

Article

Fast Aerodynamic Prediction of Airfoil with Trailing Edge Flap Based on Multi-Task Deep Learning

Chi Zhang , Zhiyuan Hu , Yongjie Shi and Guohua Xu *

National Key Laboratory of Helicopter Aeromechanics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

* Correspondence: ghxu@nuaa.edu.cn

Abstract: Conventional methods for solving Navier–Stokes (NS) equations to analyze flow fields and aerodynamic forces of airfoils with trailing edge flaps (TEFs) are known for their significant time cost. This study presents a Multi-Task Swin Transformer (MT-Swin-T) deep learning framework tailored for swift prediction of velocity fields and aerodynamic coefficients of TEF-equipped airfoils. The proposed model combines a Swin Transformer (Swin-T) for flow field prediction with a multi-layer perceptron (MLP) dedicated to lift coefficient prediction. Both networks undergo gradient updates through the shared encoder component of the Swin Transformer. Such a trained network model for computational fluid dynamics simulations is both effective and robust, significantly improving the efficiency of complex aerodynamic shape design optimization and flow control. The study further investigates the impact of integrating multi-task learning loss functions, skip connections, and the network’s structural design on prediction accuracy. Additionally, the effectiveness of deep learning in improving the aerodynamic simulation efficiency of airfoils with TEF is examined. Results demonstrate that the multi-task deep learning approach provides accurate predictions for TEF airfoil flow fields and lift coefficients. The strategic combination of these tasks during network training, along with the optimal selection of loss functions, significantly enhances prediction accuracy compared with the single-task network. In a specific case study, the MT-Swin-T model demonstrated a prediction time that was 1/7214 of the time necessitated by CFD simulation.

Keywords: airfoil with trailing edge flap; CFD; multi-task deep learning; Swin Transformer



Citation: Zhang, C.; Hu, Z.; Shi, Y.; Xu, G. Fast Aerodynamic Prediction of Airfoil with Trailing Edge Flap Based on Multi-Task Deep Learning. *Aerospace* **2024**, *11*, 377. <https://doi.org/10.3390/aerospace11050377>

Academic Editor: Christian Breitsamter

Received: 7 April 2024

Revised: 30 April 2024

Accepted: 6 May 2024

Published: 9 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Traditionally, computational fluid dynamics (CFD) simulation and wind tunnel experiments have been the main methods for obtaining the external flow field of obstacles. However, these methods are associated with substantial costs and time requirements. With the development of the computer science field, deep learning has seen rapid growth in applications such as speech recognition [1], image recognition [2], and natural language processing [3] in recent years. Against this backdrop, the field of fluid mechanics has also gradually started to explore the application of deep learning. Deep learning network models can perform effective feature recognition and dimensionality reduction on flow fields. The features after dimensionality reduction, while being quantifiable and visualizable, are also often used in predicting and optimizing the best designs among other problems. The original issue of needing massive simulations to analyze model parameters can, through feature recognition and dimensionality reduction, greatly improve computational efficiency while maintaining a certain level of accuracy.

For instance, Ribeiro et al. [4] conducted a comparison of neural network architecture for predicting laminar flow near cylinders, demonstrating that the U-Net architecture-based neural network outperformed others in simulating the flow field. Moreover, Guo et al. [5] proposed a CNN model for predicting the external flow field of rough bodies, showcasing the capability of CNN to quickly predict the flow field. However, this work primarily

focused on qualitatively estimating the velocity field rather than accurately predicting aerodynamic characteristics. Furthermore, Sekar et al. [6] introduced a reverse design method based on CNN to establish the mapping relationship between pressure coefficient and geometry. In addition, Thuerey et al. [7] investigated the accuracy of flow field prediction by deep learning model, focusing on how training dataset size and the number of weights affect the prediction accuracy. Finally, Tangsali et al. [8] explored the generalization ability of models based on encoder–decoder architecture in predicting aerodynamic flow fields with various geometric changes.

Furthermore, researchers have delved into the application of deep learning networks for predicting airfoil flow fields. One method involves utilizing deep neural networks to accurately infer Reynolds-averaged Navier–Stokes (RANS) solutions on two-dimensional airfoils [9]. Moreover, Wu et al. [10] introduced an enhanced data Generative Adversarial Network (GAN), named daGAN, capable of achieving fast and accurate flow field prediction even with sparse training data. In addition, a CNN method based on an encoder–decoder architecture has been proposed to simulate the pressure field around airfoils. Compared to the traditional CFD, this method achieves high accuracy and a significant acceleration [11]. Finally, Gupta et al. [12] proposed a combined CNN and multi-layer perceptron (MLP) approach to predict the flow field of incompressible steady layer flow through an airfoil. They used OpenFOAM to solve the Navier–Stokes (NS) equation to generate a training data set, and the results showed that the method was accurate and efficient.

The existing studies often apply small-scale datasets, focusing only on a limited number of airfoil shapes or on specific Angles of Attack (AoA) and Mach numbers. Moreover, they typically concentrate on the common, single-element airfoils, with limited consideration for multi-element airfoils, such as those with trailing edge flaps (TEFs). Additionally, there is a common challenge where these studies target only a single task, e.g., predicting the flow field or the aerodynamic forces, leading to a low data utility rate.

In addition to the common single-element airfoils, multi-element airfoils have garnered considerable attention across various disciplines. Numerical investigations using CFD methods have explored the aerodynamic performance of bionic airfoils equipped with trailing edge flaps [13]. These studies have highlighted the significant impact of flap angle and position on the airfoil's aerodynamic performance. The experimental study on NACA 0012 airfoil with varying flap shapes demonstrated superior lift–drag performance in deformed flap configuration compared to articulated flap airfoil at low AoA [14]. The application of trailing edge flaps with micro baffles in wind turbine blades has shown promise in improving performance and delayed stall onset [15]. Moreover, the influence of the trailing edge flap on the flow around a NACA0015 airfoil was investigated experimentally [16]. However, compared to single-element airfoils, the pre-processing, CFD simulation, and post-processing operations required to analyze the flow field and aerodynamic characteristics of multi-element airfoils are more complex and time-consuming. Therefore, there is considerable significance in extending the deep learning method for predicting the flow field of single-element airfoil to multi-element airfoil analyses.

The aforementioned studies primarily focus on extracting features from the airfoil shape and initial conditions, utilizing these features to either reconstruct flow field information or predict aerodynamic characteristics separately. However, by adopting multi-task deep learning methods that have emerged in the field of computer vision in recent years, it becomes feasible to predict both flow field and aerodynamic characteristics simultaneously. This multi-task learning approach includes sharing feature extraction, sharing prediction results, and optimizing loss function. Therefore, the research focus consists of sharing feature extraction. Leveraging multi-task learning allows for the concurrent prediction of the two-dimensional flow field and aerodynamic coefficients of an airfoil with TEF can be predicted simultaneously using a unified backbone network.

Several studies have advocated for the implementation of multi-task learning networks utilizing shared encoders. For instance, Yue et al. [17] introduced a Secure Multi-Task Learning (SMTL) model, consisting of a common encoder, private encoder, and gate control

shared by all tasks. Moreover, Hu and Singh [18] proposed a unified Transformer model UniT, employing a shared decoder for multi-task learning of encoded input. In addition, Rizi and Granitzer [19] proposed a Joint Automatic encoder framework for Multi-task network Embedding (JAME). As a result, this framework encodes network structure, node attributes, and tags into shared feature expressions. Chen et al. [20] investigated the possibility of combining different tasks in a single network through shared convolutional encoders. The research mentioned above focuses primarily on the combinations of multi-task networks and the aspects of loss functions, without addressing the application of multi-task learning concepts to practical engineering problems. Based on the above research, this work proposes the idea of multi-task learning into the flow field prediction, combining the latest Swin Transformer (Swin-T) with the traditional multi-layer perceptron (MLP) [21] to simultaneously predict both the flow field and aerodynamic coefficients of an airfoil with TEF. Most current studies have only focused on training with a limited number of single-element airfoils. We enhance the diversity of airfoil shapes available for training and have added the parameter of deflection angle. We provided Swin-T as a reduced-order model that balances accuracy and predictive efficiency for the design optimization and active control research of airfoils with TEF.

Therefore, the primary contribution of this study lies in the introduction of a multi-task deep learning framework designed for predicting flow field information and lift airfoil coefficients with trailing edge flaps. Unlike most existing research that typically employs CNNs as the core model, the proposed neural network architecture utilizes a combination of Swin-T [22] and MLP to form the Multi-Task Swin Transformer (MT-Swin-T). The Swin-T distinguishes itself from the CNN architecture by incorporating a multi-head self-attention mechanism that expands its receptive field, improving the expressiveness of the model while saving training parameters. Moreover, the MT-Swin-T applies a hard parameter-sharing multi-task network structure, where the same parameters are shared across the main body of the model when processing different tasks; yet, distinct output structures are employed for different tasks. This involves using the shared encoder part of the Swin-T to learn features from both the airfoil shape and its initial conditions. Subsequently, the decoder is applied to predict velocity field information, while simultaneously the MLP predicts lift coefficients. It is worth mentioning that the airfoil geometry data were sourced from the UIUC Airfoil Data Site [23]. The deep learning method proposed significantly accelerates the prediction speed of the flow field and shows prospects for achieving rapid aerodynamic design and optimization. In the design and optimization of airfoil shapes and active control, employing such a trained network model as a surrogate model enables rapid predictions of different shapes and operating conditions so as to facilitate the optimization or design of airfoils that meet design objectives. Compared to the surrogate models used in traditional optimization efforts, our method not only ensures real-time predictions but also broadens the applicable range of shapes and parameters. Moreover, it provides more detailed and visualized flow field information.

To improve the model's generalization ability, a significantly large dataset, comprising 38,880 training samples and 4104 validation samples, was utilized. This methodology offers a considerable time-savings advantage over CFD simulations, requiring high convergence demands and time-consuming iterative processes. Unlike previous studies, this paper creatively integrates deep learning methods to predict the flow field around airfoils with trailing edge flaps and applies the concept of multi-task learning to simultaneously predict velocity fields and aerodynamic force coefficients. Furthermore, this work explores the impact of different multi-task learning loss function combinations, skip connections, and multi-task learning network structures regarding the network model predictive accuracy.

To sum up, the rest of this article is organized as follows: Section 2 describes the process of dataset generation, the network structure of MT-Swin-T, and the design of the loss function. Section 3 compares the network-predicted results with the CFD results, and evaluates the influence of the different factors on the accuracy of the proposed network model. Finally, Section 4 concludes this study and proposes some future studies.

2. Methodology

2.1. Dataset Generation

From UIUC Airfoil Data Site, 200 original airfoil shapes were selected, including symmetric, asymmetric, thick, and thin airfoils. Figure 1 shows the airfoils with the maximum thickness (e858) and the minimum thickness (OA209) among the 200 original airfoils. The selection of initial parameters of the dataset is based on Latin Hypercube Sampling (LHS), which is a statistical method for generating nearly random samples from multi-dimensional distributions. Its implementation steps are as follows:

1. Divide each input variable range into an interval equal to the number of samples;
2. Randomly select a point in each interval and perform this operation on all input variables;
3. The selected points are randomly paired to generate a group of sampling points;
4. Repeat this process to generate all samples.

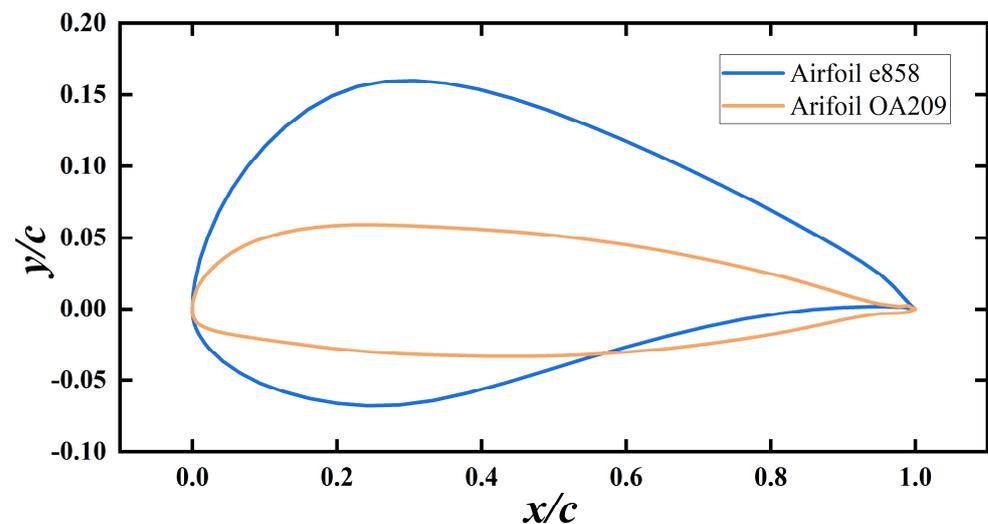


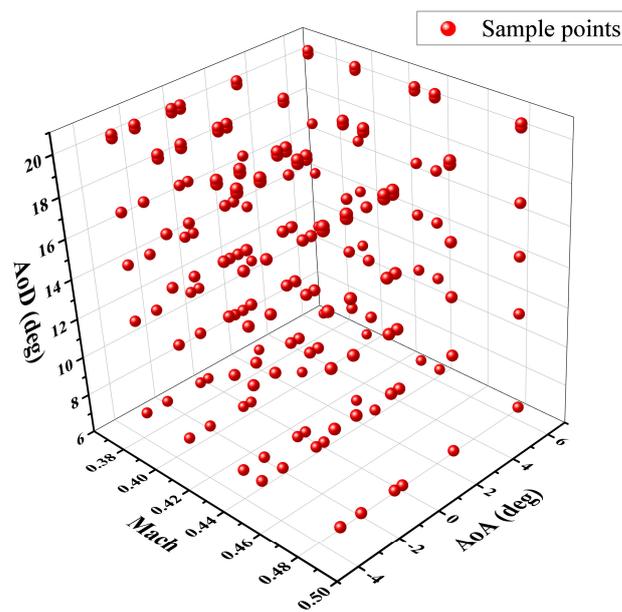
Figure 1. Geometries of airfoils with the maximum thickness (e858) and the minimum thickness (OA209).

LHS can effectively reduce the variance of samples, thus enhancing the sampling quality and achieving higher coverage while considering the same number of samples; thus, it can provide a more representative sample distribution.

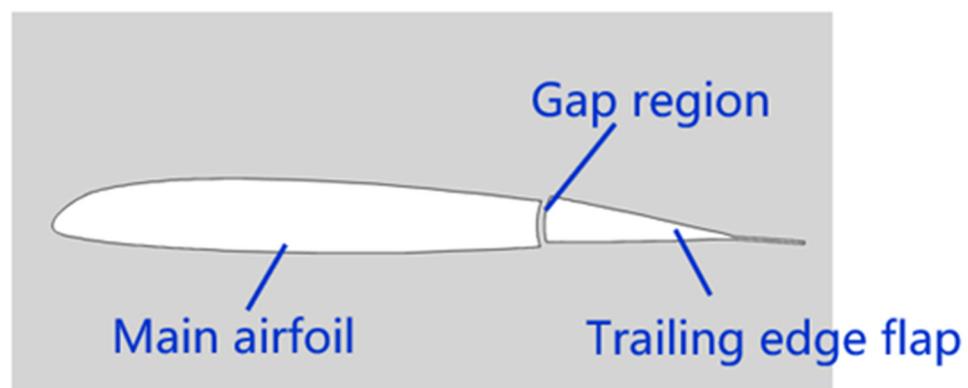
In this work, three initial condition variables are set for each case, namely, the incoming flow Mach number (Ma), the airfoil AoA, and flap Angle of Deflection (AoD). For 200 original airfoils, the sampling range of Ma is between 0.2 and 0.5, the sampling range of AoA ranges between -6 deg and 6 deg, and the sampling range of AoD ranges between 5 deg to 25 deg, as illustrated in Table 1. Considering that the simulation environment for all cases in this article is consistent and the chord length for the airfoil is uniformly 1 m, the Mach number conditions provided are proportional to the corresponding Reynolds numbers. Hence, we have presented in Table 1 the upper and lower limits of the Mach numbers along with corresponding Reynolds numbers for reference. Moreover, for each of the 200 airfoils, LHS is used in the sampling ranges of Ma , AoA, and AoD, respectively [24]. Each variable is sampled six times, and each airfoil produces 216 samples, leading to a total of 43,200 samples for the 200 airfoils. Moreover, Figure 2 represents the distribution space of parameter sampling points of the sample airfoil. Among these 200 airfoils, 180 are employed as training airfoils, 19 for validation (the validation set is used to evaluate the quality of training model unbiased during training, such as detecting whether there is over-fitting), and one is used as test airfoil (Sg6042). The training and validation sets are randomly selected before each training, resulting in 38,880 training cases, 4104 validation cases, and 216 test cases.

Table 1. Sampling range of cases.

Variable	Min	Max	Counts
Ma	0.2	0.5	6
Re	4.66×10^6	1.16×10^7	
AoA	−6 deg	6 deg	6
Deflection angle (AoD)	5 deg	25 deg	6
Conditions per airfoil	-	-	216
Number of airfoils	-	-	200
Total counts of cases	-	-	43,200

**Figure 2.** Distribution space of parameter sampling points.

As shown in Figure 3, all airfoils considered in this study maintain a chord length of 1 m and are truncated at 80% chord length. The main airfoil is positioned at the front, followed by the TEF at the rear.

**Figure 3.** Airfoil with trailing edge flap.

Moreover, Rotorcraft Aerodynamics and Aeroacoustics Solver (RADAS) [25], an in-house CFD solver based on RANS equation is used to solve the flow field, and ROE-MUSCL spatial discrete scheme is selected to generate better boundary layer capture ability. Implicit LU-SGS time discretization is selected to accelerate convergence, and the Spalart–Allmaras

(S-A) model, which is commonly used in aerodynamic external flow field simulation, is applied to model the turbulence model. To improve the reusability of airfoil grid, the main airfoil grid and flap grid are combined by overset grid method. In the simulation cases considered in the present study, all the residual convergence values were set to 5.0×10^{-5} . Expressly, 20,000 maximum iteration steps were set for the cases.

The HH-06 airfoil and its experimental data (see reference [26]) were taken as the simulation object. The geometric shape of the airfoil with TEF is shown in Figure 3. The sample overset grids are shown in Figure 4. Figure 5 presents the surface pressure coefficients for the simulation case 1~3 and the reference experimental data under the conditions of $\text{AoA} = -4.03^\circ$ and $\text{Ma} = 0.758$, in which the TEF deflection angle is 4° downward. For the investigation of grid sensitivity in this section's CFD method, the grid topology was maintained, and only the number of grid cells was altered. Different amounts of grids were chosen for computation, and three sets of grids with varying numbers of cells were established. The detailed grid information is shown in Table 2. As indicated in Figure 5, there is a noteworthy deviation from the experimental values when the number of grids is relatively low. However, once the grid count exceeds 21,047, further increments in the total grid count do not yield a substantially improved accuracy effect. Therefore, to ensure effective simulation precision, the cases involved in this paper maintain a grid count of about 21,047 or higher. Therefore, the data obtained from CFD simulation can be utilized as ground truth to validate the prediction accuracy of the network model here.

For the ease of the neural network's predictive output, the x-direction freestream velocity (V_x) and y-direction freestream velocity (V_y) from the CFD simulation data are extracted using a 128×128 Cartesian grid. Inside the airfoil, V_x and V_y data are set to zero, resulting in a data dimension of $2 \times 128 \times 128$. The leading edge of the airfoil is employed as the coordinate origin, and the range for the extraction of flow field data is denoted as the area $\{(x, y) \mid x \in [-0.5, 1.5], y \in [-0.5, 0.5]\}$. This area captures the primary characteristics of the flow field around the airfoil with a TEF, providing critical reference value for airfoil design and optimization. The lift coefficient computed by the CFD simulations for each state is applied in conjunction with the $2 \times 128 \times 128$ dimensional V_x and V_y data as the output part of the dataset.

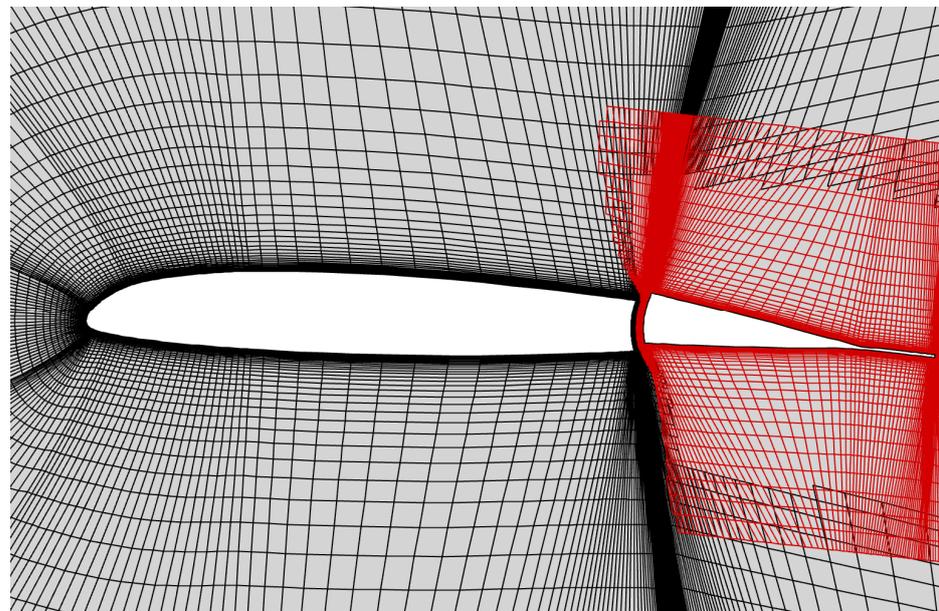


Figure 4. Overset grids of airfoil with TEF (the main airfoil grid is black, and the TEF grid is red).

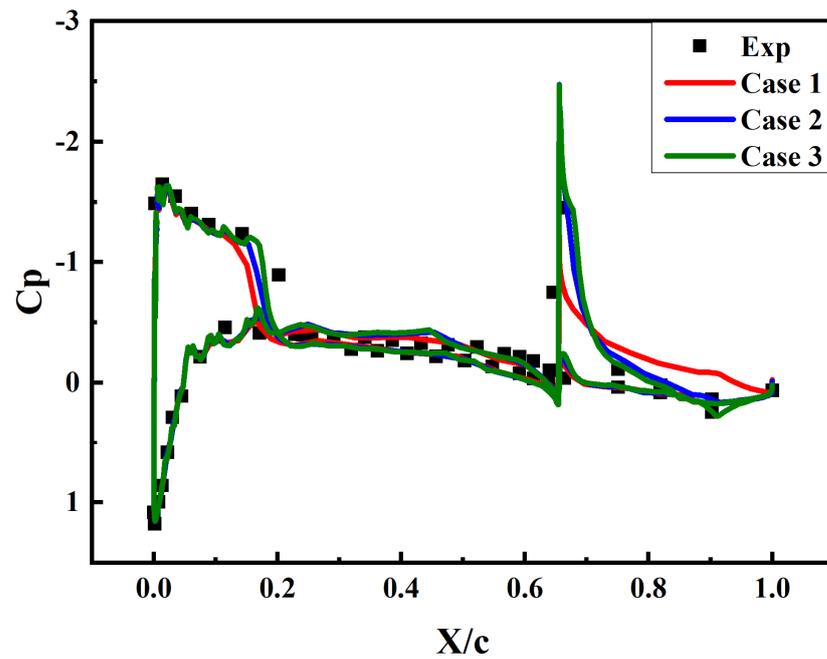


Figure 5. Surface pressure coefficients of the HH-06 airfoil with different cell numbers.

Table 2. Mesh cell numbers of validation cases.

	Main Foil Cell Numbers	Flap Cell Numbers	Total Cell Numbers
Case 1	6106	5082	11,188
Case 2	12,261	8786	21,047
Case 3	23,885	14,685	38,570

2.2. Multi-Task Swin Transformer

Input Layer: As shown in Figure 5, the input channel of the network uses freestream velocity in the x and y directions as well as the signed distance function (SDF) of airfoil as three input channels. The variable values of airfoil internal region are all set to zero, and all their dimensions are 128×128 . As far as the steady flow field of the two-dimensional airfoil is concerned, the far-field velocity is close to that of the freestream except near the airfoil and the wake influence area. Moreover, the values of $FreestreamX$ and $FreestreamY$ channels in the input layer are close to those of the output layer at some corresponding points. As a result, the gradient changes slightly during iteration, and the convergence can be accelerated.

When it comes to representing the airfoil shape, the SDF offers clear advantages over binary representation as it provides a more abstract and intuitive depiction of the airfoil geometry, allowing it to efficiently capture the airfoil's silhouette. This feature is particularly crucial when dealing with complex airfoil shapes, as it facilitates learning. Additionally, SDF supplies extra information to each data point, enriching the input for deep learning algorithms, and ultimately enhancing the model's predictive capabilities and adaptability to various airfoil shapes.

The effectiveness of SDF in neural network training has been demonstrated in [27]. Moreover, the mathematical representation of an SDF for a set of points X is the minimum distance of each given point $x \in X$ from the boundary $\partial\Omega$ of object Ω :

$$SDF(x) = \begin{cases} d(x, \partial\Omega) & x \notin \Omega \\ 0 & x \in \partial\Omega \\ 0 & x \in \Omega \end{cases} \quad (1)$$

where Ω represents the interior of the airfoil and $\partial\Omega$ denotes the airfoil boundary. Moreover, $d(x, \partial\Omega) = \min_{x_i \in \partial\Omega} (x - x_i)$ highlights the shortest distance from point x to the boundary $\partial\Omega$. While using this approach, the *SDF* value is positive outside the airfoil and null on the airfoil surface and within it, rendering it a signed value. To generate the *SDF*, airfoil geometry information derived from CFD results is applied. The calculation of the minimum distance to the main airfoil surface and the TEF surface for a 128×128 grid of points in space and the selection of the smaller of these two values as $d(x, \partial\Omega)$, yields in the formation of a structured *SDF* input for the process.

Overall network architecture: The general structure of the proposed MT-Swin-T is illustrated in Figure 6. The MT-Swin-T consists of a Swin-T for flow-field prediction and an MLP for C_L prediction. The Swin-T comprises an encoder, a decoder, and skip connections, while the MLP is based on a two-layer fully connected network. The fundamental unit of the Swin-T is the Swin Transformer Block. Concerning the encoder, to transform the input into sequence embedding, the input layer is divided into non-overlapping patches, with a patch size of 4×4 . This division method yields each patch having a feature dimension of $4 \times 4 \times 3 = 48$. Additionally, a linear embedding layer is employed to project the feature dimensions onto a certain dimension C ($C = 64$ is used in this article). The transformed patch tokens pass through several Swin-T blocks and patch merging layers, aiming to generate the hierarchical feature representations. Specifically, the patch merging layer facilitates down-sampling and dimensionality increase, whereas the Swin-T Blocks focus on learning feature representations. Inspired by U-Net [28], a symmetric transformer model-based decoder is employed. The decoder comprises Swin-T Blocks and patch-expanding layers. The extracted contextual features involve integrating the extracted contextual features with multi-scale features from the encoder through skip connections to address spatial information loss due to down-sampling. Unlike the patch merging layer, the patch expanding layer is specially designed for up-sampling. It transforms the feature maps of adjacent dimensions into larger feature maps with twice the resolution. Finally, the last patch expanding layer performs up-sampling 4 times to restore the feature map's resolution back to the input resolution (128×128). Subsequently, a linear projection layer is applied to output flow-field velocity information at each grid point.

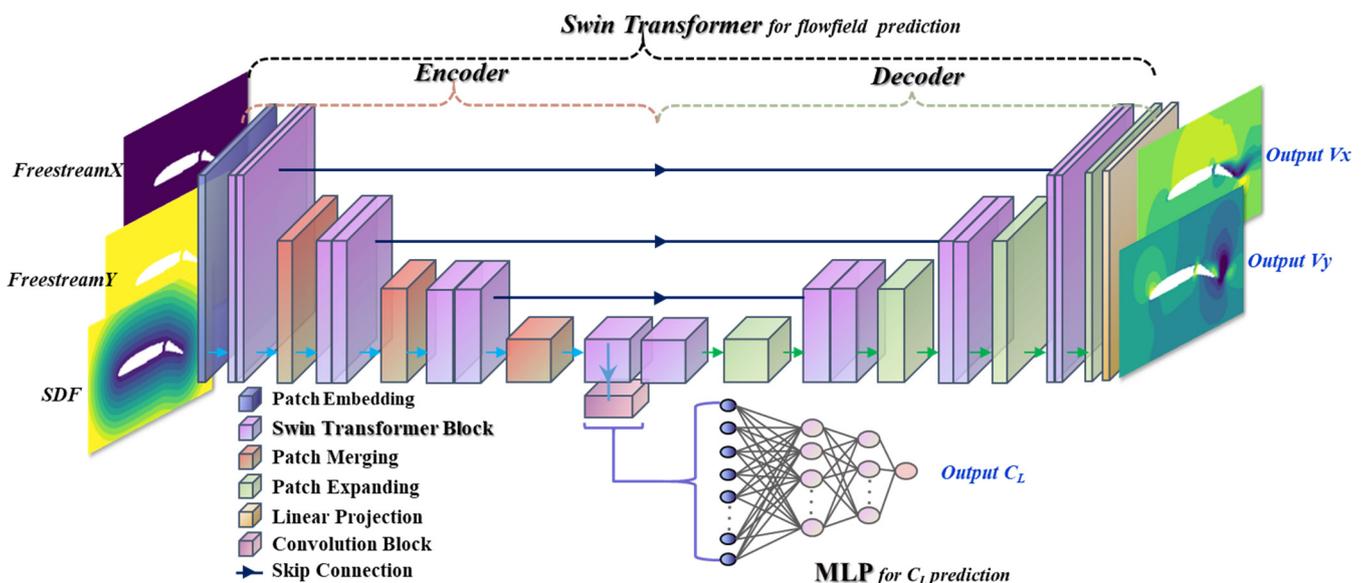


Figure 6. Overall network architecture of MT-Swin-T.

Swin Transformer block: Distinct from the traditional Multi-head Self-Attention (MSA) modules, Swin-T blocks are designed based on the concept of shifted windowing. Referring to Figure 7, there are two adjacent Swin-T blocks. Each block comprises a Layer Normalization (LN), an MSA mechanism, a residual connection, and a dual-layer MLP. The

initial block employs a Window-based Multi-head Self-Attention (W-MSA) module, while its subsequent counterpart utilizes a Shifted Window-based Multi-head Self-Attention (SW-MSA) module. The sequence of Swin-T Blocks, operating based on the shifted window approach, is mathematically expressed as follows:

$$\begin{aligned} \hat{z}^l &= W - \text{MSA}(\text{LN}(z^{l-1})) + z^{l-1}, \\ z^l &= \text{MLP}(\text{LN}(\hat{z}^l)) + \hat{z}^l \\ \hat{z}^{l+1} &= \text{SW-MSA}(\text{LN}(z^l)) + z^l \\ z^{l+1} &= \text{MLP}(\text{LN}(\hat{z}^{l+1})) + \hat{z}^{l+1} \end{aligned} \quad (2)$$

where \hat{z}^l and z^l represent the outputs of the W-MSA module and the MLP module of the first block, respectively, whereas \hat{z}^{l+1} and z^{l+1} denote the outputs of the SW-MSA module and the MLP module of the second block, respectively.

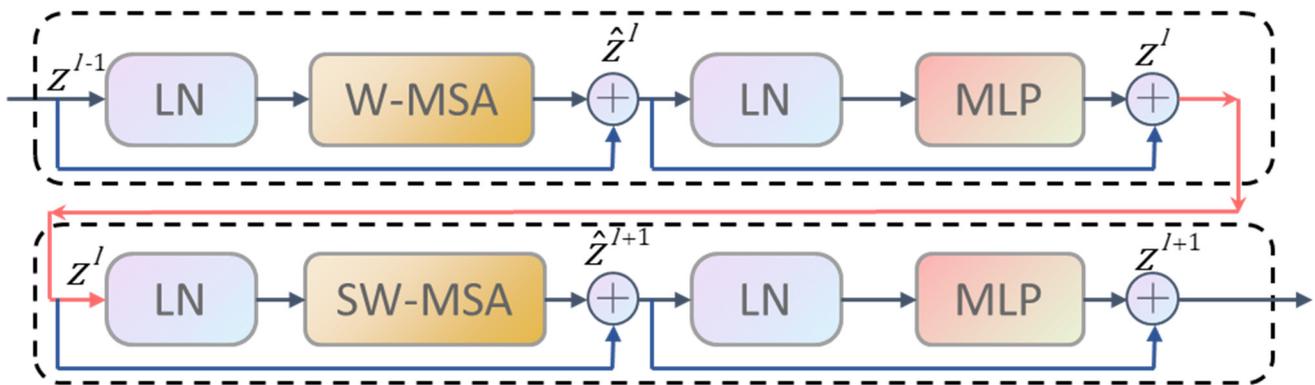


Figure 7. Swin Transformer Block.

In the encoder, tokenized inputs with a resolution of 32×32 and dimensionality of C are fed into two successive Swin-T blocks to perform feature representation learning, where both feature dimensions and resolution remain constant. Meanwhile, the patch merging layer reduces the number of tokens (two times down-sampling) and increases the feature dimensionality to twice the original dimensions. This process is repeated three times inside the encoder.

Patch merging layer: The input patches undergo subdivision based on four distinct segments, which are subsequently amalgamated via a patch merging layer. This operation reduces the feature resolution by half while quadrupling the feature dimension through concatenation. To address this, a linear layer is employed to normalize the feature dimension, scaling it to twice the original dimension.

Decoder: Mirroring the encoder, the decoder is architecturally symmetrical and is constructed using Swin-T blocks. Diverging from the encoder's patch-merging layer, the decoder incorporates a patch-expanding layer to up-sample the deeply extracted features. This layer transforms the feature maps from adjacent dimensions into a higher resolution map, achieving up-sampling twice, while concurrently reducing the feature dimension by half. For illustration, let us consider the initial patch-expanding layer. Before up-sampling, the input features ($4 \times 4 \times 8C$) undergo a linear layer, amplifying the feature dimension to twice its original size ($4 \times 4 \times 16C$). Then, a rearrangement operation is performed to upscale the feature resolution to twice that of the input, while scaling down the feature dimension to a quarter ($4 \times 4 \times 16C \rightarrow 8 \times 8 \times 4C$).

Skip connection: Similar to U-Net [28], skip connections are used to fuse multi-scale features from the encoder with up-sampled features. Shallow features are concatenated with deep features to mitigate the loss of spatial information caused by down-sampling. This is followed by a linear layer, ensuring that the dimensionality of the concatenated

features remains the same as that of the up-sampled features. For more detail, Section 4 will discuss the impact of skip connections on model performance.

Output Layer: In the final linear projection of the decoder, the feature dimensionality is restored to $2 \times 128 \times 128$, where two channels V_x and V_y are used, each having a dimension of 128×128 . Note that the velocity data in both channels is normalized within the $[0, 1]$ range to minimize the training errors caused by limited numerical precision during the training phase.

MLP for C_L prediction: A convolutional layer is employed to flatten the feature vector obtained from the final Swin-T block in the Encoder into a 1×512 one-dimensional vector, which is then fed into the MLP that includes two fully connected layers (with neuron counts sequentially being 128 and 64), to generate the output C_L .

2.3. Loss Function

Concerning the model's output, it is necessary to extract the flow field data using 128×128 grid points as well as the lift coefficient for each case and determine the loss function. For a given set of model output data and CFD simulation data, the model minimizes the total loss function, composed of two specific loss functions for two tasks. The mathematical expressions are denoted as follows:

$$\begin{aligned} \text{Loss}_{\text{Flow}} &= \frac{1}{mN_{\text{Flow}}} \sum_{l=1}^m \left(\frac{1}{2} \|Y_{ij_{\text{pred}}}^m - Y_{ij_{\text{CFD}}}^m\|^2 \right) \quad \text{for } ij \in \Omega_{\text{Flow}}, \\ \text{Loss}_{C_L} &= \frac{1}{m} \sum_{l=1}^m \left(\frac{1}{2} \|C_{L_{\text{pred}}}^m - C_{L_{\text{CFD}}}^m\|^2 \right), \\ \text{Loss} &= \lambda \text{Loss}_{\text{Flow}} + (1 - \lambda) \text{Loss}_{C_L}. \end{aligned} \quad (3)$$

where Y_{CFD} and Y_{pred} represent the CFD and the network model output velocity fields, respectively. Moreover, $C_{L_{\text{CFD}}}$ and $C_{L_{\text{pred}}}$ represent the CFD and network model output lift coefficients, respectively, whereas m denotes the batch size, which denotes the number of cases being trained simultaneously in each batch, and N_{Flow} represents the number of grid points in the flow field. To ensure that the accuracy of flow field data is sufficient for the calculation of aerodynamic characteristics, we extract the mean square error (MSE) of the flow field part and the lift coefficient separately from the output of the MT-Swin-T model, combined using a coefficient λ to form the complete network's loss function. The role of the λ coefficient consists of balancing the rate of gradient descent during training for both loss functions.

3. Results

In this section, we have studied the impacts of the loss function coefficient, skip connections, and multi-task structure on the accuracy of prediction results. We also compared the prediction time consumption between the proposed network model and the CFD simulation method. Five cases with significant differences in conditions were selected from the test set as examples for this section. The hyperparameters setting of the network model in this part and the hardware configuration have been introduced in Section 3.4.

3.1. Effect of Loss Function Coefficient

During multi-task learning, multiple losses are generated; however, the optimizer can optimize just one. Additionally, tasks with a superior number of parameters tend to dominate the shared layer model parameters, leading to other tasks with fewer parameters being unable to acquire effective shared features. Therefore, different weight coefficients must be assigned to the loss functions of the distinct tasks and multiple losses must be aggregated into one by weighted summation. This coefficient, represented as λ in the loss function, is expressed as follows: $\text{Loss} = \lambda \text{Loss}_{\text{Flow}} + (1 - \lambda) \text{Loss}_{C_L}$.

Referring to Figure 8, the velocity fields and prediction results of three cases are compared, and significantly different initial parameters under the three scenarios of loss function coefficients are encountered. The velocity fields provide a comparison between the V_x and V_y predictions of the MT-Swin-T model and the CFD results. Additionally, Figure 8 presents a comparison of 216 C_L predictions obtained by the MT-Swin-T model and CFD computation for the test airfoil Sg6042 under different input parameters. It is important to mention that the horizontal axis is the C_L obtained by CFD calculations whereas the vertical axis represents the MT-Swin-T model prediction results. Moreover, the closer they are to the diagonal line, the more similar both results are.

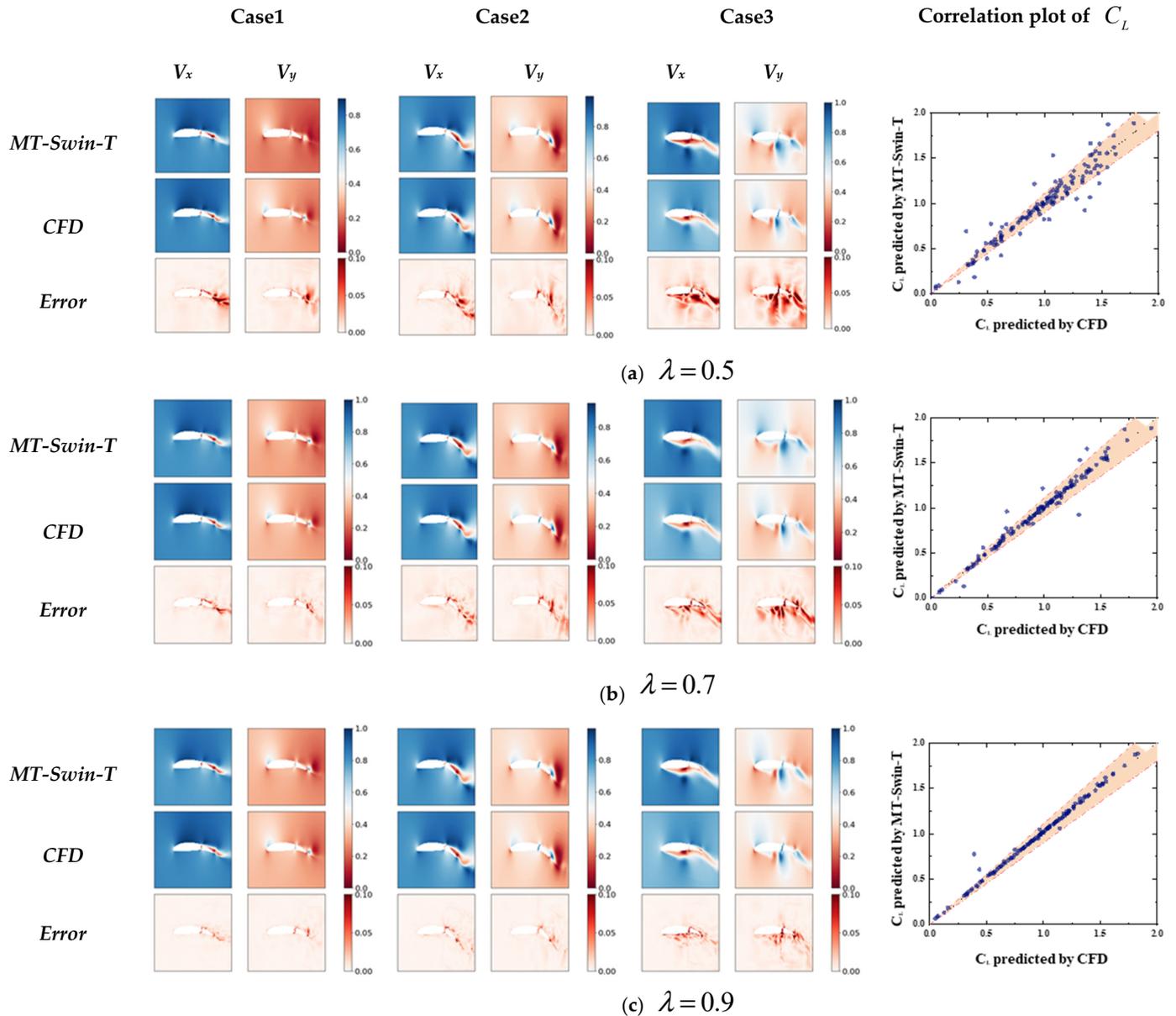


Figure 8. Comparison between MT-Swin-T and CFD prediction results. (Case 1: $Ma = 0.2146$, $AoA = -0.7811$ deg, $AoD = 6.6192$ deg; Case 2: $Ma = 0.3110$, $AoA = -5.2199$ deg, $AoD = 20.0104$ deg; Case 3: $Ma = 0.4747$, $AoA = -5.2199$ deg, $AoD = 21.2534$ deg).

As the value of λ increases, the weight of the loss function for the flow field prediction task is heightened, and the accuracy of both the flow field and aerodynamic force prediction results significantly improve. In the velocity field prediction, case 3 airfoil is at a negative AoA, leading to a substantial range of airflow separation occurring on the lower part of the airfoil. When $\lambda = 0.5$, large errors are observed in the V_x and V_y regions. However, when $\lambda = 0.7$ and $\lambda = 0.9$, the error margin is considerably reduced. The difference in the C_L prediction results is even more pronounced: when $\lambda = 0.5$, the prediction error for 20% of the cases is greater than 10%. At $\lambda = 0.7$, over 95% of the cases fall within the 10% error range, whereas at $\lambda = 0.9$, over 95% of the cases fall within the 5% error range.

Compared to the MLP network that predicts C_L , the Swin-T network that predicts flow fields has a more intricate structure and a larger parameter count substantially. During training, if the loss function coefficient λ is too small, the flow field task with a larger scale of network parameters will often dominate the shared model parameters. This domination would prevent the C_L prediction task, from having a smaller magnitude of gradient for acquiring effective shared representations. However, during the gradient updates of the shared parameters, the gradients produced by different tasks may oppose each other in direction, leading to an optimization cancellation. Consequently, this may prevent both nodes from being optimized. Therefore, when $\lambda = 0.5$, suboptimal prediction results can occur for both tasks.

3.2. Effect of Skip Connections

Skip connections effectively double the number of channels in each decoder block by linking all channels of the corresponding branches from the encoder to the decoder. These skip connections help the network in mitigating input information from the encoding layers when building flow field information in the decoding layers. Therefore, this section will examine the impact of the presence or absence of skip connections on the accuracy of the velocity field prediction results.

Referring to Figures 9 and 10, the effect of skip connections is compared based on the prediction results for cases 4 and 5, having significant differences in initial conditions. In case 4, near the main airfoil, the velocity gradient is small and no airflow separation exists; thus, there is not a large difference in the flow field prediction accuracy between MT-Swin-T with skip connections and MT-Swin-T without skip connections. However, near the deflected slat where the velocity gradient changes suddenly, and on the trailing edge, MT-Swin-T without skip connections exhibits significant errors in flow field prediction. In case 5, where the AoD is larger, the error exceeds ten percent of the prediction, covering most of the airfoil's wake area. Furthermore, the flow field predictions by MT-Swin-T without skip connections show a grid-like lack of smoothness. This is attributed to the absence of skip connections that share positional encoding information with the decoder. Consequently, MT-Swin-T without skip connections loses some transitional information between grid points during the encoding and decoding process, resulting in grid-like roughness and larger prediction errors.

Moreover, skip connections merge feature maps from lower and higher levels, retaining more spatial information to enhance prediction accuracy, alleviate the problem of gradient vanishing, and enhance the effectiveness of model training. Without skip connections, spatial information loss and the gradual disappearance of gradients during down-sampling operations in the backward propagation can occur, resulting in grid-like roughness and larger prediction errors.

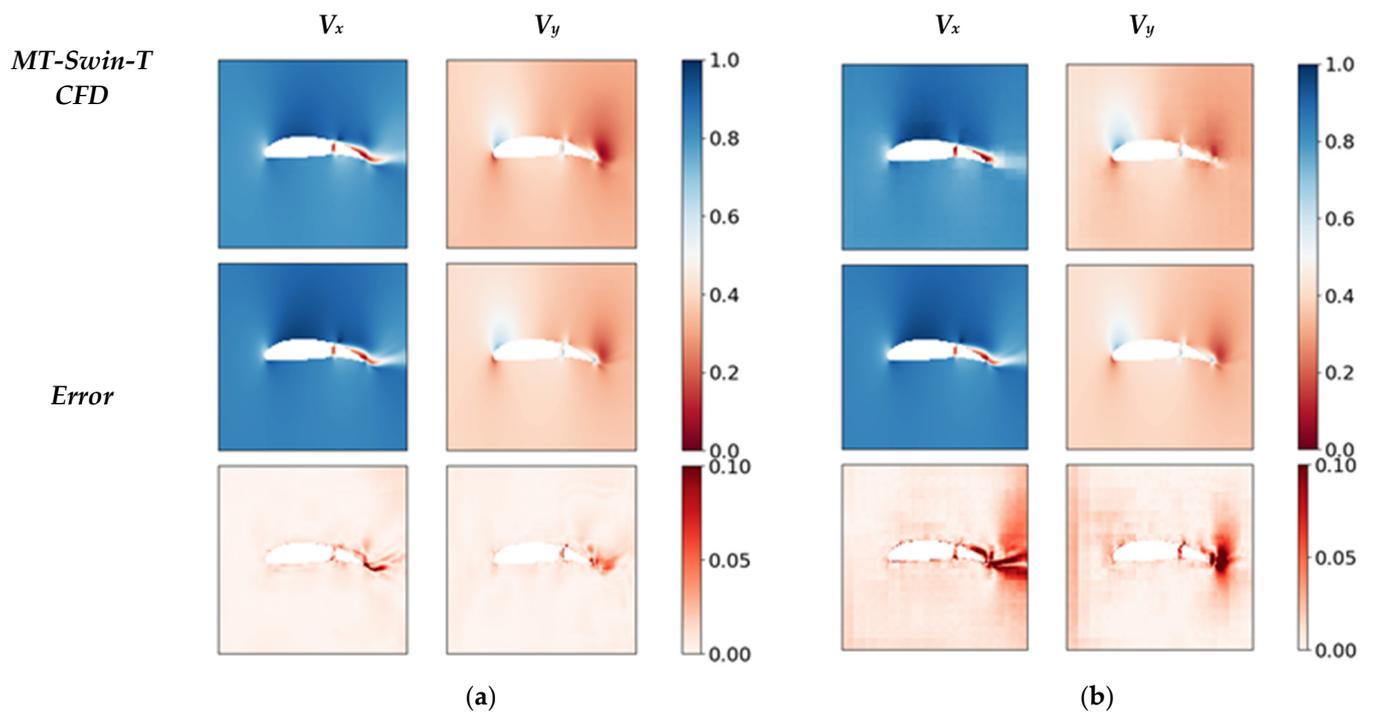


Figure 9. Comparison between MT-Swin-T and CFD prediction of flow field. (Case 4: $Ma = 0.2146$, $AoA = -0.7811$ deg, $AoD = 5.1069$ deg). (a) MT-Swin-T with skip connections; (b) MT-Swin-T without skip connections.

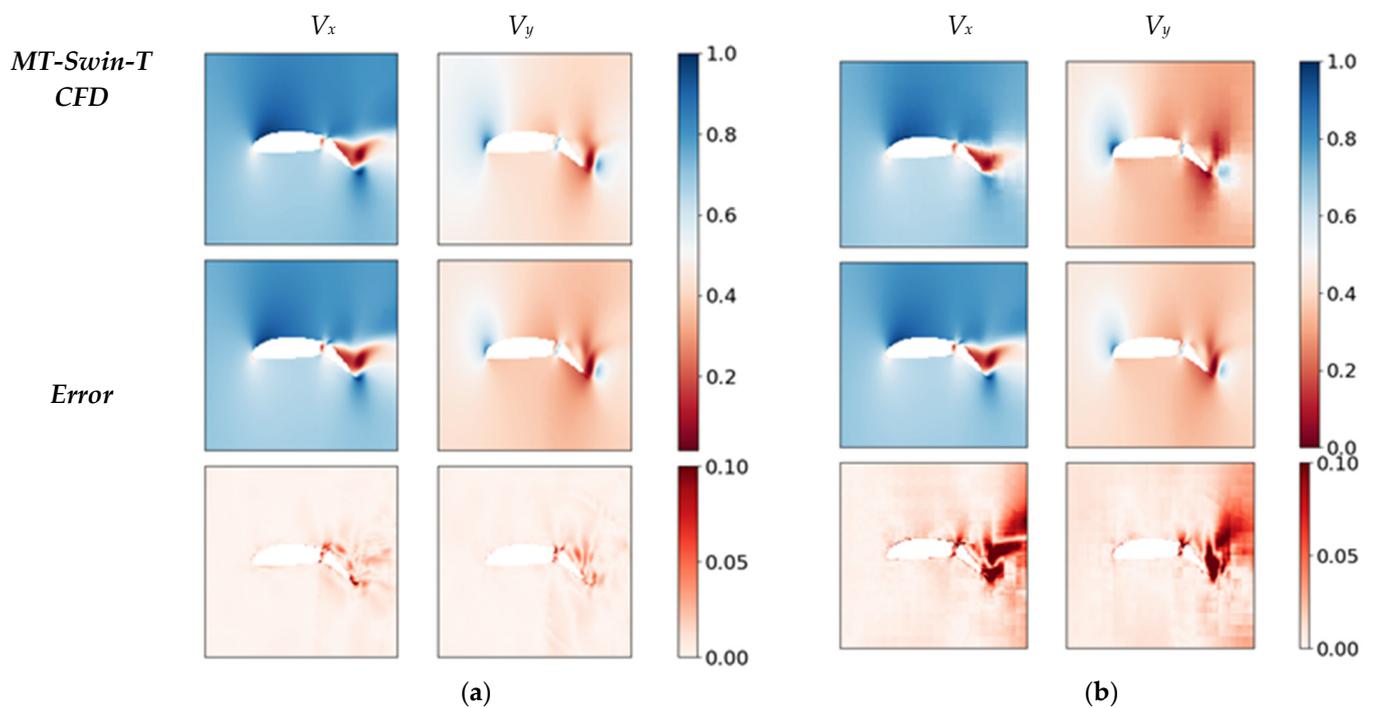


Figure 10. Comparison between MT-Swin-T and CFD prediction of flow field. (Case 5: $Ma = 0.311$, $AoA = 3.7646$ deg, $AoD = 20.7268$ deg). (a) MT-Swin-T with skip connections; (b) MT-Swin-T without skip connections.

3.3. Effect of Multi-Task Network Structure

To rapidly predict the flow field and lift coefficient of an airfoil with TEF, the method of hard parameter sharing is employed in the MT-Swin-T multi-task network structure. By comparing different modeling strategies, it is obvious that the proposed MT-Swin-T model has a clear advantage in prediction accuracy. The multi-task learning framework effectively uses the correlations between tasks, making it possible to simulate the flow field and aerodynamic performance of an airfoil with TEF using less training data.

Figure 11 illustrates a comparison between the CFD and MT-Swin-T predictions of the flow field for case 4. Applying the same network hyper-parameter settings and removing the MLP used for predicting the lift coefficient (C_L) from the Swin-T results in a slightly reduced overall accuracy in the velocity field prediction task compared to the full MT-Swin-T multitask deep learning network. In more detail, the errors in some grid points in the airflow separation region behind the slat exceeded 10%. Moreover, Figure 12 displays the correlation plot of C_L prediction results while deploying different network structures. By utilizing the MT-Swin-T multitask learning network combined with hard parameter sharing, it is possible to adjust the intermediate latent vector obtained from the encoder through the MLP task network, leading to more accurate initial condition and shape information contained in the intermediate latent vector. As a result, this will improve the accuracy of flow field prediction. Finally, a direct prediction of C_L using just the encoder and MLP combination results in larger prediction errors, with almost half of the cases exhibiting errors greater than 10%. Almost all C_L prediction errors using the complete MT-Swin-T network are less than 5%.

Apart from improving the accuracy of MLP prediction of the C_L by adjusting the intermediate latent vector obtained from the encoder through the Decoder task network, the full MT-Swin-T multitask network's increased number of parameters during training iterations improves the model's expressiveness, generalizability, and robustness; therefore, this will enhance the model's prediction performance.

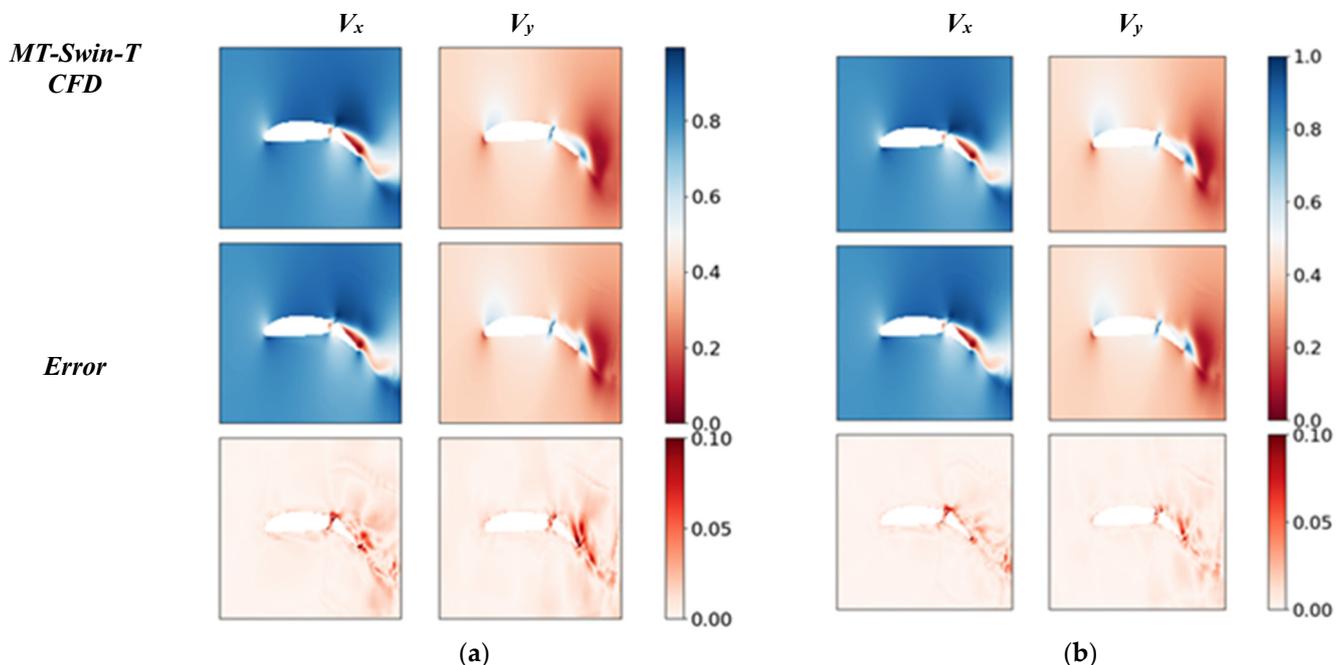


Figure 11. Comparison between CFD and MT-Swin-T prediction of flow field. (Case 4: $Ma = 0.2146$, $AoA = -0.7811$ deg, $AoD = 5.1069$ deg). (a) Prediction of Swin-T; (b) Prediction of MT-Swin-T.

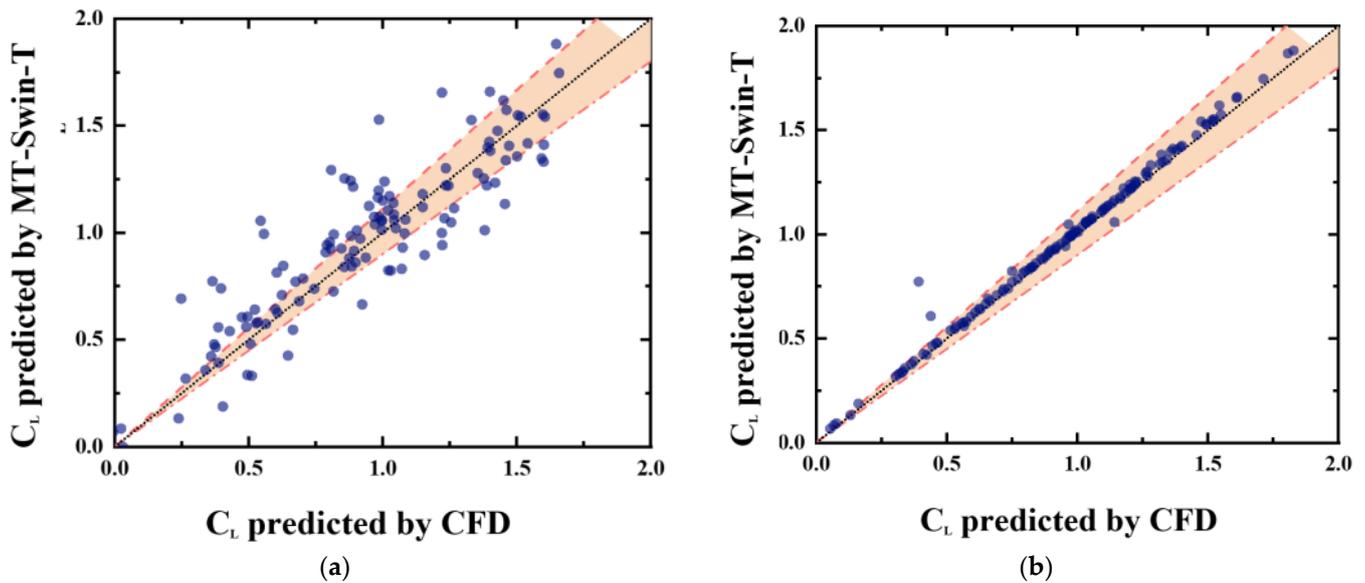


Figure 12. Comparison between CFD and MT-Swin-T prediction of C_L of different network architectures. (a) Encoder + MLP; (b) MT-Swin-T (Encoder + Decoder + MLP).

3.4. Comparison of Time-Consuming between Deep Learning and CFD

During the training process of the MT-Swin-T network, the initial learning rate was set to 2.5×10^{-4} , using a batch size of 128, and applying Adam optimizer to train the weights [29]. The number of epochs was set to 500. A learning rate scheduler was employed to adjust the learning rate in order to achieve optimal convergence of the model. Moreover, the learning rate decay parameter was set to 0.1, signifying that for every 100 epochs, the learning rate is multiplied by 0.1. Software and hardware configuration used for model training are described in Table 3, whereas the model code is implemented using the open-source deep learning library PyTorch [30]. In addition, as NVIDIA GeForce RTX 3090 GPU running on a Windows platform for training the neural network was used.

Table 3. Time cost comparison.

		Time Cost	Platform
CFD per case		632 s	CPU AMD 7452, 2 Cores used
CFD 43200 cases		293 h	CPU AMD 7452, 64 Cores
MT-Swin-T	test	0.0876 s per case	RTX 3090 GPU
	train	11.3 h	RTX 3090 GPU

Furthermore, a workstation equipped with an AMD Epyc 7452 64-Cores CPU with a base clock of 2.35 GHz was utilized to perform the CFD simulations for all datasets, where each case used two CPU cores, and 30 cases were running simultaneously. The average computation time per case was 532 s. Due to other resource requirements in the computer and the delay settings of batch run scripts, the total simulation time for 43,200 cases required more than 293 h. On the other hand, both the training and testing of the MT-Swin-T network model were carried out on a single NVIDIA RTX 3090 GPU. Training the model with 38,880 training samples and 4104 validation samples for 500 epochs required around 11.3 h, while the average computation time of 8.876 ms for testing a single case using the trained model. Therefore, it is evident that for a single case, the prediction time of the multitask deep learning method is approximately 1/7214 the time required for the CFD method's computation time. Considering the model training time, when comparing the total time consumed by both methods, the overall duration of the multitask deep learning method represents about 1/26 of that required by the CFD method.

4. Conclusions

This paper presents MT-Swin-T, a multi-task deep learning framework based on Swin Transformer, tailored for the simultaneous prediction of flow field information and lift coefficient across various airfoils equipped with TEF. In this network model, the encoder learns features related to airfoil shapes and computational conditions, while the decoder and MLP handle the prediction for flow field information and the lift coefficient, respectively. Leveraging a substantial dataset comprising up to 38,880 training samples and 4104 validation samples, the model demonstrates robust generalization capabilities. Compared to previous research, this work pioneers the application of deep learning methods and multi-task learning concepts to enhance the prediction of flow fields and aerodynamic forces for airfoils with TEF by enhancing the accuracy and efficiency of both predictive tasks. To sum up, the results show the following:

1. MT-Swin-T is capable of accurately predicting both the velocity field and the lift coefficient of airfoils with trailing edge flaps; however, the loss function combination adopted for different tasks significantly impacts the network's prediction accuracy. Within this network, the highest accuracy of predictions is obtained when the weight parameter is set to 0.9;
2. The skip connections in the MT-Swin-T network architecture facilitate the integration of feature maps from lower and higher levels, resulting in preserving more spatial information, mitigating the problem of gradient vanishing, and enhancing the prediction accuracy of the velocity field;
3. Compared to the solitary use of mitigated Encoder and MLP, the MT-Swin-T multi-task learning structure improves the accuracy of lift coefficient predictions; similarly, when being compared to Swin-T, the MT-Swin-T multi-task learning structure also yields a better accuracy of velocity field predictions;
4. In comparison to CFD simulations, the MT-Swin-T network model does not require high convergence criteria and time-consuming iterative processes. For an individual case analysis, the prediction time of MT-Swin-T is only 1/7214 of that required by the CFD method.

In summary, the MT-Swin-T model introduced in this study demonstrates interest in high-precision prediction of flow fields and aerodynamic forces associated with airfoils with TEF. Looking ahead, our research will expand the applicability of the network model proposed in this paper by adding data not included in the current parameter range based on the already trained model, using the method of transfer training. Furthermore, we target to propose novel deep learning techniques that incorporate physical constraints, therefore enhancing the interpretability of the flow field simulation framework.

Author Contributions: Conceptualization, C.Z.; methodology, Z.H.; programming, C.Z.; formal analysis, C.Z.; writing—original draft preparation, C.Z.; writing—review and editing, G.X. and Y.S.; visualization, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was co-supported by a project funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), National Key Laboratory Foundation (No. 6142202202 & RAL202203).

Data Availability Statement: All data generated or analyzed during this study are included in this article.

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

References

1. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [[CrossRef](#)]
2. Shafiq, M.; Gu, Z. Deep Residual Learning for Image Recognition: A Survey. *Appl. Sci.* **2022**, *12*, 8972. [[CrossRef](#)]
3. Min, B.; Ross, H.; Sulem, E.; Veyseh, A.P.B.; Nguyen, T.H.; Sainz, O.; Agirre, E.; Heintz, I.; Roth, D. Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey. *ACM Comput. Surv.* **2023**, *56*, 1–40. [[CrossRef](#)]
4. Ribeiro, M.D.; Rehman, A.; Ahmed, S.; Dengel, A. DeepCFD: Efficient Steady-State Laminar Flow Approximation with Deep Convolutional Neural Networks. *arXiv* **2020**, arXiv:2004.08826.
5. Guo, X.; Li, W.; Iorio, F. Convolutional Neural Networks for Steady Flow Approximation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 481–490.
6. Sekar, V.; Khoo, B.C. Fast Flow Field Prediction over Airfoils Using Deep Learning Approach. *Phys. Fluids* **2019**, *31*, 057103. [[CrossRef](#)]
7. Thuerey, N.; Weissenow, K.; Prantl, L.; Hu, X. Deep Learning Methods for Reynolds-Averaged Navier-Stokes Simulations of Airfoil Flows. *AIAA J.* **2020**, *58*, 25–36. [[CrossRef](#)]
8. Tangsali, K.; Krishnamurthy, V.R.; Hasnain, Z. Generalizability of Convolutional Encoder–Decoder Networks for Aerodynamic Flow-Field Prediction Across Geometric and Physical-Fluidic Variations. *J. Mech. Des.* **2021**, *143*, 051704. [[CrossRef](#)]
9. Chen, L.-W.; Thuerey, N. Towards High-Accuracy Deep Learning Inference of Compressible Flows over Aerofoils. *Comput. Fluids* **2022**, *250*, 105707. [[CrossRef](#)]
10. Wu, H.; Liu, X.; An, W.; Lyu, H. A Generative Deep Learning Framework for Airfoil Flow Field Prediction with Sparse Data. *Chin. J. Aeronaut.* **2021**, *35*, 470–484. [[CrossRef](#)]
11. Duru, C.; Alemdar, H.; Baran, Ö.U. CNNFOIL: Convolutional Encoder Decoder Modeling for Pressure Fields around Airfoils. *Neural Comput. Appl.* **2021**, *33*, 6835–6849. [[CrossRef](#)]
12. Gupta, R.; Jaiman, R. A Hybrid Partitioned Deep Learning Methodology for Moving Interface and Fluid–Structure Interaction. *Comput. Fluids* **2021**, *233*, 105239. [[CrossRef](#)]
13. Hao, L.; Gao, Y.; Wei, B.; Song, K. Numerical Simulation of Flow over Bionic Airfoil. *Int. J. Aerosp. Eng.* **2021**, *2021*, 5556463. [[CrossRef](#)]
14. Kamliya Jawahar, H.; Azarpeyvand, M.; Ilário Da Silva, C.R. Acoustic and Flow Characteristics of an Airfoil Fitted with Morphed Trailing Edges. *Exp. Therm. Fluid Sci.* **2021**, *123*, 110287. [[CrossRef](#)]
15. Ye, J.; Salem, S.; Wang, J.; Wang, Y.; Du, Z.; Wang, W. Effects of Micro-Tab on the Lift Enhancement of Airfoil S-809 with Trailing-Edge Flap. *Processes* **2021**, *9*, 547. [[CrossRef](#)]
16. He, G.; Deparday, J.; Siegel, L.; Henning, A.; Mulleners, K. Stall Delay and Leading-Edge Suction for a Pitching Airfoil with Trailing-Edge Flap. *AIAA J.* **2020**, *58*, 5146–5155. [[CrossRef](#)]
17. Yue, Z.; Ye, F.; Zhang, Y.; Liang, C.; Tsang, I.W. Deep Safe Multi-Task Learning. *arXiv* **2021**, arXiv:2111.10601.
18. Hu, R.; Singh, A. UniT: Multimodal Multitask Learning with a Unified Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1439–1449.
19. Salehi Rizi, F.; Granitzer, M. Multi-Task Network Embedding with Adaptive Loss Weighting. In Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Hague, The Netherlands, 7–10 December 2020; pp. 1–5.
20. Chen, H. Multi-Resolution Multi-Task Network and Polyp Tracking. In *Computer-Aided Analysis of Gastrointestinal Videos*; Bernal, J., Histace, A., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 81–89, ISBN 978-3-030-64340-9.
21. Popescu, M.-C.; Balas, V.E.; Perescu-Popescu, L.; Mastorakis, N. Multilayer Perceptron and Neural Networks. *WSEAS Trans. Circuits Syst.* **2009**, *8*, 579–588.
22. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
23. Selig, M.S. *UIUC Airfoil Data Site*; Department of Aeronautical and Astronautical Engineering, University of Illinois at Urbana-Champaign: Urbana, IL, USA, 1996.
24. Shields, M.D.; Zhang, J. The Generalization of Latin Hypercube Sampling. *Reliab. Eng. Syst. Saf.* **2016**, *148*, 96–108. [[CrossRef](#)]
25. Hu, Z.; Xu, G.; Shi, Y. A New Study on the Gap Effect of an Airfoil with Active Flap Control Based on the Overset Grid Method. *Int. J. Aeronaut. Space Sci.* **2021**, *22*, 779–801. [[CrossRef](#)]
26. Hassan, A.A.; Straub, F.K.; Noonan, K.W. Experimental/Numerical Evaluation of Integral Trailing Edge Flaps for Helicopter Rotor Applications. *J. Am. Helicopter. Soc.* **2005**, *50*, 3–17. [[CrossRef](#)]
27. Park, J.J.; Florence, P.; Straub, J.; Newcombe, R.; Lovegrove, S. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 165–174.
28. Siddique, N.; Paheding, S.; Elkin, C.P.; Devabhaktuni, V. U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. *IEEE Access* **2021**, *9*, 82031–82057. [[CrossRef](#)]

29. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
30. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimeshain, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2019; pp. 8024–8035.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.