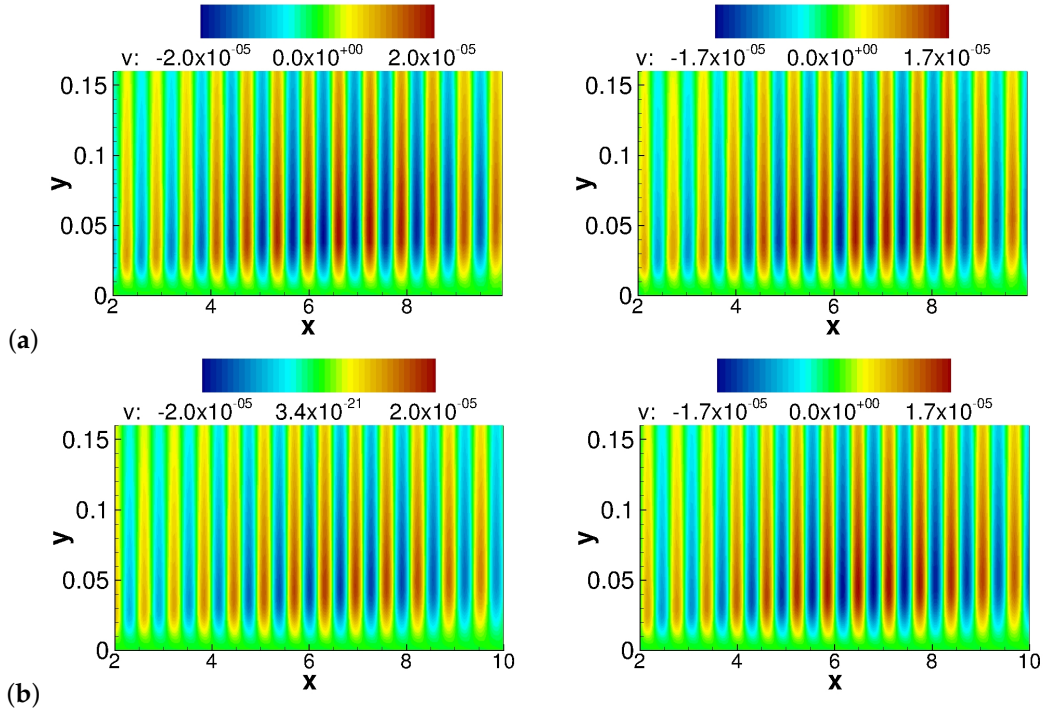

Figure 13. Modes obtained from (**a**) autoencoder and (**b**) POD: v-velocity contours.

3.3. Steady Hypersonic Boundary-Layer Flow

The AE results are discussed first. In Figure 14, u-profiles obtained directly from the similarity solutions are compared with the reconstructed profiles for three streamwise locations, x, pressure gradients, β, and wall-temperature ratios, $T_w/T_{t,\infty}$. Although the agreement is overall very good, small inaccuracies can be observed. For example, in Figure 14a, the reconstructed profile overshoots the reference profile by a minimal amount near the boundary layer edge. The outputs of each AE layer for one of the cases, $x = 0.278$ m, $\beta = 0$, $T_w/T_{t,\infty} = 0.4$, are depicted in Figure 15. In the figures, the different colors signify the outputs from different filters of the convolutional layers. As is typical for neural networks, no immediate sense can be made of the intermediate network states.

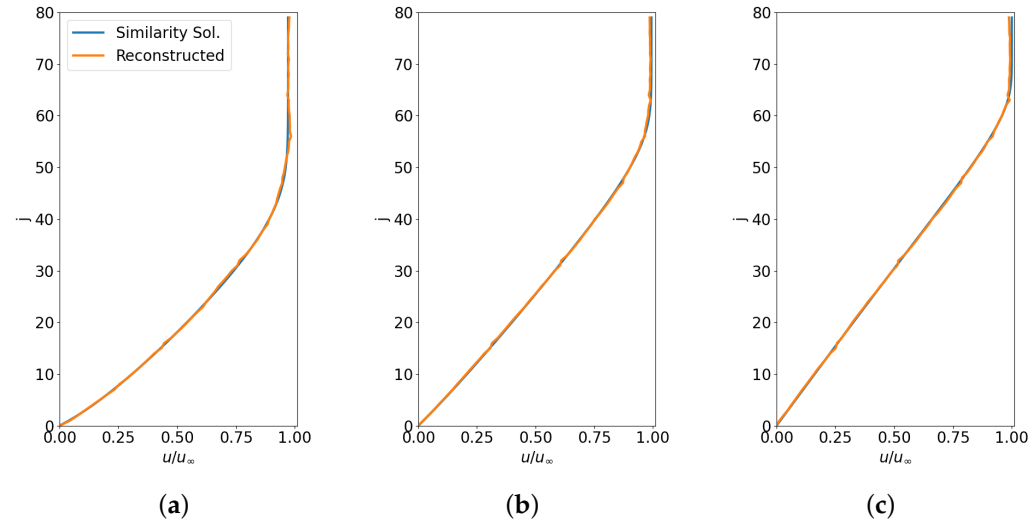

Figure 14. Comparison between input (blue) and reconstructed (orange) u-profiles for (**a**) $x = 0.085$ m, $\beta = -0.004$, $T_w/T_{t,\infty} = 0.2$, (**b**) $x = 0.278$ m, $\beta = 0$, $T_w/T_{t,\infty} = 0.4$, and (**c**) $x = 1$ m, $\beta = 0.001$, $T_w/T_{t,\infty} = 0.8$.

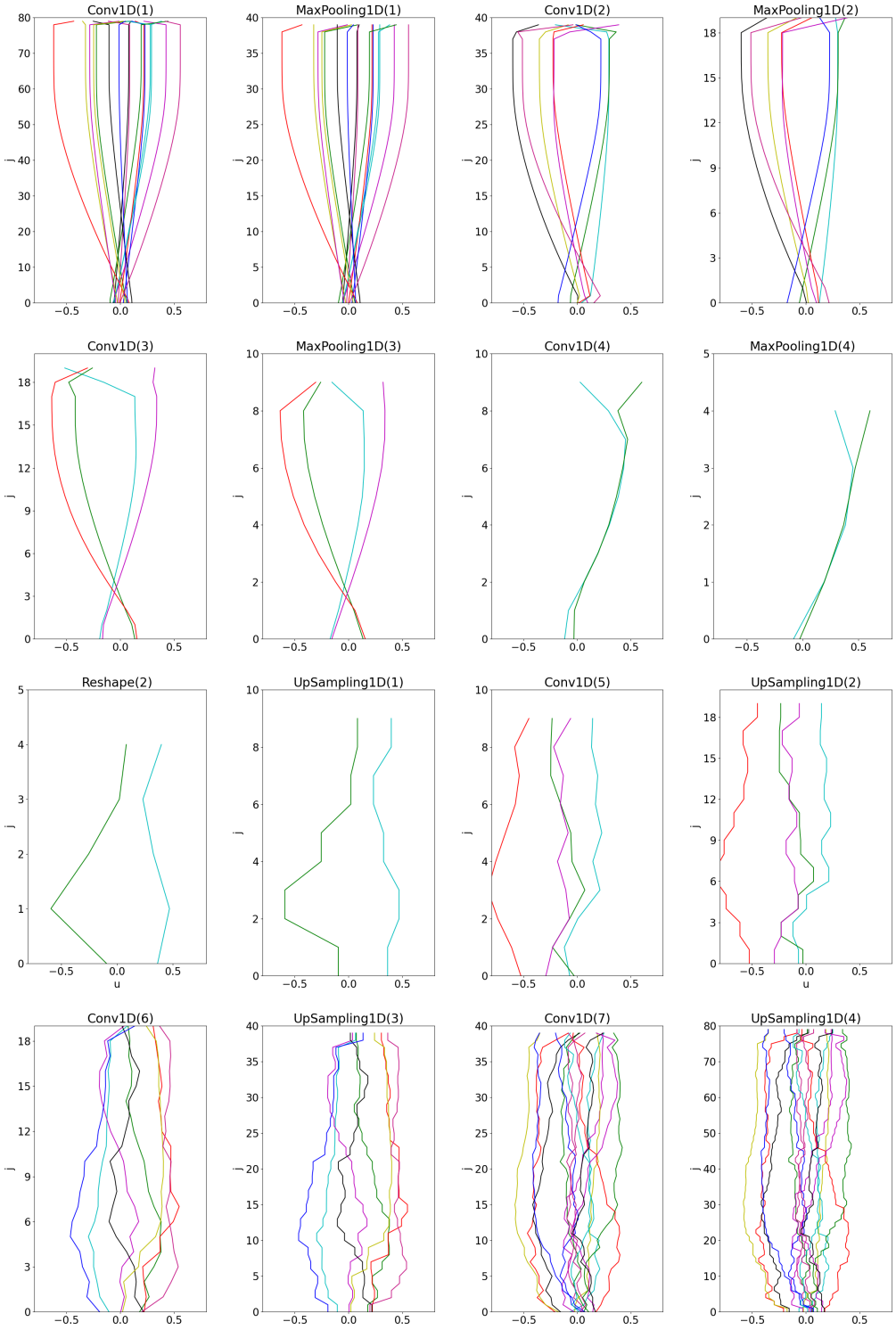

Figure 15. Outputs of each AE layer for $x = 0.278$ m, $\beta = 0$, $T_w/T_{t,\infty} = 0.4$. The title above each sub-figure indicates the name of the hidden layer for which the output is plotted. Each color represent a different filter of each convolutional layer.

In an attempt to interpret the latent vector, its three components, r_1, r_2, and r_3, are plotted against the x-coordinate, pressure gradient, β, and wall-temperature ratio, $T_w/T_{t,\infty}$, in Figure 16. For the images in the left column of Figure 16, the pressure gradient parameter and temperature ratio were fixed, $\beta = -0.003$ and $T_w/T_{t,\infty} = 0.6$; for the images in the

center column of Figure 16, $x = 1$ m and $T_w/T_{t,\infty} = 0.6$; and for the images in the right column of Figure 16, $x = 1$ m and $\beta = -0.003$.

The three components of the latent vector **r** show a clear dependence on the pressure gradient and wall-temperature ratio. Specifically, r_1 and r_3 display notable variations with changes in β and $T_w/T_{t,\infty}$. r_2 also shows variations, though to a lesser extent compared to r_1 and r_3.

In contrast, the correlation between the latent vector components and the x-coordinate is weak. This is not surprising since the profiles were obtained from similarity solutions with a normalized y-coordinate, making the spatial variation along x less significant in the encoded representation. These observations suggest that the autoencoder primarily captures the effects of the pressure gradient and wall-temperature ratio in its latent space, considering them as key features of the input profiles.

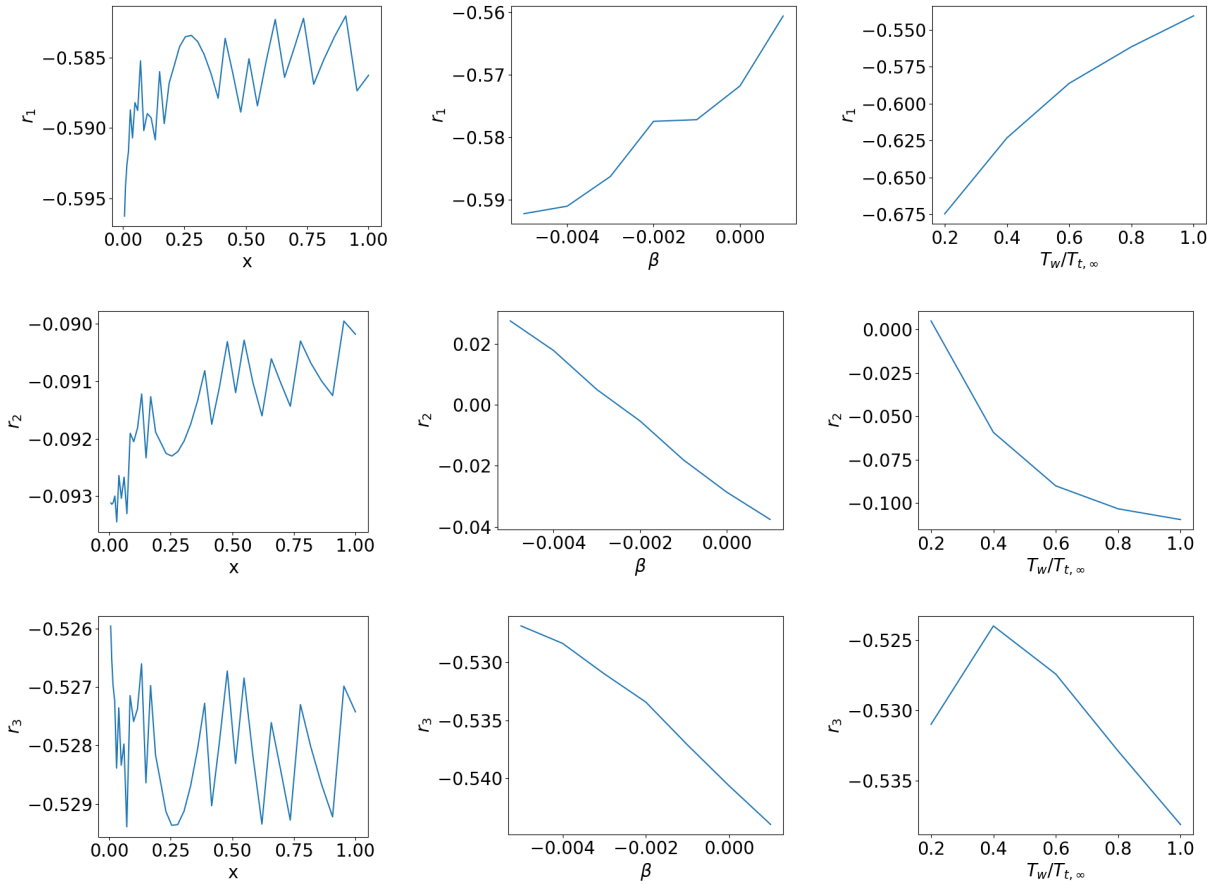

Figure 16. Components of latent vector, **r**.

The results from the CNN-based approach are discussed next. In Figure 17, velocity and temperature profiles for $\beta = -0.0025$, $M_e = 5.585$, $T_e = 58.01$ K, $\rho_e = 0.0567$ kg/m^3, $\mu_e = 3.82 \times 10^{-6}$ kg/(ms), $P_e = 945.6$ Pa, $Re_x =$ 7,351,901 and $T_w/T_{t,\infty} = 0.5$ are compared. Since the chosen parameters were not part of the training data set but inside the range of the training data, this suggests that the model can interpolate between data points. The profiles obtained from the similarity solution and the estimated profiles obtained from the CNN are in good agreement. The estimated profiles are not as smooth as the reference profiles. If deemed necessary for a practical application, the estimated profiles could be smoothed.

A data point far outside the bounds of the training data set (except for the wall temperature ratio) was also considered, $\beta = -0.006$, $M_e = 5.06$, $T_e = 68.63$ K, $\rho_e = 0.0864$ kg/m^3, $\mu_e = 4.63 \times 10^{-6}$ kg/(ms), $P_e = 1703.2$ Pa, $Re_x = 11.943 \times 10^6$ and $T_w/T_{t,\infty} = 0.6$. For the chosen data point, the CNN overpredicts the boundary-layer thickness (Figure 18) and

the prediction is noisy. Nevertheless, even though the chosen parameters are outside the bounds of the training set, the near-wall temperature distribution is well represented.

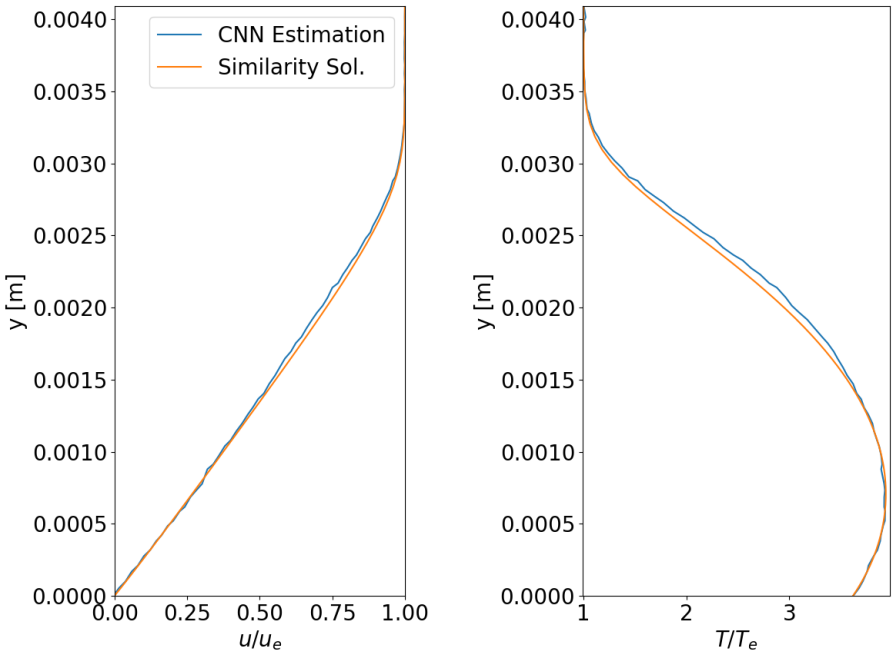

Figure 17. Similarity solution and estimated CNN profiles for $\beta = -0.0025$, $M_e = 5.585$, $T_e = 58.01$ K, $\rho_e = 0.0567\,\mathrm{kg/m^3}$, $\mu_e = 3.82 \times 10^{-6}\,\mathrm{kg/(ms)}$, $P_e = 945.6$ Pa, $Re_x = 7{,}351{,}901$ and $T_w/T_{t,\infty} = 0.5$.

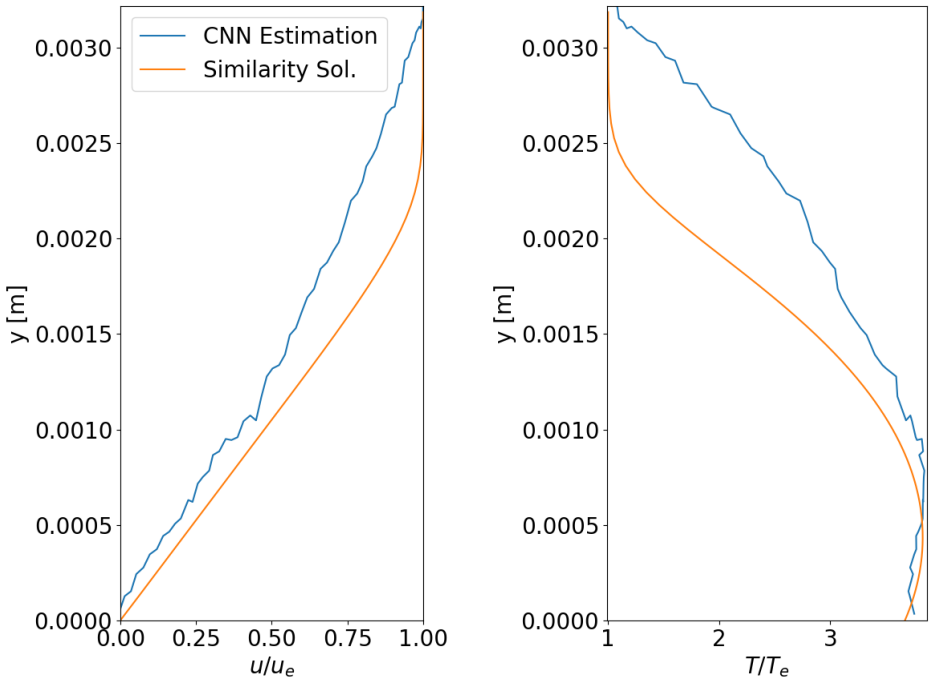

Figure 18. Similarity solution and estimated CNN profiles for $\beta = -0.006$, $M_e = 5.06$, $T_e = 68.63$ K, $\rho_e = 0.0864\,\mathrm{kg/m^3}$, $\mu_e = 4.63 \times 10^{-6}\,\mathrm{kg/(ms)}$, $P_e = 1703.2$ Pa, $Re_x = 11.943 \times 10^6$ and $T_w/T_{t,\infty} = 0.6$.

4. Conclusions

This paper explores the suitability of machine learning (ML)-based approaches for the data-driven modal decomposition of unsteady fluid flows and for the compression of steady boundary layer profiles into a compact sub-space. The von Kármán vortex street behind a circular cylinder and Tollmien–Schlichting waves in a compressible Mach 1.6 flat plate boundary layer were analyzed with an autoencoder (AE) and with the proper orthogonal decomposition (POD).

The modes obtained from the AE and POD were very similar. For the circular cylinder wake, the root-mean-squared error of an instantaneous flow field reconstructured from the leading three AE and POD modes revealed that the AE modes represent the flow field better. It is believed that this can be attributed to the ability of the AE to capture nonlinearities.

The present results suggest that AE-based approaches are well-suited for identifying modes (which capture large-scale coherent fluid motion) that are chiefly responsible for the observed fluid dynamics. Furthermore, as AE-based modes provide a compact basis, they are well suited for the development of reduced order models. For larger datasets or flows that are governed by strongly nonlinear phenomena, AE-based modal decompositions are a very attractive alternative to POD since they do not require the solution of large singular value problems and since the underlying neural networks can by design capture nonlinearities. A drawback of the AE approach is however that it requires large training datasets.

Both an AE and a convolutional neural network (CNN) were employed for generalizing hypersonic laminar boundary layer profiles based on the streamwise coordinate, pressure gradient, and wall temperature ratio. For the AE, the latent vector correlated well with the pressure gradient and temperature ratio, but not with the streamwise coordinate. This was attributed to the fact that the AE was trained with scaled profiles.

A new network architecture, which combines a one-layer perceptron and a CNN, was proposed to estimate the velocity and temperature profiles from the boundary-layer edge parameters. When the boundary layer edge parameters were chosen within the bounds of the training database, the estimated profiles were in good agreement with the similarity solution profiles. However, estimation of profiles outside the bounds of the training database was found to be challenging. Different architectures or a larger training database could prove helpful for overcoming this limitation.

In conclusion, the present results indicate that ML-based approaches, particularly AE, are effective for modal decomposition and reduced-order modeling of unsteady fluid flows, while the proposed CNN architecture shows promise for generalizing hypersonic boundary layer profiles. Further research into network architectures and training data diversity is recommended to enhance the robustness and accuracy of the methods.

Author Contributions: Conceptualization, B.B. and A.G.; methodology, B.B.; software, B.B.; validation, B.B.; formal analysis, B.B.; investigation, B.B.; resources, A.G.; data curation, B.B.; writing—original draft preparation, B.B.; writing—review and editing, A.G.; visualization, B.B.; project administration, A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available upon reasonable request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Taira, K.; Hemati, M.S.; Brunton, S.L.; Sun, Y.; Duraisamy, K.; Bagheri, S.; Dawson, S.T.; Yeh, C.A. Modal analysis of fluid flows: Applications and outlook. *AIAA J.* **2020**, *58*, 998–1022. [CrossRef]
2. Lumley, J.L. *The Structure of Inhomogeneous Turbulent Flows*. Atmospheric Turbulence and Radio Wave Propagation: Nauka, Moscow, 1967; pp. 166–178.
3. Schmid, P.J. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **2010**, *656*, 5–28. [CrossRef]

4. Sirovich, L. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Q. Appl. Math.* **1987**, *45*, 561–571. [CrossRef]
5. Murata, T.; Fukami, K.; Fukagata, K. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *J. Fluid Mech.* **2020**, *882*. [CrossRef]
6. Noack, B.R.; Afanasiev, K.; MORZYŃSKI, M.; Tadmor, G.; Thiele, F. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *J. Fluid Mech.* **2003**, *497*, 335–363. [CrossRef]
7. Schmidt, O.T.; Colonius, T. Guide to spectral proper orthogonal decomposition. *AIAA J.* **2020**, *58*, 1023–1033. [CrossRef]
8. Chiatto, M.; Shang, J.K.; de Luca, L.; Grasso, F. Insights into low Reynolds flow past finite curved cylinders. *Phys. Fluids* **2021**, *33*, 035150. [CrossRef]
9. Wen, G.; Gross, A. Local linear stability analysis of laminar separation bubble for helicopter blade section undergoing dynamic stall. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; p. 2241.
10. Brunton, S.L.; Noack, B.R.; Koumoutsakos, P. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **2020**, *52*, 477–508. [CrossRef]
11. Kutz, J.N. Deep learning in fluid dynamics. *J. Fluid Mech.* **2017**, *814*, 1–4. [CrossRef]
12. Paredes, P.; Venkatachari, B.S.; Choudhari, M.M.; Li, F.; Chang, C.L.; Irfan, M.I.; Xiao, H. Toward Transition Modeling in a Hypersonic Boundary Layer at Flight Conditions. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; p. 0103.
13. Milano, M.; Koumoutsakos, P. Neural network modeling for near wall turbulent flow. *J. Comput. Phys.* **2002**, *182*, 1–26. [CrossRef]
14. Gross, A.; Fasel, H.F. High-order accurate numerical method for complex flows. *AIAA J.* **2008**, *46*, 204–214. [CrossRef]
15. Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 1 March 2022).
16. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
17. Rowley, C.W.; Colonius, T.; Murray, R.M. Model reduction for compressible flows using POD and Galerkin projection. *Phys. D Nonlinear Phenom.* **2004**, *189*, 115–129. [CrossRef]
18. Towne, A.; Schmidt, O.T.; Colonius, T. Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. *J. Fluid Mech.* **2018**, *847*, 821–867. [CrossRef]
19. Thumm, A.; Wolz, W.; Fasel, H. Numerical simulation of spatially growing three-dimensional disturbance waves in compressible boundary layers. In *Laminar-Turbulent Transition*; Springer: Berlin/Heidelberg, Germany, 1990; pp. 303–308.
20. Gross, A.; Fasel, H. Modification of ninth-order weighted essentially nonoscillatory scheme for mixed subsonic/supersonic flow. *AIAA J.* **2010**, *48*, 2698–2702. [CrossRef]
21. Tumin, A. Nonlinear interaction in a three-dimensional compressible boundary layer. In Proceedings of the 19th ICAS, Congress, Anaheim, CA, USA, 18–23 September 1994; pp. 1114–1123.
22. Barraza, B.; Gross, A.; Leinemann, M.; Hader, C.; Fasel, H.F. Transition Model for Second Mode and Crossflow Instabilities in Hypersonic Flow. In Proceedings of the AIAA SCITECH 2024 Forum, Orlando, FL, USA, 8–12 January 2024; p. 2189.