Advances in Aerodynamics

RESEARCH **Open Access**

Aerodynamic flow analysis using conditional convolutional autoencoder in various flow conditions and application to CFD-based design optimization

Wontae Hwang[1], Donggun Lee[1], Junghun Shin[2], Kumwon Cho[3] and Seongim Choi[1*]

*Correspondence:
schoi1@gist.ac.kr

[1] Department of Mechanical Engineering, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea
[2] OXR Convergence Research Center, Korea Institute of Science and Technology Information, Daejeon 34141, Republic of Korea
[3] Industry-University Cooperation Foundation, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea

Abstract

This study investigates the accuracy and efficiency of a convolutional autoencoder in predicting flow solutions of diverse characteristics, including strong local nonlinearity and unsteady wake vortices. Modifications to the standard U-net method were made suitable for non-Cartesian CFD mesh topology, enhancing solution accuracy. Additionally, conditions for predicting flows in unseen environments are integrated into a bottleneck layer between the encoder and decoder structures, guiding flow interpolation or extrapolation and parameter types. For direct comparison, this study uses a proper orthogonal decomposition (POD)-based ROM with linear reconstruction using dominant basis vectors from the flow solution space. Interpolation and extrapolation of generalized coordinates are performed using Gaussian process regression (GPR) and Long Short-Term Memory (LSTM) networks, respectively. The Conditional Unet (CUnet)'s accuracy is demonstrated through inviscid transonic airfoil flows, capturing shock waves effectively. Additionally, it can also be used for predicting the flow field of the three-dimensional shape of the Onera M6 wing. Vortex shedding flows around an Eppler airfoil at a 16-degree angle of attack in turbulent conditions were well-resolved, with root mean squared errors under 1% compared to full-order CFD results. Remarkably, the CUnet's computational efficiency is highlighted as the wall clock CPU time for these 2D flows was less than one second. Finally, the ROM's effectiveness is further validated through successful multi-point shape optimization, minimizing wave drag of RAE 2822 airfoils across subsonic to transonic conditions. This resulted in a maximum drag reduction of 37.38% at Mach 0.74 without performance degradation at off-design conditions.

Keywords: Reduced order model, Computational fluid dynamics, Deep learning, Design optimization

1 Introduction

Recent advancements in numerical analysis methods and the computational power of parallel GPU/CPU architectures have enabled the widespread utilization of Computational Fluid Dynamics (CFD) in the performance analysis of complex physical systems of aircraft, ships and turbomachinery. However, its direct application to the design optimization of these systems is still largely infeasible due to the requirement of repetitive function evaluations by tens of thousands of CFD calculations. To overcome this computational cost barrier, surrogate models or meta-models are generally employed. In the early stages, research focused on surrogate models capable of predicting a small number of scalar metrics that represent the system performance. Gaussian Process Regression (GPR) [1, 2], Radial Basis Functions (RBF) [3, 4], and Support Vector Machines (SVM) [5, 6] have been successfully utilized as meta-modeling techniques for accurately predicting lift and drag coefficients, enabling efficient design optimization with satisfactory accuracy.

However, practical design problems requiring high accuracy necessitate not only the aerodynamic coefficients but also the entire flow field distribution in two-dimensional or three-dimensional space, often with a need for transient flow analysis in certain cases. Therefore, extensive research has been conducted on reduced order modeling (ROM) techniques, and various dimensionality reduction methods have been introduced to facilitate the meta-modeling techniques with high-dimensional flow field data. Examples include Proper Orthogonal Decomposition (POD) [7–10], Reduced Basis Methods [11, 12], and Balancing Methods [13, 14]. Among these, the POD snapshot method has been widely used in various studies due to its non-intrusive implementation, independent of CFD solver and mesh type [15, 16]. The Dynamic Mode Decomposition (DMD) methods with some variations in the original principle were proposed for the flow prediction at future times [17, 18] along with the Discrete Empirical Interpolation Method (DEIM) [19, 20] to better predict flows of nonlinearities. For nonlinear feature extraction in time series data, Eivazi et al. [21] proposed a methodology of the ROM in combination with the long short-term memory (LSTM) with an autoencoder network for nonlinear dimensionality reduction. This approach is compared with other non-intrusive ROM methods and is shown to be effective in predicting the evolution of fluid flows. However, the principle of the linear reconstruction by orthogonal basis functions contains inherent limitations in predicting flows with strong and local nonlinearities.

On the other hand, in recent years, machine learning methods in the AI areas, including image processing in computer vision [22–24], are being successfully employed as a black box type of the ROMs. Kang et al. [25] utilized the POD to pre-determine low-dimensional data for autoencoders and applied the Variational Autoencoder (VAE) method to extract interpretable latent vectors and develop a flow prediction model for transonic flows based on these vectors. Catalani et al. [26] developed a Geodesic Convolutional Neural Network (GCNN) to better accommodate peculiarities in geometry of interest, and tested it using the RAE 2822 airfoil to predict high Mach number flows from the training set of low-Mach number flows. Hasegawa et al. [27] proposed a machine-learned reduced order model (ML-ROM) for the prediction of unsteady flows by combining a CNN-AE and an LSTM in a sequential training process. The method allowed for accurate prediction of flow fields even for unseen shapes of bluff bodies.

One of the important factors of the CNN methods is the feature extraction point topology for the image process. If the CFD mesh points need to be mapped to a structure of image pixels, there is a risk of introducing errors by interpolating the non-Cartesian discrete solution points obtained from high-fidelity CFD solvers onto a Cartesian grid, thus degrading the solution accuracy. Occlusion maps were employed in some studies to consider diverse geometric shapes using CNN [28]. In such cases, the model may inaccurately perceive the shape, leading to increased errors near the solid boundaries. If such models are employed in design optimization, they can adversely impact the results. Peng et al. [28] reconstructed the CFD solutions into image pixels, with solutions on the pixel-point topology interpolated from nearby CFD grid points. This transformation makes the unsteady airfoil flows easily trained to the CNN autoencoder. Deng et al. [29] proposed a method that combines a body-fitted mesh interpolation and deep learning techniques, using the U-net for the spatial network and Fourier Neural Operator (FNO) networks, for temporal prediction of unsteady flows.

In the present study, a modified CNN architecture is proposed as a Conditional Unet (CUnet), which extends the standard Unet [30] by incorporating structures that account for appropriate flow conditions and temporal steps. The advantage of the proposed model lies in its direct utilization of CFD mesh used for full-order model (FOM) CFD solver through simple matrix transformations during the model training process, which consequently avoids the resolution degradation caused by interpolation. That is, complex preprocessing steps are unnecessary [26, 27, 29], and the CUnet can fully replace expensive CFD solvers within the trained domain.

Another advantage developed in the present study is that to handle parameters of different types for the flow field prediction, i.e., geometric parameters, flow conditions, or physical time, a skip layer is inserted into the conventional Unet as a condition to the decoder. To predict time-dependent flows, changes to the architecture of the CUnet were minimal by the extrapolation of flow fields according to time steps by transforming the input data into a windowed dataset format commonly used in time-series model training.

The validation of the CUnet was performed by solving the flows over steady-state airfoils and the Onera M6 wing, as well as addressing unsteady flows and conducting multi-point airfoil shape optimization to reduce wave drag under various Mach number conditions. First, inviscid flows of the RAE 2822 airfoil were solved at a range of Mach numbers from 0.6 to 0.8 and angles of attack from 1° to 3° to calculate shock waves. Additionally, to assess the prediction accuracy of CUnet for nonlinear flow fields, we compared its predictions with the results obtained using the POD-GPR method, which is based on linear basis functions. Also, moving beyond the 2D shapes of airfoils in steady state, the flow fields at various Mach numbers for the 3D shape of the Onera M6 wing were resolved. Furthermore, to validate the accuracy of CUnet for time-dependent unsteady flows, we analyzed the wake vortex characteristics of the Eppler 387 airfoil at a high angle of attack of 16 degrees at a Reynolds number of 1.0×10^5. For the prediction of unsteady flows, we also compared the performance of CUnet with the POD-LSTM method as well as the FOM solutions in terms of accuracy and efficiency. The proposed model exhibited nearly identical results to the FOM solutions at significantly reduced cost (0.05% of the CFD computation cost).

Finally, the validated CUnet was applied to perform multi-point optimization with the goal of minimizing wave drag for four design points in the transonic airfoil. Unlike FOM CFD-based optimization, which incurs exponentially increasing computational costs with the number of design points, the CUnet-based optimization algorithm offers the

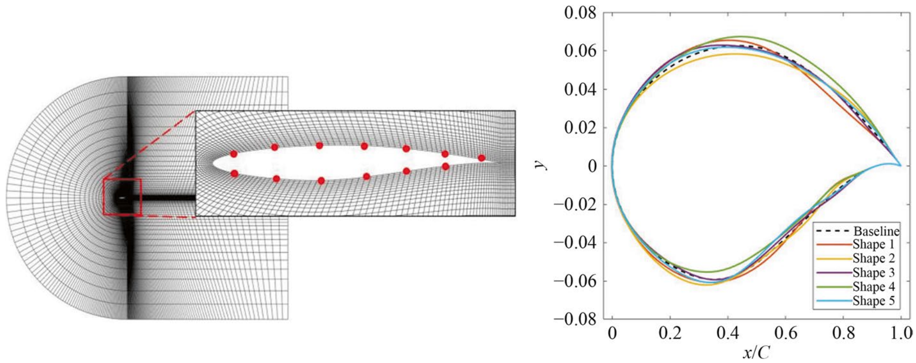

Fig. 1 C-type mesh and bump location of RAE 2822 airfoil (left) and examples of deformed airfoil (right)

advantage of comparable computational costs to single-point optimization. The results of single-point and multi-point optimization are directly compared across a wide range of Mach numbers, demonstrating the applicability of the CUnet-based optimization algorithm to practical design scenarios. The optimized airfoil demonstrated an overall performance improvement across the range of target Mach numbers, with a maximum drag reduction of 37.38% at Mach number 0.74.

2 Methodology

2.1 Training data preparation

2.1.1 Numerical simulation

To obtain the transonic flow field with randomly defined shapes, Mach numbers, and angles of attack (AoAs) for model training, steady-state CFD simulations were conducted. The Stanford University multi-block (SUmb) in-house code was employed to solve the steady-state Euler equations. A second-order centered differencing scheme was used for spatial discretization, and the convective terms were treated using the JST (Jameson-Schmidt-Turkel) method. The RAE 2822 was selected as the baseline airfoil, and for shape parameterization and variations, the Hicks-Henne bump function [31], as shown in Eqs. (1)-(2), was applied.

$$b(x) = \sin\left[\pi x^{\log\left(\frac{1}{2}\right)/\log(t_1)}\right]^{t_2}, \tag{1}$$

$$y = y_0 + \sum_{i}^{n} \mu_i b(x_i), \tag{2}$$

where y_0 represents the surface coordinates of the baseline airfoil, and μ_i denotes the weighting factors for the design variables. x_i corresponds to the maximum displacement points of the bump function, with a total of 13 points pre-specified on the surface and shown in Fig. 1. A C-type mesh with the size of 351×51 was employed and shown in Fig. 1. The Mach number was set within a transonic regime between 0.6 and 0.8 to include flow nonlinearity of shock waves. The AoAs varied between 1° and 3°. Shape parameters and flow conditions were varied by the Latin Hypercube Sampling

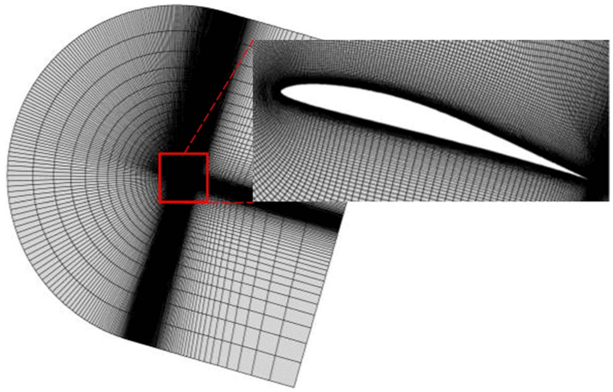

Fig. 2 C-type mesh of Eppler 387 airfoil

techniques. A database consisting of a total of 2,000 cases of steady-state transonic flow field was constructed, among which 1665 cases were used as training data, while 185 cases were used as validation data for the model to prevent overfitting issues. Finally, the performance of the trained model was tested using 150 test data.

For the prediction of the steady-state flow field over the Onera M6 wing under transonic conditions, the flow field over the wing was computed. The Reynolds-Averaged Navier–Stokes (RANS) equations were solved using the Spalart–Allmaras(SA) turbulence model. Flow conditions were determined using Latin Hypercube Sampling techniques, setting 253 values between Mach 0.8 and 1.2. Out of the total 253 data points, 225 were used as training data, 25 as validation data, and the remaining 3 as test data.

For the prediction of unsteady and highly turbulent flows, airfoil flows at high angle of attack were calculated. The RANS equations were solved employing the SA turbulence model. Spatial discretization was carried out by Roe's 3rd order flux splitting method, whereas the time integration of the solver employed the diagonalized alternating direction implicit (DADI) method. The Eppler 387 airfoil was selected, and a C-type structured mesh of size 689×145, as depicted in Fig. 2, was constructed. The unsteady characteristic is mostly induced by the high angle of attack at 16° at Reynolds number of 1.0×10^{5}. The CFD calculations were carried out over a time period of 25 using a non-dimensional time step of 0.1, resulting in a total of 250 solution snapshot data. To ensure convergence, transient flows up to 11.0 were discarded, and only 140 solution snapshot data from 11.1 to 25.0 were used as training data. Among them, 98 time steps were used as training data, and the remaining 42 time steps were used as test data.

2.1.2 Data pre-processing

In this study, CFD solution data obtained from the FOM calculation were directly used for model training without any modifications in the grid topology in order to avoid resolution degradation potentially caused by interpolation into the additional coordinate system. To make the original flow data applicable to the CNN-based model, a mapping was performed to transform the array structure as follows. The computational grid topology was initially laid out in a structured manner using an index system (i, j), where the circumferential direction corresponds to the i direction and the normal direction to the surface corresponds to the j direction. Now, all i coordinates corresponding to a fixed j at the original grid points are collected and rearranged into a one-dimensional vector. This process is performed sequentially until reaching $j = max$, ultimately generating a vector composed of the entire flow field data. The generated flow field vector was reshaped into a 2D matrix to match the input format of the CNN using Algorithm 1 (shown in Fig. 3), which also removed unnecessary far-field regions. While a sufficient far-field region is typically required for accurate FOM CFD simulations, empirically it can hinder the extraction of flow characteristics during CNN model training. As a result, the mesh sizes of RAE 2822, Onera M6 wing and Eppler 387 were reduced to 328×40, 88×16 and 648×120, respectively. This algorithm is applicable to most 2D structured meshes with minor modifications. Furthermore, since the model's output results maintain the grid topology used in the FOM calculation, it has the advantage of being able to perform post-processing tasks such as calculating lift coefficients without additional back-mapping.

Moreover, there exist scale differences of $O\left(10^1\right)$ to $O(10^2)$ between various physical quantities (u, v, p) used for training, with a particularly significant scale difference of $O\left(10^3\right)$ to $O(10^4)$ for the pressure field between the airfoil surface and the far-field. In such cases, during the training process, the model may become biased towards large values in specific data or get trapped in local minima. To address this issue, the training data was subjected to a maximum-minimum normalization within a specified range.

2.2 Conditional Unet

2.2.1 Model structure

The CNN [32] has been applied in computational fluid dynamics [33, 34], as it can process flow field data by image recognition and pattern recognition. Unlike traditional artificial neural networks (ANNs), the CNN addresses the issue of performance degradation caused by the loss of spatial information of the data by utilizing filters. The CNN is composed of convolutional and sampling layers, and the operation of the convolutional layer can be described by Eq. (3).

$$s_{ijm} = \sum_{k=0}^{K-1} \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} z_{i+p,j+q,k} W_{pqkm} + b_m. \tag{3}$$

$z_{i,j,k}$ represents the input value at point (i, j, k), while W_{pqkm} denotes the filter weights at point (p, q, k), and b_m represents the bias of the m-th filter. s_{ijm} represents the output of the convolutional layer. Figure 4a and b provides a schematic representation of the operations performed in the convolutional layer. The input is a three-dimensional matrix of size $M \times N \times K$, representing height, width, and the number of channels, respectively. As the input passes through each convolutional layer, features are extracted by filters with a size of $H \times H$, followed by non-linear activation function operations. In this study, the ELU (Exponential Linear Unit) function, as shown in Eq. (4), was employed. The weight α was set to 1.

$$f(x) = \begin{cases} x & x \geq 0; \\ \alpha(e^x - 1), & x < 0. \end{cases} \tag{4}$$

Algorithm 1 Flow field pre-processing

Input:

vector form of flow field, F

total number of indices in the i direction, i_t

total number of indices in the j direction, j_t

number of i indexes to remove, i_{slice}

number of j indexes to save, j_{save}

take indexes within the range, $\textsc{arrange}(min, max)$

rearrange the dimensions of array, $\textsc{reshape}$(array,[#row, #column])

Output:

pre-processed flow field matrix F_p

$index_{slice} \leftarrow$ create empty matrix

for $k = 1, 2, \ldots, j_{save}$ **do**

  $index \leftarrow \textsc{arrange}(i_{slice} + ki_t, (i_t - i_{slice}) + ki_t)$

  $index_{slice} \leftarrow index_{slice} \cup index$

end for

$F \leftarrow F[index_{slice}]$

$F_p \leftarrow \textsc{reshape}(F, [(i_t - 2i_{slice}), j_{save}])$

Fig. 3 Flow field pre-processing algorithm

The sampling layer performs compression or expansion operations on the output of the convolutional layer. As shown in Fig. 4c, the Max pooling process can eliminate unnecessary information and reduce the amount of information. Conversely, in the decoding process, as shown in Fig. 4d, up-sampling layer interpolates adjacent low-dimensional information to high-dimensional information. Each convolutional layer consists of linear operations, but due to the non-linear activation functions and multiple layers of hidden structures, CNN excels in learning complex non-linear data.

In the present study, two types of Conditional Unet (CUnet) models, based on the CNN architecture, are proposed to predict both steady and unsteady flows with high accuracy. The overall structure of Unet is similar to that of a Convolutional Autoencoder, but it incorporates skip connections between the encoder and the decoder. This allows the Unet to utilize not only the low-dimensional information from previous layers during the decoding process but also the high-dimensional information from corresponding encoder layers, enabling enhanced data reconstruction with improved resolution. Figure 5 represents the CUnet model for predicting steady flows, which considers three parameters: shape, Mach number, and angle of attack simultaneously. It consists of three hidden layers, with an additional structure in the bottleneck layer to incorporate user-defined flow conditions into the encoded latent vector.

The operational process of the model is as follows: pre-processed grid coordinates (x, y) of size 328×40 are used as inputs, which pass through the encoder and are represented as a latent vector of size 64. Then, two flow conditions are concatenated with the latent vector, and a fully-connected layer is applied for the appropriate blending of information. The modified latent vector is decoded to output velocity field (u, v) and pressure field (p) that satisfy the specified geometry and flow conditions. The model is trained with a batch size of 32, a learning rate of 10^{-3}, and a total of 10,000 training iterations.

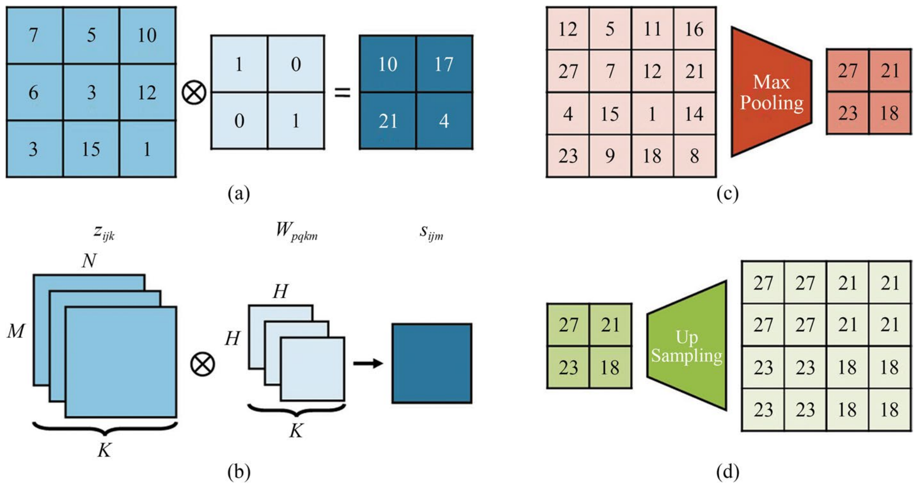

Fig. 4 Operations in the convolutional layer and the sampling layer [27]

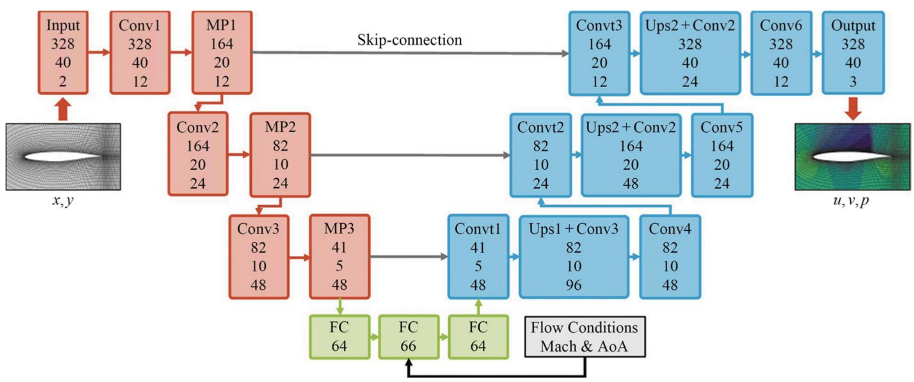

Fig. 5 Structure of conditional Unet for steady flow prediction with varying shapes and flow conditions of 2D airfoil. The model consists of convolutional, transposed convolutional, max pooling, up-sampling, and fully-connected layers, and the value in the blocks is the output of each operation

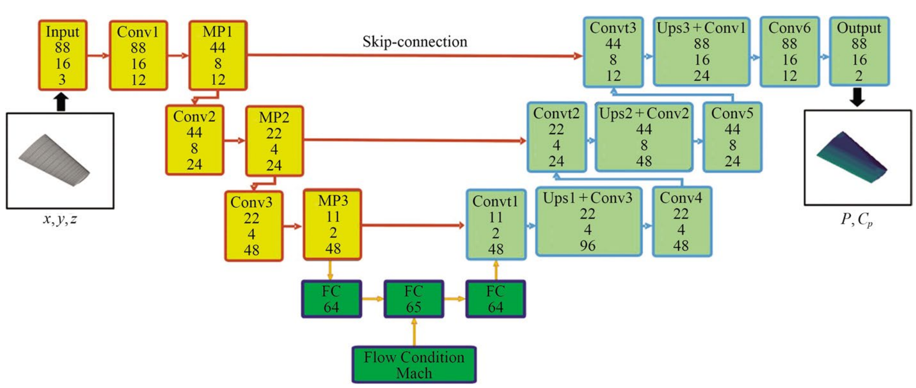

Fig. 6 Structure of conditional Unet for steady flow prediction with Onera M6 wing

The architecture of the CUnet for predicting the flow field of a three-dimensional shape in steady-state flow is illustrated in Fig. 6. The three-dimensional shape used for training the model is the Onera M6 wing. The input to the model consists of

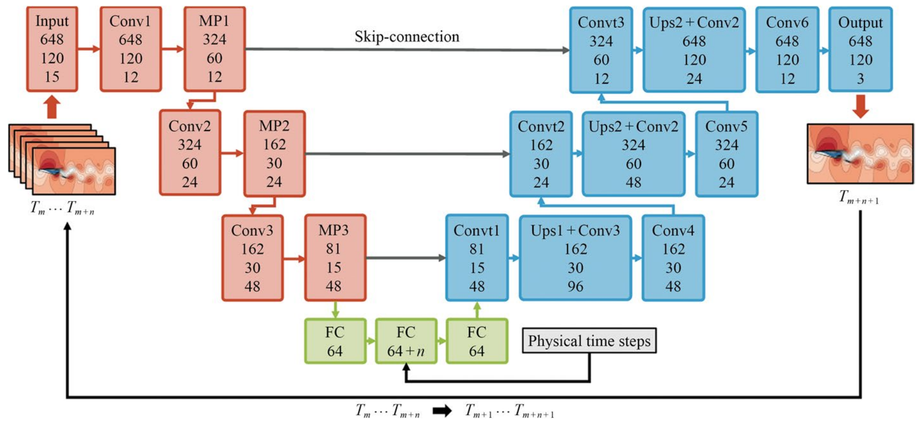

Fig. 7 Structure of conditional Unet for unsteady flow prediction

pre-processed grid coordinates (x, y, z) of size 88×16, which are then represented as a latent vector of size 64 through the encoder. Flow conditions are combined with the encoded latent vector in the latent space. This combined latent vector is decoded to output a pressure field corresponding to specific flow conditions. The data for training and testing the model were taken from transonic flow conditions with Mach numbers ranging from 0.8 to 1.2. Out of a total of 253 data points, 25 were used for validation, 225 for training, and 3 for testing. The model underwent 15,000 training iterations, and the optimal batch size and learning rate of 250 and 10^{-2}, respectively, were determined through grid search.

The structure of the CUnet for predicting unsteady flows is slightly different and is illustrated in Fig. 7. To train time series data, we transformed the training data into a windowed dataset with a pre-specified horizon length and train the model to predict flow field $q(T_{m+n+1})$ at time T_{m+n+1} with input flow field data of $q(T_m) \ldots q(T_{m+n})$ corresponding to times from T_m to T_{m+n}, where T_m can be an arbitrary time and n is set to 5. The size of input snapshot, $n = 5$, was selected based on offline tests by varying the values from 3 to 7 as the model achieved the highest accuracy when using 5 snapshots. The predicted flow field $q(T_{m+n+1})$ is reused as an input for the prediction at the next time step, and the input at that time includes $q(T_{m+1}) \ldots q(T_{m+n+1})$. This iterative process allows the model to predict flow fields in future time steps. To input multiple time steps of flow fields into the encoder simultaneously, they were stacked as multiple channels, resulting in a final input data size of $648 \times 120 \times 15$, including three types of flow fields. The rest of the model architecture remains the same as the CUnet for predicting steady flows. The model was trained with a batch size of 4, a learning rate of 10^{-3}, and a total of 5,000 training iterations.

In addition, for the enhancement of the model, batch normalization [35] was applied between the convolutional layers and activation layers in each hidden layer. Even if normalization is performed at the initial input, the internal covariate shift occurs during the training process as the parameters are updated, leading to a degradation in the effectiveness of the activation functions. Batch normalization helps suppress such phenomena, reducing sensitivity to initial parameters and accelerating the learning process. Moreover, the normalization process acts as a form of noise, thereby preventing overfitting without the need for an additional dropout layer [36]. Additionally, a learning rate scheduler module was implemented to finely explore the vicinity of optimal parameters as the training progresses. In this case, an exponential decay step of 200 and a decay rate of 0.9 were set.

2.2.2 Customized loss function

Two different loss functions were used for model training. To optimize the airfoil shape using CUnet, it is crucial to accurately predict the drag coefficient, which serves as the objective function. Therefore, an additional loss function was defined for the surface flow field that is used in drag coefficient computation. This has the effect of guiding the model to prioritize the learning of the surface flow field.

$$\varepsilon_f = \frac{1}{N}\sum_{i=1}^{N}(q_{t,i} - q_{p,i})^2. \tag{5}$$

$$\varepsilon_{SL} = \frac{1}{M}\sum_{i=1}^{M}(q_{t,i} - q_{p,i})^2. \tag{6}$$

$$\varepsilon_s = w_1\varepsilon_f + w_2\varepsilon_{SL}. \tag{7}$$

Equation (5) represents the Mean Squared Error (MSE) between the ground truth and the predicted values for the entire flow field, while Eq. (6) represents the MSE specifically for the surface flow field. The subscripts t and p denote the ground truth and the predicted values, respectively. These two equations were combined using weighted summation to form the loss function, where the weights w_1 and w_2 were set to 0.3 and 0.7, respectively.

The second loss function incorporates the Gradient Difference Loss (GDL) [37] into the SL. The mathematical expression for GDL is given by Eq. (8).

$$\begin{aligned}\varepsilon_{GDL} =& \frac{1}{N_x}\frac{1}{N_y}\frac{1}{N_\phi}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{k=1}^{N_\phi}\Big[\big|\big(q_{t(i+1,j,k)} - q_{t(i,j,k)}\big) - \big(q_{p(i+1,j,k)} - q_{p(i,j,k)}\big)\big| \\ &+ \big|\big(q_{t(i,j+1,k)} - q_{t(i,j,k)}\big) - \big(q_{p(i,j+1,k)} - q_{p(i,j,k)}\big)\big|\Big].\end{aligned} \tag{8}$$

$$\varepsilon_s = w_1\varepsilon_f + w_2\varepsilon_{SL} + w_3\varepsilon_{GDL}. \tag{9}$$

GDL, as shown in Fig. 8, has the characteristic of incorporating gradient errors between grid points into the evaluation of the loss function. This allows the model to learn the correlations between adjacent grid points, improving convergence of the model and prediction accuracy. The weights used in Eq. (9) are set to 0.3, 0.3, and 0.4, respectively.

2.3 Machine learning with POD methods

2.3.1 Proper orthogonal decomposition

The POD technique is a widely used linear dimensionality reduction method for creating ROMs based on high-fidelity snapshot data obtained from FOM analysis, including CFD calculations [7, 8]. By subtracting the mean component $\overline{q}(p)$ from the snapshot matrix composed of flow field data, the correlation matrix can be constructed as shown in Eq. (10).

$$\boldsymbol{F}(\mu_i) = \begin{bmatrix} q(p_1, \mu_i) - \overline{q}(p_1) \\ \vdots \\ q(p_n, \mu_i) - \overline{q}(p_n) \end{bmatrix}. \tag{10}$$

$$\boldsymbol{F}^T \boldsymbol{F} v_i = {\lambda_i}^2 v_i. \tag{11}$$

In Eq. (10), $q(p, \mu)$ represents a vector field at a specific grid point p, time or shape parameter μ, as independent variables. By performing a singular value decomposition on the matrix of $\boldsymbol{F}^T \boldsymbol{F}$, the eigenvalue λ and eigenvector v of the correlation matrix can be computed. The left singular vector u and the corresponding modal coefficient a are obtained. As the first few dominant eigen pairs of the eigenvalues ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$) and basis vectors alone can typically capture over 90% of the data information, the flow field can be re-expressed as a linear combination, as shown in Eq. (12).

$$\begin{bmatrix} q(p_1, \mu_i) \\ \vdots \\ q(p_n, \mu_i) \end{bmatrix} \approx \begin{bmatrix} \overline{q}(p_1) \\ \vdots \\ \overline{q}(p_n) \end{bmatrix} + \sum_{j=1}^{d} a_j(\mu_i) u_j. \tag{12}$$

Since each flow field snapshot is represented by the modal coefficients a, if the values of a corresponding to μ are known, the entire flow field can be reconstructed using Eq. (12), resulting in significant cost savings.

In this study, the POD-based ROM was constructed for comparison purposes with the Conditional Unet-based ROM when shape and time parameters are varied arbitrarily in steady and unsteady flow fields, respectively. To determine the appropriate number of modal coefficients, the energy or power spectrum of the basis vectors was analyzed as shown in Fig. 9 for the simulation data mentioned in Section 2.1.1. It can be observed that the initial modes capture a significant portion of the overall flow field information. The number of POD modes used in each model configuration will be specified in the following sections.

2.3.2 Multi-variate Gaussian Process Regression for mode interpolation

For the modal coefficients that are not included in the training data but are needed for the prediction at unseen conditions, either interpolation or extrapolation needs to be calculated for accurate reconstruction of the flow fields. Whereas the flow at future time steps is always on the extrapolation, those of unknown shape or at unseen flow conditions can be predicted by the interpolation or the regression out of the modal coefficients.

Gaussian Process Regression (GPR) is a supervised learning method popularly used in multi-dimensional interpolation, often called Kriging [38], named after a mathematician

Krige, and it is defined by mean and covariance function of training data [39]. The GPR is expected to select a function, out of all possible candidate functions, that best represents the exact response of the unseen input parameters.

In the present study, a Multi-Variate Gaussian Process Regression (MGPR) model is employed to calculate a set of mode coefficients at unseen test conditions, whose mathematical form is shown in Eq. (14).

$$f^T = \left[f(\mu_1)^T, \ldots, f(\mu_m)^T\right] \sim MGP(m(\mu), k(\mu, \mu\prime)). \tag{13}$$

$$MGP(\mu; m, k) = \frac{1}{(2\pi)^{d/2}} |k|^{-1/2} \exp\left[-\frac{1}{2}(\mu - m)^T k^{-1}(\mu - m)\right]. \tag{14}$$

$$k\left(\mu, \mu'\right) = \sigma^2 \exp\left[-\frac{(\mu - \mu')^T(\mu - \mu')}{2l^2}\right]. \tag{15}$$

Where $m(\mu)$ and $m\left(\mu, \mu'\right)$ in Eq. (13) represent the mean and covariance of the input training data, respectively. The kernel function between two data points is formulated by the squared exponential kernel. The values of σ and l are determined by the maximum likelihood estimation among function candidates generated by Eqs. (13)-(15). To train the GPR model, a total of 15 variables, i.e., 13 shape parameters and 2 flow conditions, were used as inputs, while the corresponding POD modal coefficients were used as outputs. In this case, 580 modal coefficients were employed, which corresponded to an energy level of 99.9%. The constructed POD-GPR model is compared with the proposed CUnet for predicting transonic steady flow.

2.3.3 Long Short-Term Memory network for time series prediction

For time-series data and flow prediction at future time, the Long Short-Term Memory (LSTM) network is effective as a kind of Recurrent Neural Network (RNN) specialized in learning from sequential data, such as in natural language processing, and was proposed by Hochreiter and Schmidhuber in 1997 [40, 41]. Each LSTM module consists of neural networks called the cell state (C_t), forget gate (f_t), input gate (i_t), and output gate (o_t). The computations taking place within each neural network are as follows.

$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right), \tag{16}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i), \tag{17}$$

$$\widetilde{C}_t = \tanh(W_C.[h_{t-1}, x_t] + b_C), \tag{18}$$

$$C_t = f_t.C_{t-1} + i_t.\widetilde{C}_t, \tag{19}$$

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o), \tag{20}$$

$$h_t = o_t.\tanh(C_t), \tag{21}$$

where h_t is the hidden state, which is an output of each module, and x_t is the cell input. W and b denote the weights and biases of each gate, respectively, and σ represents the sigmoid function. Based on this structure, LSTM is capable of processing time series problems by selectively combining or discarding old and new information according to the optimized weights.

In this study, LSTM was employed to predict the unknown flow fields of the future time steps based on the past time steps of an unsteady flow field. The training data consisted of the same windowed data described in Section 2.2.1, and the mode coefficients extracted from the POD analysis were used to train the LSTM. According to a prior study [21] using POD-LSTM, adopting an excessive number of mode coefficients during LSTM training can have a negative impact on a model performance. The less dominant POD modes with lower energy contributions exhibit highly oscillatory characteristics, which introduce noise into the prediction and degrade model performance. This phenomenon occurred in the same way in the results of this study, as shown in Fig. 10. Therefore, the final model was constructed using only 9 mode coefficients. The LSTM layer consisted of two layers, and the hidden size was set to 20. A batch size of 1 was used, and the model was trained for a total of 5000 iterations with a learning rate of 10^{-4}.

3 Results and discussion

3.1 Steady transonic flows with varying shapes and conditions of 2D airfoil

For 150 randomly configured test cases with varying shape, Mach number, and angle of attack as the parameters, the CUnet was used to predict the velocity components (u, v) and pressure (p) at transonic Mach numbers that were not tested for training. Models were generated for the two loss functions SL and GDL, respectively. These results were compared with those obtained from the POD-GPR method using the same test data. Figure 11 shows the predicted pressure fields for three randomly selected cases, and the contour plots of the absolute errors between the FOM solutions and the prediction results are presented in Fig. 12. Table 1 shows the velocity and pressure prediction errors in percentage.

As a result of the flow field prediction, CUnet with GDL exhibited the highest level of agreement with the FOM results. The average error for the entire set of test data, based on the pressure field, was 1.08%, with a minimum error of 0.33% and a maximum error of 3.04%. Moreover, it accurately predicted the intensity and location of shock waves, which are highly sensitive to the airfoil shape and flow conditions. When SL was used as the loss function, CUnet also demonstrated a small error of 1.46% and accurately predicted the shock waves. However, the contour lines appeared unstable as if the solution had not converged, and a striped pattern of errors around the airfoil was observed, as shown in Fig. 12a. Therefore, the absence of these phenomena in the model with GDL suggests that training the AI model to learn the gradients between adjacent data points is effective in improving accuracy in flow field prediction. Lastly, POD-GPR exhibited the poorest prediction results. Although it performed well for the first case, which had minimal shock wave occurrence, it showed reduced prediction accuracy for cases with strong shock waves, as seen in the third case. This can be attributed to the mathematical limitations of the linear POD approach, which fails to adequately capture the discontinuities and nonlinearities in the training data. While this study focused on structured grid-based data, it is expected that future extensions to flow field prediction using Graph Neural Networks (GNN) [42] can be applied to handle unstructured grid-based data, with the computational cost to create and train the GNN being another issue as well as the solution accuracy, which the present Conditional Unet does not suffer much.

Another objective of this study is to investigate the cost efficiency of the AI-based ROM for optimal design using. The performance metrics of design (lift and drag coefficients) require precise information of surface flows, and therefore, the ROM should predict them accurately. Figure 13 shows a comparison between the extracted surface pressure distribution from the previously predicted flow fields and the corresponding FOM results. In Table 2, the accuracy of CUnet is notably superior in predicting surface velocity and pressure of steady flow compared to the POD-GPR method. The data presented in Table 2 elaborates on the relative prediction errors for the u, v, and p components, substantiating this assertion. The relative prediction errors for u, v, and p components using the CUnet with GDL and SL models are demonstrably lower than those recorded with POD-GPR. For example, the average prediction error for the u component is 0.87% and 1.05% for CUnet with SL and GDL, respectively, which is 0.29% and 0.11% lower than the 1.16% noted for POD-GPR. This trend of reduced error margins continues with the v and p components, enhancing the accuracy and reliability of predictions.

In the prediction of surface pressure distribution, the CUnet with SL, which places greater emphasis on the accuracy of the surface flow field, demonstrated the highest accuracy. As shown in Fig. 13a, the results of CUnet exhibit excellent agreement with the FOM results. The CUnet with GDL also showed considerable accuracy in the prediction

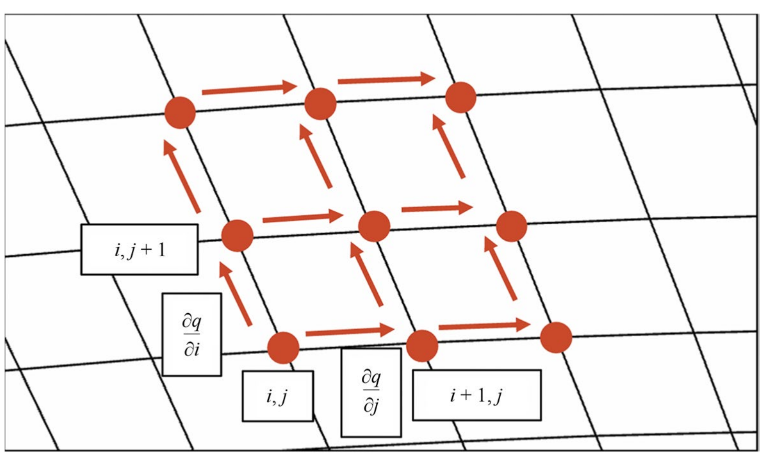

Fig. 8 Geometrical interpretation of Gradient Difference Loss

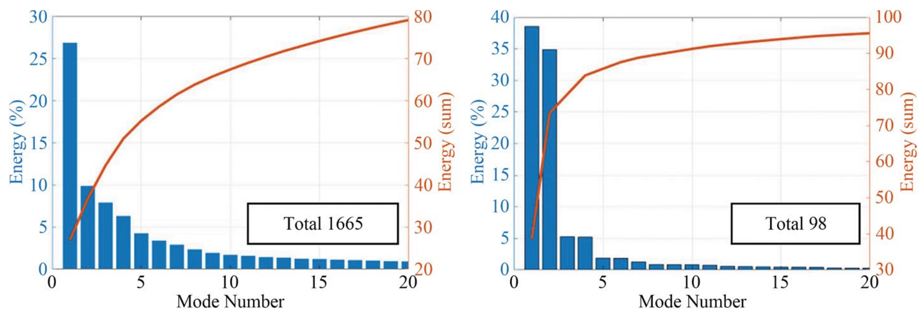

Fig. 9 The energy spectrum for a basis vector and cumulative energy level of steady flow (left) and unsteady flow (right) data

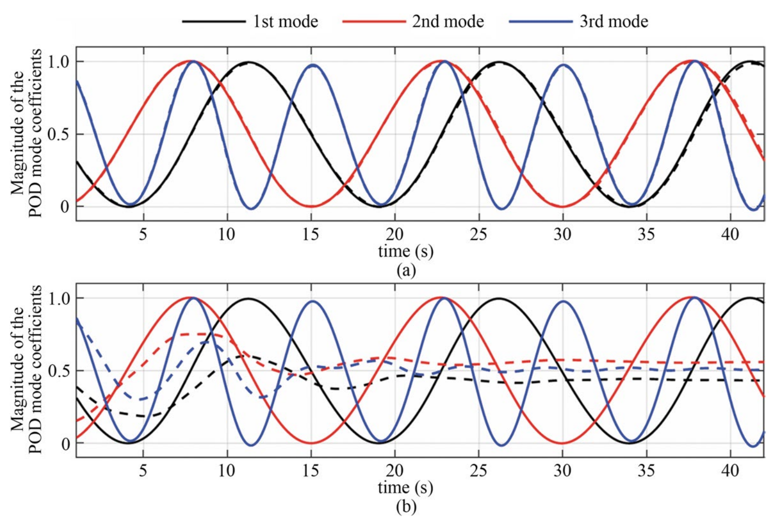

Fig. 10 Temporal mode coefficient prediction results according to the number of POD modes used for LSTM learning. Each model was trained using mode coefficients of 9 (energy level of 99%) and 98 (100%)

but with a slight increase in error due to the reduced weight onto the surface loss component. Similar to previous findings, the POD-GPR performed well in predicting the pressure distribution outside the shock wave region but showed limitations in accurately predicting the location and the strength of shock waves. Consequently, the proposed CUnet in this study accurately predicts flow fields with a wide parameter space, such as shape and flow conditions, while incorporating discontinuous physical phenomena like shock waves.

3.2 Steady transonic flows with varying conditions of Onera M6 wing

From a dataset of 253 data cases, composed of randomly sampled flow conditions at transonic Mach numbers from 0.8 to 1.2 using Latin Hypercube Sampling on the same geometry, three data cases were used as test cases. The CUnet was employed to predict the pressure components (p) and pressure coefficients at untrained Mach numbers of 0.83, 0.95, and 1.17. These results are visible in Fig. 14a, while Fig. 14b calculates the relative prediction errors between the pressure fields obtained from CUnet and those from the FOM. Figure 15 compares the distributions of pressure coefficients at Mach

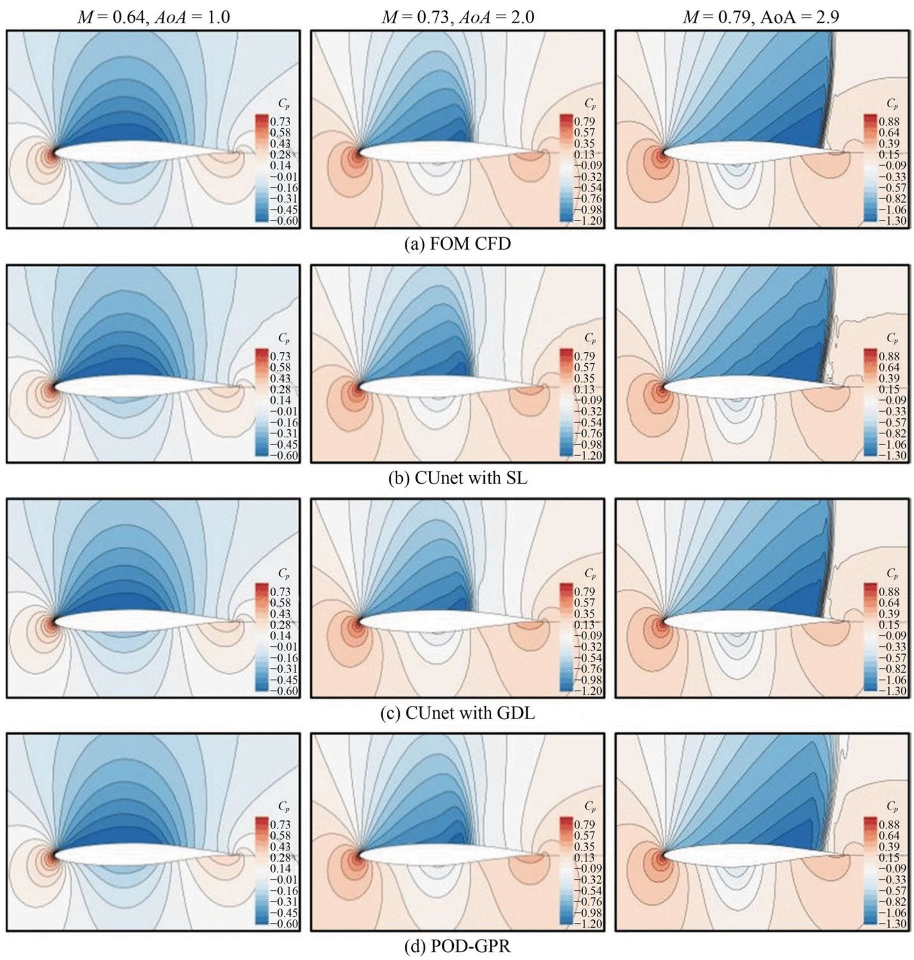

Fig. 11 Pressure fields predicted by CUnet with SL, CUnet with GDL, and POD-GPR at different Mach numbers and angles of attack

0.95, where the largest relative errors were observed in Fig. 14b, between FOM and CUnet. As shown in Fig. 15, the differences in pressure coefficient distributions between CUnet and FOM are minimal, demonstrating that CUnet is an applicable technique for 3D geometries.

3.3 Unsteady flows with high angle of attack

Unsteady flows for unseen time steps over a high AoA airfoil were predicted using the CUnet. While unsteady turbulent flows of wind-turbine flows were solved using the conditional Convolutional Autoencoder method [43], we only demonstrated the airfoil flows in the present study. Out of the total 140 time steps, 98 steps were used for model training, and the remaining 42 time steps were used as test data. The GDL, which is advantageous for predicting the overall flow field, was employed as the loss function. The proposed model was compared to the POD-LSTM to validate its performance. Figure 16 illustrates the velocity field predictions by two ROMs and the corresponding results of FOM at three time steps. Figure 17 compares the velocity variations

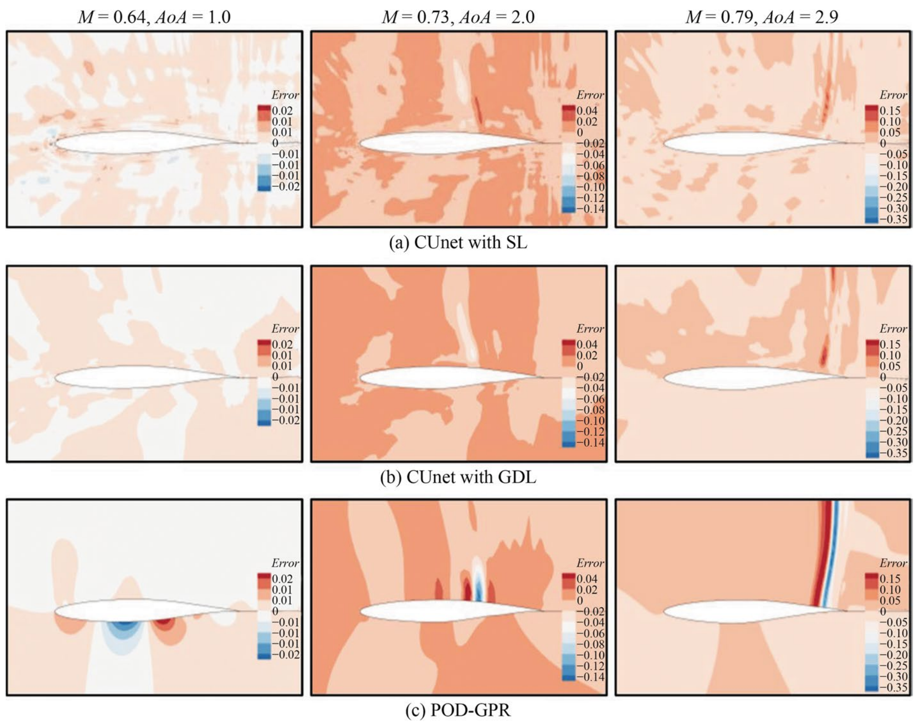

Fig. 12 Absolute errors of pressure fields predicted by CUnet with SL, CUnet with GDL, and POD-GPR at different Mach numbers and angles of attack

Table 1 Relative prediction error of velocity and pressure in the entire domain of steady flow (from Figs. 11 and 12)

	u **(%)**			***v*** **(%)**			***p*** **(%)**		
	Avg.	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**
CUnet with SL	2.04	1.34	6.55	1.57	1.06	3.25	1.46	0.86	4.58
CUnet with GDL	1.51	0.56	5.72	1.01	0.45	2.25	1.08	0.33	3.04
POD-GPR	0.55	0.04	1.93	1.27	0.28	4.44	3.98	0.31	13.76

throughout the entire tested time steps at three points located in the downstream region for more detailed analysis. Figure 18 compares the pressure coefficient distribution plots between FOM and ROM models at three different time steps. Figure 19 represents the contour plots of the absolute errors between the FOM results and the ROM predictions. Table 3 presents the relative prediction errors for the u, v, and p components for both the POD-LSTM and CUnet with GDL models, allowing for a clear comparison of the latter's advantages. The average prediction error for the u component is nearly identical between the two models, with POD-LSTM at 0.41% and CUnet with GDL at 0.42%, demonstrating a marginal difference of just 0.01%. However, in terms of minimum error, CUnet with GDL shows superior performance at 0.06%, which is 0.18% lower than 0.24% for POD-LSTM.

The contour plots of the flow field in Fig. 16b demonstrate that CUnet accurately captures the periodicity and shape of the wake flow over time. Furthermore, it accurately

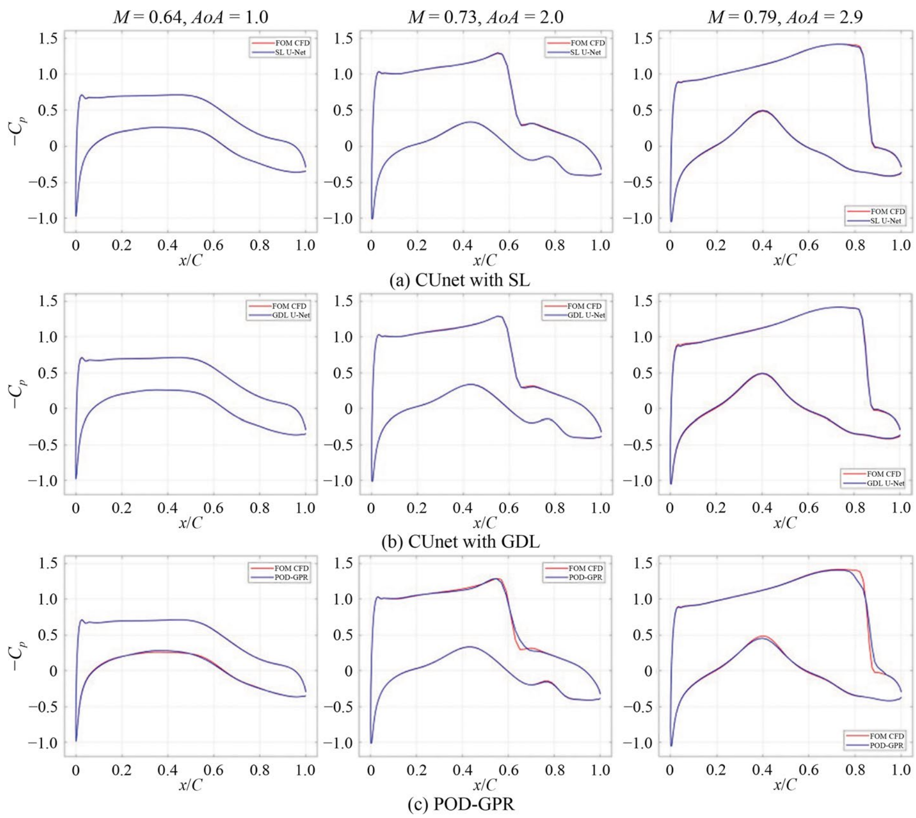

Fig. 13 Surface pressure coefficient distribution predicted by CUnet with SL, CUnet with GDL, and POD-GPR at different Mach numbers and angles of attack

Table 2 Relative prediction error of surface velocity and pressure of steady flow (from Fig. 13)

	u **(%)**			***v*** **(%)**			***p*** **(%)**		
	Avg.	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**
CUnet with SL	0.87	0.29	3.83	0.40	0.21	1.18	0.69	0.21	2.43
CUnet with GDL	1.05	0.35	4.21	0.59	0.31	1.46	0.87	0.26	3.70
POD-GPR	1.16	0.12	4.11	1.13	0.40	3.20	3.73	0.42	12.67

predicts the vortices formed by flow separation on the airfoil surface. However, it is important to note that despite the model's proficiency in handling complex flow dynamics, the U-net architecture inherently accumulates errors over sequential predictions. This is particularly noticeable in Fig. 18 at higher time steps where the dependency on prior outputs introduces slight deviations in accuracy. As evident in Fig. 19, there is a slight accumulation of errors over time due to the nature of using the output of the previous time step as the input for the next step. As shown in Fig. 20, over time, the Unet exhibits a relatively greater accumulation of errors compared to the LSTM model. The relative error of the figure was calculated according to Eq. (22), where N represents the

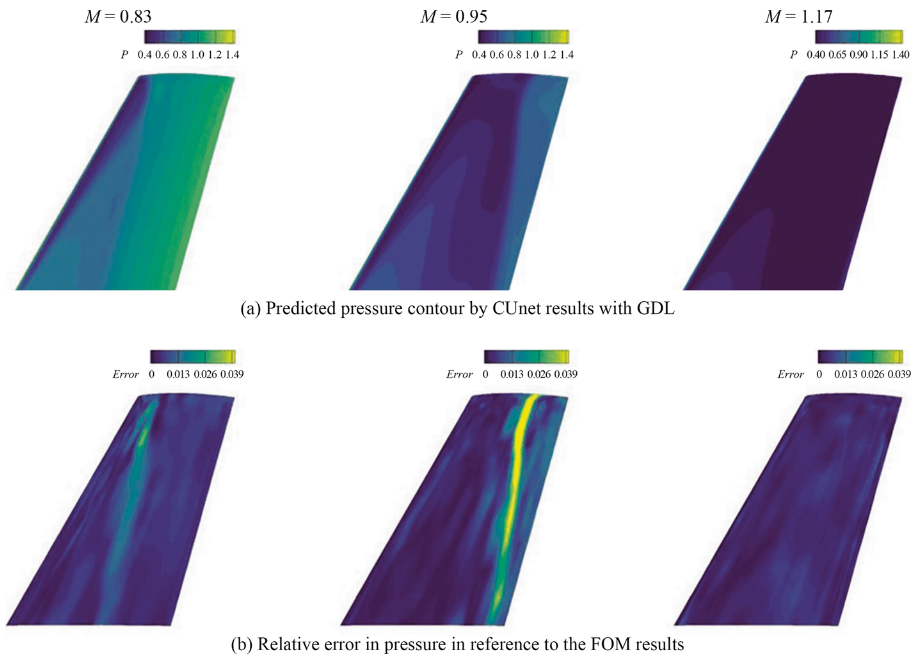

Fig. 14 Pressure contour and its relative error of the wing surface predicted by CUnet with GDL

total size of the grid. This issue can be mitigated by incorporating an Attention mechanism [44] into the skip connections of Unet. Comparing the results of CUnet with the POD-LSTM model specialized in time series prediction, no significant differences are observed. These results are meaningful in showing that unsteady flow prediction is possible only with a CNN-based model without using an additional RNN-based model. The advantages lie in not requiring additional dimensionality reduction models for flow field learning and enabling faster predictions compared to RNN-based models due to their parallelizability. In this study, the prediction focused on the relatively simple vortex-shedding flow around an airfoil. However, future research should explore the application of the proposed CNN-based method to more complex periodic flows involving rotation or pitching motions. Also, similar to previous studies, input data computed by the FOM solver is used for the initial flow prediction. However, the steady CUnet architecture demonstrated in Section 3.1 can be utilized as a pre-conditional model for generating initial input data, which holds the potential to overcome the limitation of relying on FOM solver outputs for initial flow prediction.

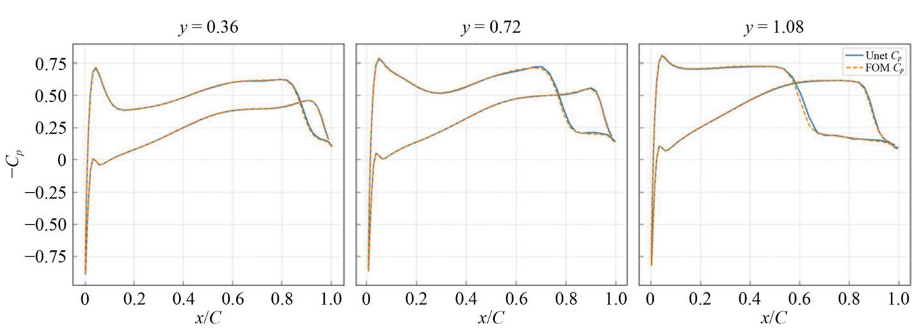

Fig. 15 Comparison of the C_p plot at three span-wise locations on the wing surface at Mach 0.95

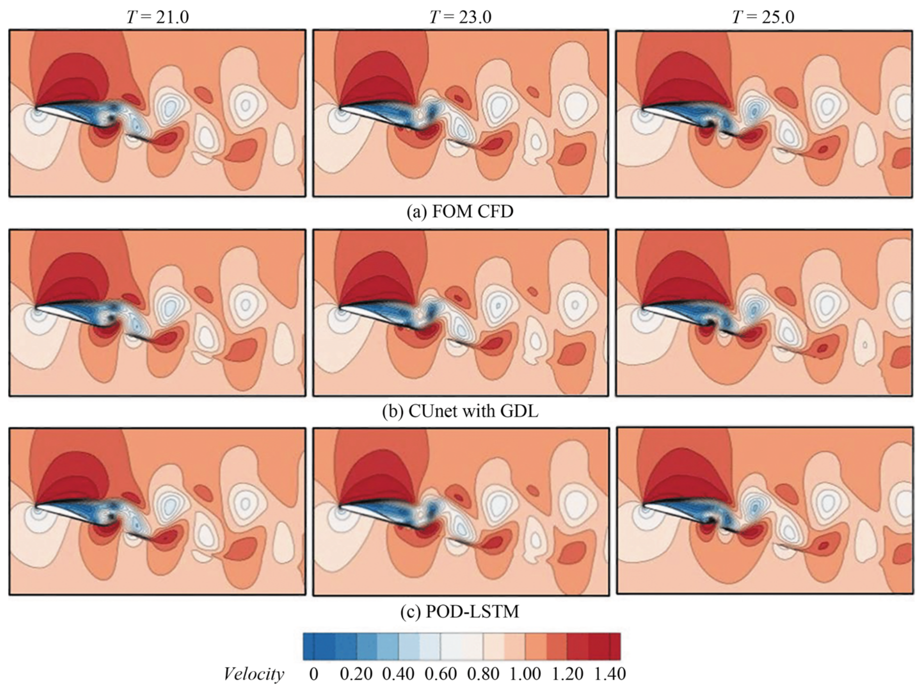

Fig. 16 Velocity fields predicted by CUnet with GDL and POD-LSTM at different time steps

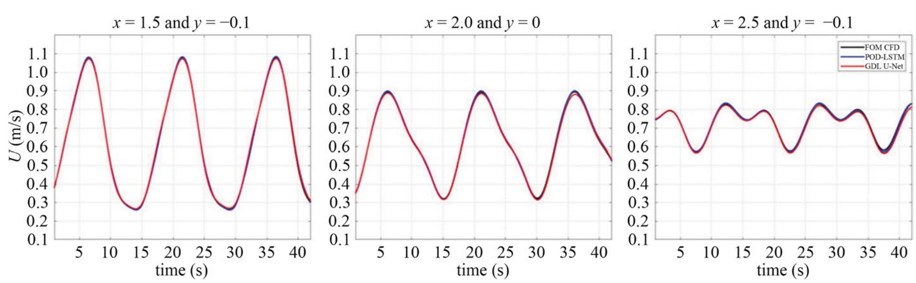

Fig. 17 Prediction of *u* velocity according to time steps at three different points in the wake area

$$Mean\ Relative\ Error = \frac{1}{N}\left(\sum_{j=1}^{j_{max}}\sum_{i=1}^{i_{max}}\left|\frac{P_{FOM_{(i,j)}} - P_{Unet_{(i,j)}}}{P_{FOM_{(i,j)}}}\right|\right). \tag{22}$$

3.4 Comparison of cost efficiency

Table 4 shows the total computation time required for predicting steady and unsteady flows using ROMs compared to FOM CFD calculations. The simulations were performed on a computer with an Intel i5-12400F CPU and an NVIDIA GeForce RTX 4090 GPU. When using the same computer architecture, the FOM solver took 280 s to compute a single case of steady flow, whereas CUnet only required 0.142 s, which is approximately 0.05% of the CFD solver. Furthermore, the computation time for unsteady flow using CUnet was 0.136 s, showing a similar level of cost reduction compared to the steady flow case. Based on these results, CUnet in this study demonstrates significantly higher

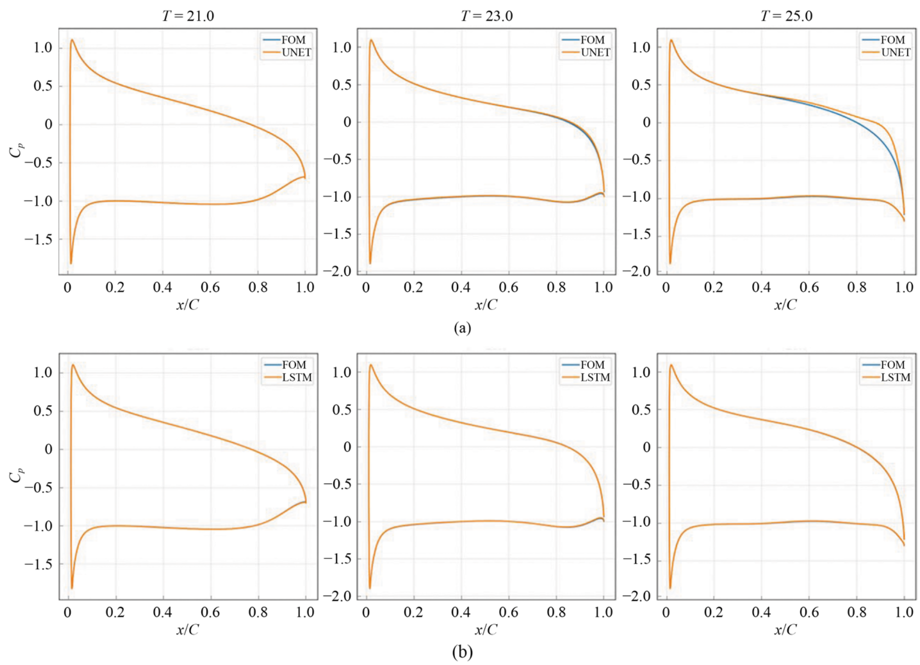

Fig. 18 Surface pressure coefficient distribution predicted by CUnet with GDL and POD-LSTM at different time steps

efficiency compared to the FOM CFD, considering the same computational resources, and enables near real-time flow prediction. With verified accuracy and efficiency, CUnet holds potential not only for optimal design, but also for future applications such as digital twin and extended reality (XR) simulations.

3.5 Design application: multi-point shape optimization of transonic airfoil

3.5.1 Definition of design problem

A multi-point design optimization of airfoil shape at transonic flow conditions is carried out using the CUnet model. The benchmark multi-point design problem, previously

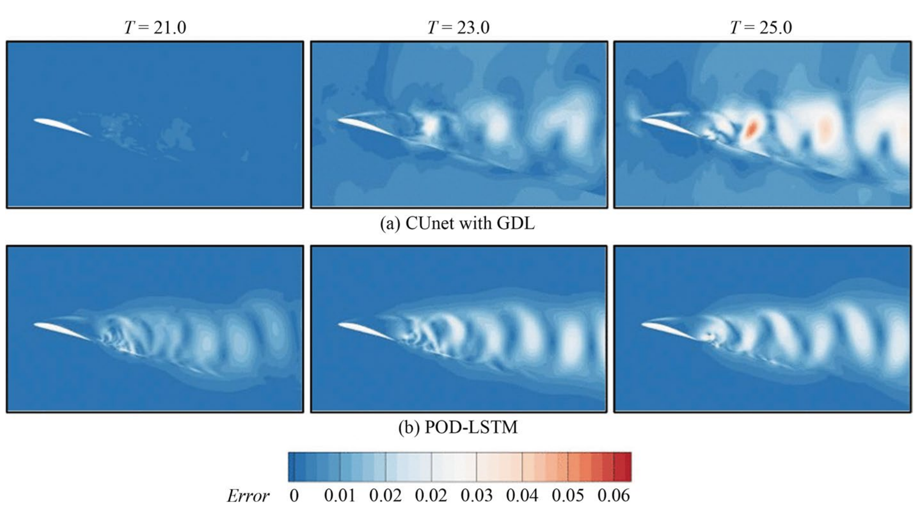

Fig. 19 Absolute errors of velocity fields predicted by CUnet with GDL and POD-LSTM at different time steps

Table 3 Relative prediction error of velocity and pressure in the entire domain of unsteady flow (from Figs. 16 and 19)

	u **(%)**			***v*** **(%)**			***p*** **(%)**		
	Avg.	**Min.**	**Max.**	**Avg.**	**Min.**	**Max.**	**Avg.**	**Min.**	**Max**
CUnet with GDL	0.42	0.06	0.92	2.40	0.24	4.34	0.18	0.02	0.48
POD-LSTM	0.41	0.24	0.74	2.28	1.25	4.27	0.13	0.07	0.28

solved by Drela [45] and Nemec et al. [46], presented challenges due to the increasing number of objective and constraint functions with the number of design points and the high cost of FOM CFD for the function evaluation. A major advantage of the present study is the dramatic reduction of computational cost by utilizing the ROM. It becomes more effective in multi-point design optimization in which the direct use of the FOM CFD calculation is nearly impossible. However, it should be noted that to best utilize the present ROM model, the objective or constraint functions that involve the entire solution field such as the inverse design of target pressure or velocity field can be used, which is left for future work.

For the optimization process, the NSGA-II algorithm [47], a gradient-free optimization method, was employed. The design points are selected at four Mach numbers of 0.68, 0.71, 0.74, and 0.76. The RAE 2822 airfoil was chosen as the baseline, and shape variations were applied using the Hicks-Henne bump function. The design variables included the magnitudes of 13 bump functions applied at predetermined locations on the airfoil surface and four AoAs for each design condition. To define the objective functions, target values for lift coefficient ($C_{L,t}$) and drag coefficient ($C_{D,t}$) were pre-set. The objective functions J_D and J_L, formulated in Eqs. (23) and (24), aimed to minimize wave drag while maintaining the lift coefficient at a desired level. The value of $C_{L,t}$ was carefully chosen within the feasible range obtainable by the CUnet, while the value of $C_{D,t}$ was deliberately set as unattainable to allow for sufficient minimization of drag coefficient.

The objective function J_D focused on drag reduction at the four design Mach numbers, with individual weights assigned as 1, 1, 2, and 3, respectively. Conversely, J_L aimed to

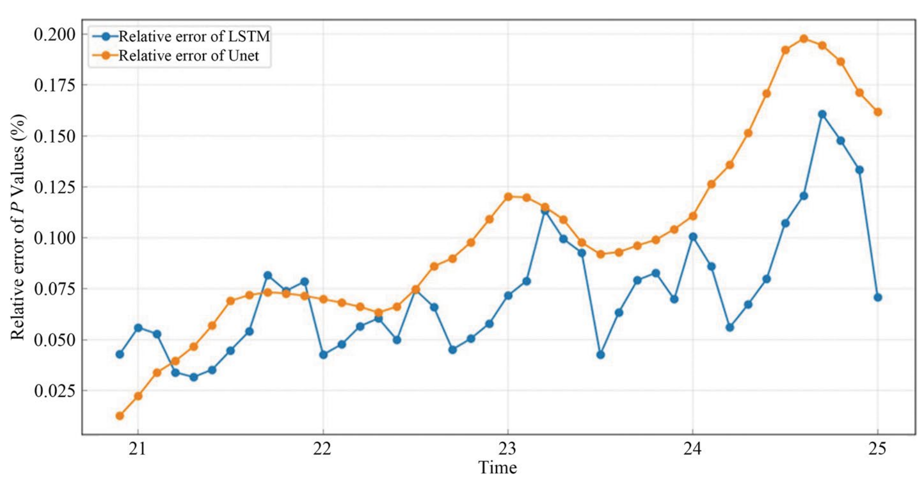

Fig. 20 Mean relative error of pressure values predicted by Unet and LSTM

Table 4 Comparison of runtime required for steady and unsteady flow prediction with FOM CFD and ROM models

Method for steady flow		Runtime (sec)	Method for unsteady flow		Runtime (sec)
FOM CFD (1 case)		280	FOM CFD (1 time step)		186
CUnet	Training	3,376	CUnet	Training	2,768
	Prediction (1 case)	**0.142**		Prediction (1 time step)	**0.136**
POD-GPR	Training	600	POD-LSTM	Training	1,412
	Prediction (1 case)	**0.127**		Prediction (1 time step)	**0.139**

maintain the lift coefficient without reducing drag, with all weights set to be identical. Finally, a combined objective function, J_m, was formulated in Eq. (25) as a weighted sum of J_D and J_L, with a scaling weight of 10^{-4} to achieve similarity in the magnitude of the two objective function values.

$$J_D = \sum_{i}^{4} w_i \left(1 - \frac{C_{D,i}}{C_{D,t}}\right)^2, \tag{23}$$

$$J_L = \sum_{i}^{4} \left(1 - \frac{C_{L,i}}{C_{L,t}}\right)^2, \tag{24}$$

$$\min J_m = w_m J_D + J_L, \tag{25}$$

$$\text{subject to} \begin{array}{rl} C_{L,t} &= 0.733, \\ C_{D,t} &= 0.003, \\ x_l &\leq 0. \end{array} \tag{26}$$

To validate the results of the multi-point design optimization, a single-point shape optimization was also independently carried out. A constrained optimization with an

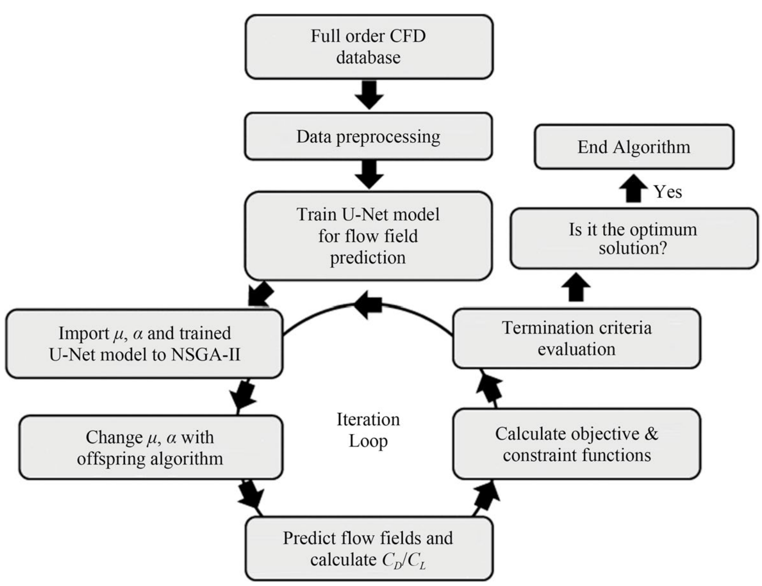

Fig. 21 Schematic diagram of the design process

Table 5 Parameters for optimization algorithm

Design iterations	Crossover prob. (%)	Mutation prob. (%)	Population no.	Offspring no.
3000	60	10	100	20

explicit form of the $C_{L,t}$ was conducted with the objective function of C_D. The target Mach number is 0.74, with the target lift coefficient of 0.733 being the same as that of the multi-point optimization.

$$\min J_s = C_D, \tag{27}$$

$$\text{subject to} \begin{array}{rl} C_{L,t} & = 0.733, \\ x_l & \leq 0. \end{array} \tag{28}$$

Figure 21 shows the schematic of the entire optimization process for the design, where the FOM CFD evaluations are replaced by the CUnet model to enable nearly real-time cost benefit. A new design candidate returned by the NSGA-II at each design iteration provides the flow solution field as well as the aerodynamic performance at the computation cost of a fraction of a second by the CUnet model. Hyper-parameters used for the optimization algorithm are summarized in Table 5. The maximum design iteration is set at 3,000, and the initial population contains 100 airfoil shapes selected randomly from the database of 2,000 candidates. During each design iteration except for the 1st generation, a total of 20 offspring of new design candidates are generated to produce the best fitness in the next generation.

3.5.2 Design results

Design results from the single- and multi-point optimizations are shown in Fig. 22, with detailed comparisons between the baseline and the optimized configurations

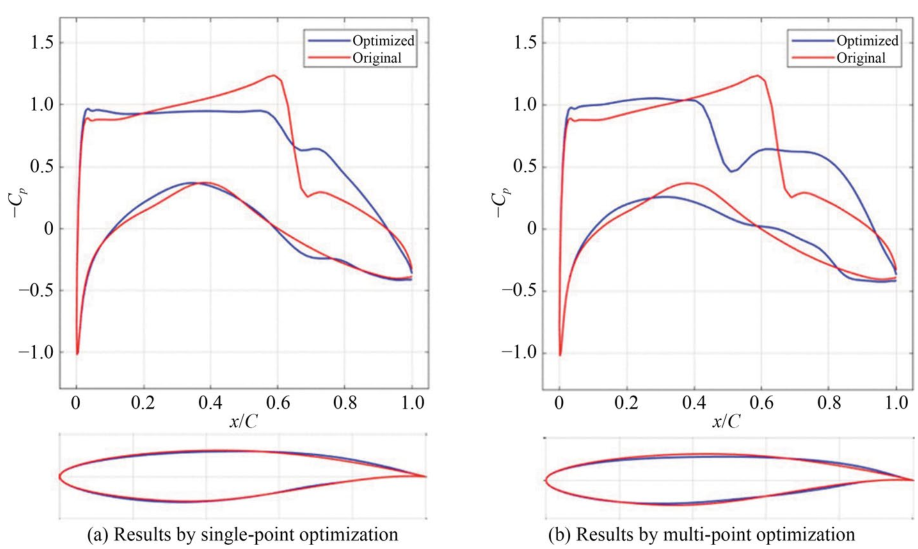

Fig. 22 Pressure coefficient distribution and airfoil shape at $M = 0.74, C_L = 0.733$ obtained by single-point optimization (left) and multi-point optimization (right)

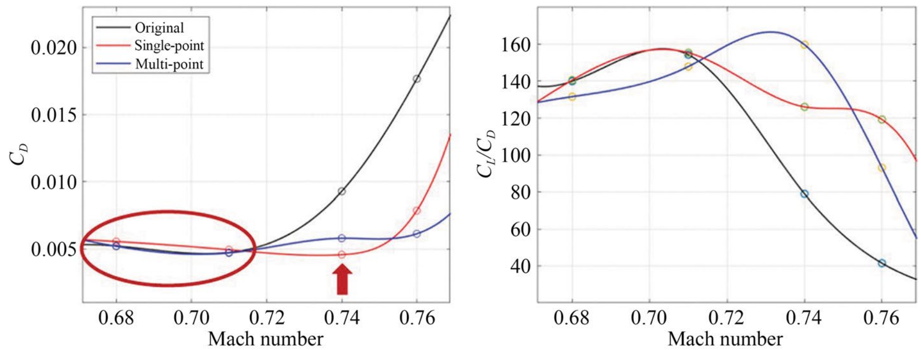

Fig. 23 Distribution of drag coefficient (left) and lift-drag ratio (right) according to Mach number at $C_L = 0.733$

with respect to the surface pressure coefficient distribution and airfoil geometry. The optimal solutions obtained by the CUnet model are validated by the FOM CFD calculation, and the prediction error in the drag coefficient is as little as 2%.

In Fig. 22a, for the design achieved by the single point optimization, shock strength seemed greatly reduced and the resultant wave drag was also reduced by 50.59% at Mach 0.74. The optimal shape obtained through the multi-point optimization also reduced the shock wave, shown in Fig. 22b, but it did not completely disappear, and the drag coefficient was reduced by a slightly smaller value of 37.38%. However, as can be seen in Fig. 23a, an increase in drag coefficient was observed at the off-design Mach numbers of the single-point optimization results. For the Mach numbers lower than the design point ($M = 0.74$), the drag coefficient was increased by 6% on average, whereas the amount of reduction was relatively smaller at the Mach number higher than the design point. This clearly highlights the disadvantage of the single-point design, where the optimal performance targeted at the single design condition can easily deteriorate. On the other hand, the optimal airfoil designed by the multi-point optimization does not show performance deterioration at the off-design Mach numbers, and a larger reduction in the drag coefficient was observed at high Mach numbers through large weights imposed on the objective function. Figure 23b shows the comparison of the lift-to-drag ratios from the two design results, and the overall performance of the optimal airfoil obtained by the multi-point optimization shows improvement with respect to a wide range of Mach numbers.

Furthermore, it is noteworthy that the utilization of the CUnet model for design purposes demonstrates both accuracy and effectiveness, particularly in achieving a remarkable reduction in computational cost. The optimization process involved a total of 60,000 function evaluations conducted over 3,000 design iterations, which necessitated CPU time of 1.94 h for single-point optimization and 2.37 h for multi-point optimization. In contrast, if the FOM CFD approach was directly employed in the present design optimization using the same computational platform, it would demand 194 days for single-point optimization and 776 days for multi-point optimization. Consequently, it becomes evident that the CUnet model empowers practical design optimization by leveraging readily available standard computing resources, thereby significantly diminishing computational expenses while concurrently delivering optimal solutions with utmost accuracy.

4 Conclusions and future work

A CUnet model has been developed, enabling rapid prediction of flow fields for various conditions: shape, Mach number, AoA, and time. This model has been employed to predict steady transonic flows with various shapes and flow conditions, as well as unsteady flows generated by a high AoA airfoil at 16 degrees. The distinctive feature of this model lies in its direct utilization of discrete solution points obtained from CFD solvers during the training process, without the need for interpolation or extrapolation, thereby avoiding unnecessary loss of accuracy. Additionally, a layer capable of incorporating flow conditions and time steps was added to the existing Unet architecture. The steady transonic flow predictions using the developed model exhibited an average error of approximately 1% when compared to FOM CFD solutions across the entire range of random shapes, Mach numbers $\in [0.6, 0.8]$, and AoA $\in [0^\circ, 3^\circ]$. Notably, the model accurately predicted the position and intensity of shock waves, which are highly sensitive to input conditions. For unsteady flow predictions using the CUnet, the average error was also below 1%, and the behavior of the flow affected by flow separation was faithfully captured. These results demonstrated superior accuracy compared to widely used ROMs such as POD-GPR and POD-LSTM. Furthermore, the flow field predictions using CUnet required only a fraction of time, approximately 0.1 s, which corresponds to a mere 0.05% of the computation time required by the FOM CFD solver. To show an example of the application of the validated model, multi-point optimization was performed at four different Mach number design conditions. Since CUnet provides solutions at the same grid points as FOM CFD, its outputs can be directly used for aerodynamic coefficient calculations. The results of multi-point optimization exhibited performance improvements across the entire Mach number range, in contrast to the degraded performance observed in single-point optimization under off-design conditions. Notably, despite utilizing a non-gradient-based algorithm, the computational time for this experiment was only 2.37 h. In comparison, the use of FOM CFD would require an estimated 776 days, assuming equivalent computational resources. Based on these findings, the developed CUnet model was demonstrated to calculate flow solutions in real-time with nearly the same level of accuracy as the FOM CFD solver used for training, thus highlighting its applicability in optimal design. This signifies a potential expansion of CFD simulations, which have been limited due to computational cost constraints, into engineering and industrial fields such as digital twins and extended reality (XR).

The framework developed in this study exhibits several limitations. We have directly applied CFD data generated from structured grids to the training of the deep learning model, successfully avoiding interpolation errors due to pixelation and the generation of artificial roughness on the airfoil surfaces. However, the present method is sensitive to the mesh topology, and the generation of body-fitted curvilinear grids for more complex geometries is highly challenging. A Cartesian grid with more accurate representation with the geometry surface can directly benefit from the present method. Additionally, due to the limited range of design variables used in the training data, some shock waves still persist on the optimized airfoil. Expanding the range of design variables necessitates additional training data and costs, a chronic issue in data-driven methodologies. Therefore, future efforts will focus on incorporating physics-informed loss functions [48, 49] to reduce the quantity of training data and to explore the possibilities for extrapolation.

Acknowledgements
Not applicable.

Authors' contributions
Mr. Wontae Hwang produced all results in the manuscript and carried out full analysis as the first author. Mr. Donggun Lee performed the revisions and conducted the model design and analysis for the 3D example. Dr. Seongim Choi is the corresponding author and is responsible for collaborating with other authors in developing the methods, reviewing the manuscript and supervising all research activities and findings. Dr. Junghun Shin conceived the presented idea initially. Dr. Kumwon Cho contributed to the interpretation of the results. All authors read and approved the final manuscript.

Funding
This work was supported by the National Research Council of Science & Technology (NST) grant by the Korea government (MSIT) (No. CRC21013).

Data availability
Data for training and testing the proposed model are available upon reasonable request.

Declarations

Competing interests
The authors declare that they have no conflict of interest.

Received: 11 June 2024 Revised: 9 October 2024 Accepted: 4 November 2024
Published online: 01 September 2025

References

1. Morita Y, Rezaeiravesh S, Tabatabaei N et al (2022) Applying Bayesian optimization with Gaussian process regression to computational fluid dynamics problems. J Comput Phys 449:110788. https://doi.org/10.1016/j.jcp.2021.110788
2. Duan Y, Cooling C, Ahn JS et al (2019) Using a Gaussian process regression inspired method to measure agreement between the experiment and CFD simulations. Int J Heat Fluid Flow 80:108497. https://doi.org/10.1016/j.ijheatfluidflow.2019.108497
3. Wang X, Kou J, Zhang W (2019) Unsteady aerodynamic modeling based on fuzzy scalar radial basis function neural networks. Proc Inst Mech Eng G J Aerosp Eng 233(14):5107–5121. https://doi.org/10.1177/0954410019836906
4. Erhart KJ, Gerace SA, Divo EA et al (2009) An RBF interpolated generalized finite difference meshless method for compressible turbulent flows. In: Proceedings of the ASME 2009 International Mechanical Engineering Congress and Exposition, Florida, 13-19 November 2009. Heat Transfer, Fluid Flows, and Thermal Systems, Parts A, B and C, Vol 9. ASME, New York, pp 571–581. https://doi.org/10.1115/IMECE2009-11452
5. Paulete-Periáñez C, Andrés-Pérez E, Lozano C (2019) Surrogate modelling for aerodynamic coefficients prediction in aeronautical configurations. In: Proceedings of the 8th European Conference for Aeronautics and Space Sciences (EUCASS), Madrid, 1-4 July 2019. https://doi.org/10.13009/EUCASS2019-870
6. Andrés-Pérez E, Paulete-Periáñez C (2021) On the application of surrogate regression models for aerodynamic coefficient prediction. Complex Intell Syst 7:1991–2021. https://doi.org/10.1007/s40747-021-00307-y
7. Lumley JL (1967) The structure of inhomogeneous turbulent flows. In: Yaglom AM, Tatarsky VI (eds) Atmospheric Turbulence and Radio Wave Propagation. Nauka, Moscow
8. Tan BT (2003) Proper orthogonal decomposition extensions and their applications in steady aerodynamics. Dissertation, Singapore-MIT Alliance
9. Alfonsi G, Primavera L (2007) The structure of turbulent boundary layers in the wall region of plane channel flow. Proc R Soc A 463(2078):593–612
10. Ortali G, Demo N, Rozza G (2022) A Gaussian process regression approach within a data-driven POD framework for engineering problems in fluid dynamics. Math Eng 4(3):1–16. https://doi.org/10.3934/mine.2022021
11. Quarteroni A, Manzoni A, Negri F (2016) Reduced basis methods for partial differential equations: an introduction. Springer International Publishing, Switzerland
12. Ali S, Ballarin F, Rozza G (2020) Stabilized reduced basis methods for parametrized steady Stokes and Navier-Stokes equations. Comput Math Appl 80(11):2399–2416. https://doi.org/10.1016/j.camwa.2020.03.019
13. Hall KC, Ekici K, Thomas JP et al (2013) Harmonic balance methods applied to computational fluid dynamics problems. Int J Comput Fluid Dyn 27(2):52–67. https://doi.org/10.1080/10618562.2012.742512
14. Shiea M, Buffo A, Vanni M et al (2020) Numerical methods for the solution of population balance equations coupled with computational fluid dynamics. Annu Rev Chem Biomol Eng 11:339–366. https://doi.org/10.1146/annurev-chembioeng-092319-075814

15. Mohammadi A, Shimoyama K, Karimi MS et al (2021) Efficient uncertainty quantification of CFD problems by combination of proper orthogonal decomposition and compressed sensing. Appl Math Model 94:187–225. https://doi.org/10.1016/j.apm.2021.01.012
16. Abreu LI, Cavalieri AVG, Schlatter P et al (2020) Spectral proper orthogonal decomposition and resolvent analysis of near-wall coherent structures in turbulent pipe flows. J Fluid Mech 900:A11. https://doi.org/10.1017/jfm.2020.445
17. Schmid PJ (2010) Dynamic mode decomposition of numerical and experimental data. J Fluid Mech 656:5–28. https://doi.org/10.1017/S0022112010001217
18. Tu JH (2013) Dynamic mode decomposition: theory and applications. Dissertation, Princeton University
19. Chaturantabut S, Sorensen DC (2010) Nonlinear model reduction via discrete empirical interpolation. SIAM J Sci Comput 32(5):2737–2764. https://doi.org/10.1137/090766498
20. Joshi SS, Prasad R, Choi S (2022) Discrete empirical interpolation based hyper-reduced order model for steady hypersonic flows. Paper presented at the AIAA SCITECH 2022 Forum, San Diego, 3-7 January 2022. https://doi.org/10.2514/6.2022-0079
21. Eivazi H, Veisi H, Naderi MH et al (2020) Deep neural networks for nonlinear model order reduction of unsteady flows. Phys Fluids 32(10):105104. https://doi.org/10.1063/5.0020526
22. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
23. Kutz JN (2017) Deep learning in fluid dynamics. J Fluid Mech 814:1–4. https://doi.org/10.1017/jfm.2016.803
24. Fukami K, Fukagata K, Taira K (2020) Assessment of supervised machine learning methods for fluid flows. Theor Comput Fluid Dyn 34:497–519. https://doi.org/10.1007/s00162-020-00518-y
25. Kang YE, Yang S, Yee K (2022) Physics-aware reduced-order modeling of transonic flow via β-variational autoencoder. Phys Fluids 34(7):076103. https://doi.org/10.1063/5.0097740
26. Catalani G, Costero D, Bauerheim M et al (2023) A comparative study of learning techniques for the compressible aerodynamics over a transonic RAE2822 airfoil. Comput Fluids 251:105759. https://doi.org/10.1016/j.compfluid.2022.105759
27. Hasegawa K, Fukami K, Murata T et al (2020) Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. Theor Comput Fluid Dyn 34:367–383. https://doi.org/10.1007/s00162-020-00528-w
28. Peng JZ, Chen S, Aubry N et al (2020) Time-variant prediction of flow over an airfoil using deep neural network. Phys Fluids 32(12):123602. https://doi.org/10.1063/5.0022222
29. Deng Z, Liu H, Shi B et al (2023) Temporal predictions of periodic flows using a mesh transformation and deep learning-based strategy. Aerosp Sci Technol 134:108081. https://doi.org/10.1016/j.ast.2022.108081
30. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells W et al (eds) Medical image computing and computer-assisted intervention – MICCAI 2015. 18th International Conference, Munich, 5-9 October 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham, pp 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
31. Hicks RM, Henne PA (1978) Wing design by numerical optimization. J Aircr 15(7):407–412. https://doi.org/10.2514/3.58379
32. LeCun Y, Boser B, Denker JS et al (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551. https://doi.org/10.1162/neco.1989.1.4.541
33. Fukami K, Fukagata K, Taira K (2019) Super-resolution reconstruction of turbulent flows with machine learning. J Fluid Mech 870:106–120. https://doi.org/10.1017/jfm.2019.238
34. Jin X, Cheng P, Chen WL et al (2018) Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder. Phys Fluids 30(4):047105. https://doi.org/10.1063/1.5024595
35. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning, Lille, 6-11 July 2015
36. Garbin C, Zhu X, Marques O (2020) Dropout vs. batch normalization: an empirical study of their impact to deep learning. Multimed Tools Appl 79:12777–12815. https://doi.org/10.1007/s11042-019-08453-9
37. Mathieu M, Couprie C, LeCun Y (2015) Deep multi-scale video prediction beyond mean square error. arXiv preprint, arXiv:1511.05440. https://doi.org/10.48550/arXiv.1511.05440
38. Krige DG (1951) A statistical approach to some basic mine valuation problems on the Witwatersrand. J South Afr Inst Min Metall 52(6):119–139
39. Gramacy RB (2020) Surrogates: Gaussian process modeling, design, and optimization for the applied sciences. CRC Press, New York
40. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
41. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536. https://doi.org/10.1038/323533a0
42. Scarselli F, Gori M, Tsoi AC et al (2009) The graph neural network model. IEEE Trans Neural Netw 20(1):61–80. https://doi.org/10.1109/TNN.2008.2005605
43. Hyunsol P (2023) High-accuracy wind turbine flow analysis and prediction based on deep learning. Paper presented at the KSAS 2023 Spring Conference, Ramada Plaza Jeju, Jeju, 19-21 April 2023
44. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Curran Associates Inc, Red Hook
45. Drela M (1998) Pros & cons of airfoil optimization. Front Comput Fluid Dyn 1998:363–381. https://doi.org/10.1142/9789812815774_0019
46. Nemec M, Zingg DW, Pulliam TH (2004) Multipoint and multi-objective aerodynamic shape optimization. AIAA J 42(6):1057–1065. https://doi.org/10.2514/1.10415
47. Deb K, Pratap A, Agarwal S et al (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197. https://doi.org/10.1109/4235.996017

48. Wu C, Zhu M, Tan Q et al (2023) A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. Comput Methods Appl Mech Eng 403:115671. https://doi.org/10.1016/j.cma.2022.115671
49. Moseley B, Markham A, Nissen-Meyer T (2023) Finite basis physics-informed neural networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. Adv Comput Math 49(4):62. https://doi.org/10.1007/s10444-023-10065-9

Publisher's Note