Mechanical and Aerospace Engineering
B. Guan (Ed.)

doi:10.3233/ATDE250018

Inverse Design of Transonic Airfoils Using Denoising Diffusion Probabilistic Models

Guocheng TAO [a], Yang LIU [a] and Yan YAN [b,1]
[a] Zhejiang University, Hangzhou, Zhejiang, 310058, China
[b] Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, 215123, China

Abstract. Inverse design offers significant advantages in aerodynamic design, such as improved performance and efficiency. In this paper, a denoising diffusion probabilistic model (DDPM) is adopted as a generative model to produce pressure coefficient distribution data for the RAE2822 airfoil. Through the forward noise addition process and the reverse denoising process, the trained DDPM model can sample a large amount of pressure coefficient distribution data from a standard normal distribution. Two neural networks are then employed: one maps the pressure coefficient distribution to geometric parameters, linking the pressure field with geometric parameters, and the other maps the pressure coefficients to lift and drag coefficients. Computational fluid dynamics (CFD) validation of the sampled data shows that the CFD results are close to the generated pressure distributions, demonstrating the effectiveness and reliability of the proposed approach.

Keywords. Transonic airfoils, denoising diffusion probabilistic models, inverse design

1. Introduction

Surrogate models are essential in airfoil design as they provide a computationally efficient means to approximate complex aerodynamic performance metrics. These models enable rapid evaluations of airfoil shapes by capturing the intricate relationships between geometric parameters and aerodynamic performance without the need for extensive and time-consuming simulations. By using surrogate models, engineers can explore a vast design space, identify optimal configurations, and perform sensitivity analyses with reduced computational resources. This approach not only accelerates the design process but also enhances the accuracy and reliability of the results, ultimately leading to the development of high-performance airfoils that meet stringent aerodynamic requirements. Du et al. developed a convolutional neural network framework (DPCNN) for airfoil design and performance prediction, achieving high prediction accuracy, robustness, and fast computation speed, significantly improving the efficiency of airfoil design optimization [1]. Mufti et al. introduced DIP-ShockNet, a domain-informed probabilistic deep learning framework, to predict transonic flow fields with shock waves, achieving superior accuracy and uncertainty estimation compared to traditional methods [2]. Tian et al. developed a novel pressure-based optimization (PBO) method using deep learning techniques, demonstrating significant improvements in airfoil design accuracy and drag reduction compared to traditional methods [3].

[1] Corresponding Author: Yan Yan, yan.yan@xjtlu.edu.cn.

Inverse design plays a crucial role in airfoil design by enabling the optimization of aerodynamic performance through the systematic adjustment of geometric parameters. Instead of relying on traditional trial-and-error methods, inverse design techniques allow engineers to specify desired performance characteristics, such as lift, drag, or pressure distribution, and then compute the corresponding airfoil shape. This approach significantly accelerates the design process, improves accuracy, and ensures that the resulting airfoils meet specific performance targets. By leveraging advanced algorithms and computational models, inverse design enhances the efficiency and effectiveness of airfoil development, leading to better-performing and more efficient aerodynamic structures. Wang proposed a novel inverse design method for supercritical airfoils using generative models in deep learning, employing CVAE and CVAE-GAN models to generate target wall Mach distributions [4].

Denoising Diffusion Probabilistic Models (DDPM) have emerged as powerful generative models due to their ability to produce high-quality samples by systematically reversing a forward diffusion process. These models leverage a series of noise additions followed by learned denoising steps, resulting in superior sample fidelity and diversity compared to traditional generative models. In this work, we apply DDPM to the generation of pressure coefficient distributions on airfoil surfaces, demonstrating its effectiveness in capturing the complex aerodynamic characteristics necessary for accurate airfoil design.

The structure of this paper is as follows: The validation of the numerical calculations is presented in Sec. 2. Sec. 3 introduced the inverse design framework. The results of the inverse design are analysed in Sec. 4. Finally, Sec. 5 provides a summary of the conclusions.

2. Validation

2.1. Parameterization

In this work, the CST method [5,6] is applied to parameterize the RAE2822 airfoil. This method consists of shape and class functions. The class function is used to define the general classes of geometry, whereas the shape function is used to define specific shapes within the geometry class. The order of Bernstein polynomials of 5 is selected for the CST parameterization, corresponding to six parameters for each of the upper and lower airfoils. The comparison of RAE2822 airfoil CST fitting with the original geometry is shown in Figure 1. The geometry of the CST fitting fits the original geometry well. All of the above indicate that the CST method is efficient enough to parameterize the RAE2822 airfoil.

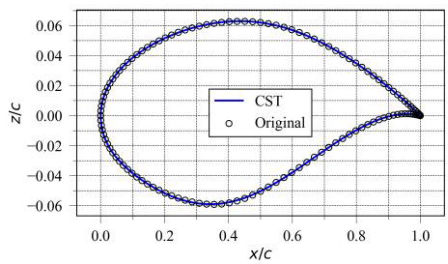

Figure 1. Comparison of RAE2822 airfoil CST fitting with the original geometry.

2.2. Numerical Calculation

The Spalart-Allmaras (SA) turbulence model, combined with the implicit, density-based Reynolds-Averaged Navier-Stokes (RANS) model, was employed in the computations using the commercial software Ansys Fluent. In this study, the Roe flux-difference splitting (Roe-FDS) method was utilized for space discretization. Flow-field gradients were computed using a second-order accurate upwind spatial discretization with a Green-Gauss node-based scheme, and dynamic viscosity was assumed to be constant.

The grids employed featured a hybrid unstructured topology, with a wall normal growth of boundary layer cells. The initial grid spacing normal to the airfoil was set as 4×106 to ensure the overall $y^+ < 1$ for both high-fidelity grid and low-fidelity grid. The grid growth rate was 1.1, Figure 2 shows the computational grid around RAE 2822 airfoil. The adopted grid size is approximately 64893. The operating conditions are as follows: Mach number is 0.73, Reynolds number is 6.5×106, angle of attack is 2.79° and freestream temperature is 300K [7,8].

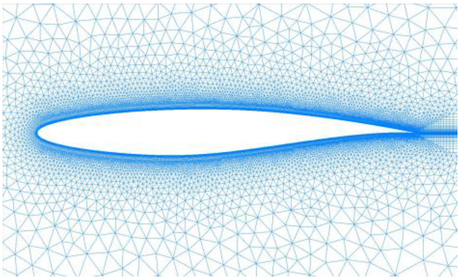

Figure 2. Sketch of grids of RAE2822 airfoil.

Figure 3 shows the computational pressure coefficient distributions of RAE2822 airfoil compared with experimental ones. Good agreement with the experiment results can be observed, which further proves the accuracy of the above CFD model. And the pressure coefficient contours shown in Figure 4.

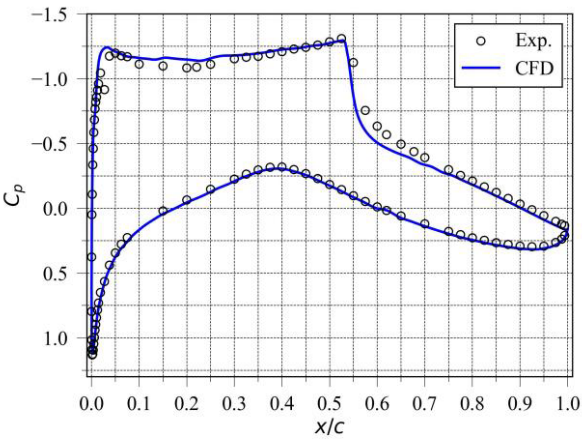

Figure 3. Comparison of RAE2822 airfoil pressure coefficient distribution for CFD results and experimental results.

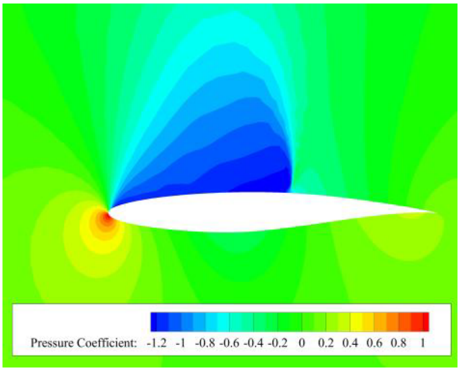

Figure 4. Pressure coefficient contour for RAE2822 airfoil.

3. Inverse Design Framework

3.1. Unconditional Generation of Pressure Coefficient

Jonathan Ho proposed the Denoising Diffusion Probabilistic Models (DDPM) in 2020 [9]. This method has significantly advanced generative modeling techniques, offering a robust framework for high-quality sample generation. DDPM leverage a Markov chain to generate data by iteratively adding Gaussian noise to training samples over a series of timesteps. During the training process, the model learns to reverse this noise-adding process, effectively denoising the samples step-by-step. This involves training the model to predict the noise at each step, allowing it to reconstruct the original data from a noisy version. This diffusion process creates a gradual path from a complex data sample to Gaussian noise, thereby simplifying the reverse denoising task. It does this by breaking the process into multiple intermediate steps, as illustrated in Figure 5.

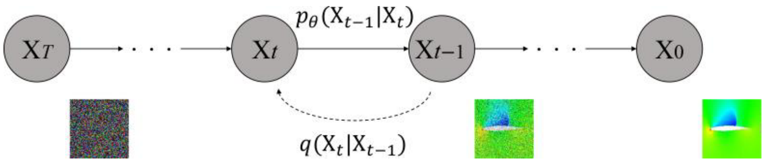

Figure 5. The diffusion process and its reverse for the pressure coefficient contour of RAE2822 airfoil.

The training and sampling processes of DDPM are shown as Algorithm 1 and Algorithm 2, respectively [7]. Algorithm 1 describes the training procedure for DDPM. In this algorithm, the model is trained by repeatedly sampling data points x_0 from the data distribution, selecting a random timestep t, and generating Gaussian noise ϵ. The training objective is to minimize the mean squared error between the added noise and the model's predicted noise for the given timestep. This process iteratively adjusts the model parameters to learn how to denoise the data at each timestep. Algorithm 2 outlines the sampling procedure, where the trained model generates new data samples. Starting with Gaussian noise x_T, the algorithm progressively denoises the sample by reversing the diffusion process. At each timestep t, the sample x_{t-1} is computed using the model's denoising function and adjusted for the timestep's parameters. If $t > 1$, additional

Gaussian noise z is incorporated. This iterative process continues until the model reconstructs the original data sample x_0 from the noise.

Algorithm 1 Training	**Algorithm 2** Sampling
1: **repeat**	1: $x_T \sim N(\mathbf{0}, \mathbf{I})$
2: $x_0 \sim q(x_0)$	2: **for** $t = T, \ldots, 1$ **do**
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$	3: $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\epsilon \sim N(\mathbf{0}, \mathbf{I})$	4: $x_{t\text{-}1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1\text{-}\alpha_t}{\sqrt{1\text{-}\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t\mathbf{z}$
5: Take gradient descent step on $\nabla_\theta \left\lVert \epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\rVert^2$	5: **end for**
6: **until** converged	6: **return** x_0

3.2. UNet Architecture

DDPM often leverage U-Net architectures due to their effectiveness in capturing and reconstructing complex data structures through a series of downsampling and upsampling layers. The U-Net architecture used in DDPM typically includes residual blocks, self-attention mechanisms, and various convolutional operations to process and generate high-quality samples.

As illustrated in Figure 6, the U-Net begins with a 1D convolution layer (kernel size 7, stride 1, padding 3) processing the initial input of size 1×128. The downsampling path follows, consisting of several residual blocks, each paired with a downsampling layer using a 1D convolution (kernel size 4, stride 2, padding 1). These blocks, some of which incorporate self-attention mechanisms, reduce the feature map dimensions progressively from 64×128 to 256×16 while extracting rich features. At the bottleneck, feature maps are further processed through residual blocks, ensuring the preservation and effective utilization of input information. The upsampling path mirrors the downsampling path, using upsampling operations followed by residual blocks and concatenation with corresponding downsampling features, progressively increasing dimensions back to 1×128. The output, after passing through convolution and residual blocks, matches the target dimensions.

3.3. Residual Block Architecture

The Residual block is a residual module designed for one-dimensional convolutional networks. As shown in Figure 7, it incorporates multiple convolutional layers, group normalization, activation functions, a time embedding projection, and an optional attention mechanism. The module starts with a group normalization and a Swish activation function, followed by a convolutional layer that adjusts the input to the required dimensions. A time embedding projection is added to the output of this convolutional block. The second part of the module includes another group normalization, a Swish activation function, a dropout layer, and a convolutional layer. To ensure compatibility of dimensions for the residual connection, a shortcut path is used, which either applies a 1×1 convolution or an identity mapping, depending on the input and output dimensions. If attention is enabled, an attention block is applied at the end. The primary function of this block is to stabilize the training process and extract deeper features through residual connections.

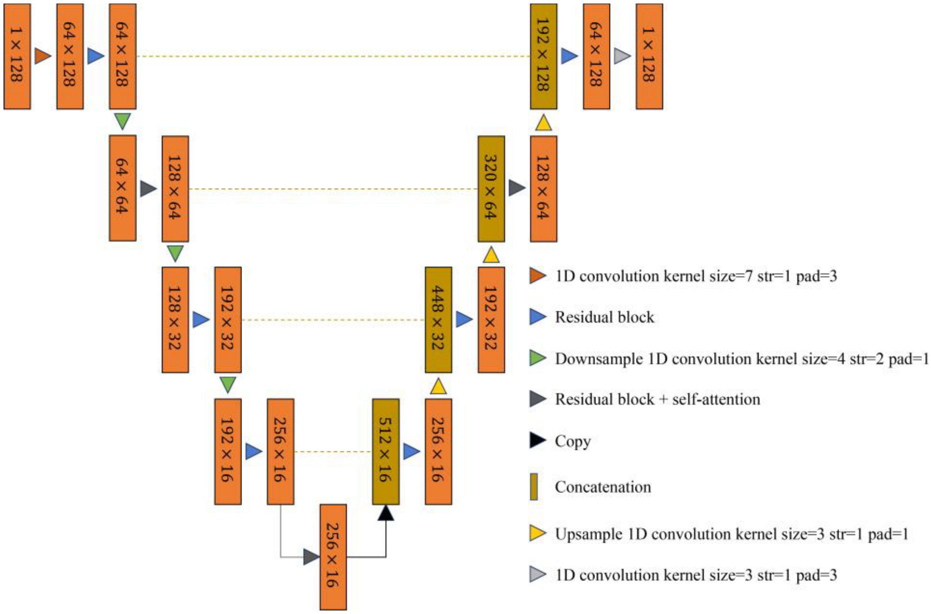

Figure 6. UNet architecture.

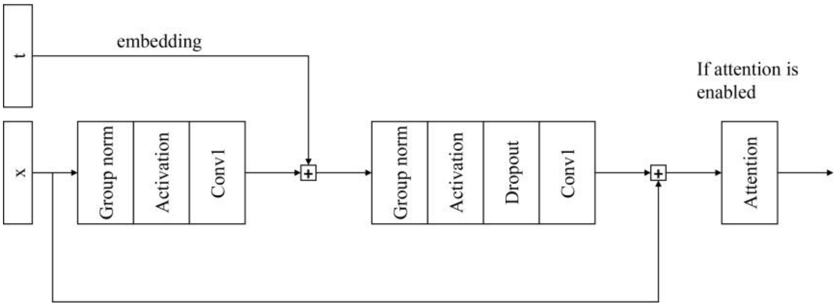

Figure 7. Residual block architecture.

3.4. Pressure Distribution-geometry Neural Network Model

The neural network is designed to predict geometric parameters from the pressure coefficients generated by DDPM. It utilizes fully connected layers, each incorporating linear layers, batch normalization, ReLU activation function, and dropout. This architecture enables efficient feature extraction and stable training, allowing for accurate mapping from pressure coefficients to geometric parameters.

3.5. Pressure Distribution-aerodynamic Performance Neural Network Model

The neural network is designed to predict aerodynamic performance from the pressure coefficients generated by DDPM. The architecture of the neural network is consistent with that of the Pressure distribution-geometry neural network model.

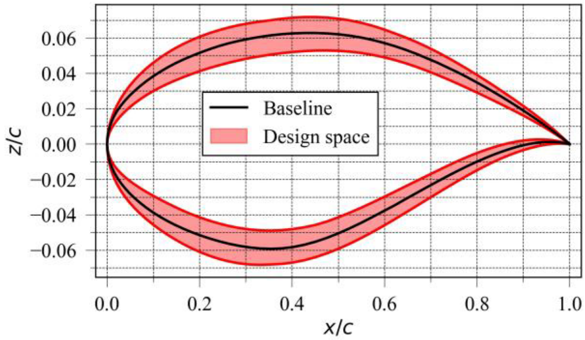

Figure 8. Design space.

4. Results and Discussion

In this paper, 2400 airfoil samples were generated using the Latin hypercube sampling (LHS) method. The design space of airfoil is shown in Figure 8. After CFD calculations, the pressure coefficient distribution data for each airfoil surface were extracted. A total of 128 points were sampled on the upper and lower surfaces, which were subsequently used as input for the DDPM model.

4.1. Training Process

In this study, the parameters for the DDPM model were set as follows: The initial beta value, β_1, was set to 1×10^{-4}, and the final beta value, β_T, was set to 0.02, with a total of T=1000 timesteps. These parameters were chosen to ensure a smooth diffusion process from the original data to Gaussian noise, facilitating the training of the reverse denoising model.

For neural networks, we utilized the AdamW optimizer with a weight decay of 1e-4 for training the model. The initial learning rate was set to 0.0001 for the first 500 steps to ensure stable training. After 500 steps, the learning rate was increased to 0.1 to expedite convergence. The batch size was set to 32 to efficiently utilize computational resources while maintaining stability during training. The total training duration was 1000 steps. MSE is used as the loss function. The 128-dimensional pressure coefficient data was first reduced using PCA. The cumulative energy of the PCA as a function of the number of principal components is shown in Figure 9. When the number of principal components reached 19, the cumulative energy exceeded 0.99. Therefore, we selected 20 principal components. The network consisted of five hidden layers, with the number of neurons in each layer being 256, 256, 512, 256, and 256. The model was trained on a single NVIDIA Tesla GPU. The training time for the pressure distribution-geometry neural network model was approximately 17 minutes, while the pressure distribution-aerodynamic performance neural network model took around 16 minutes. The DDPM model required about 281 minutes to train. CFD simulations were performed on 28 CPUs, with each CFD case taking approximately 5 minutes to complete. Although the model training time is longer compared to a single CFD simulation, once training is completed, predictions can be generated in less than one second. This significant reduction in computation time offers substantial savings for subsequent analysis or optimization processes.

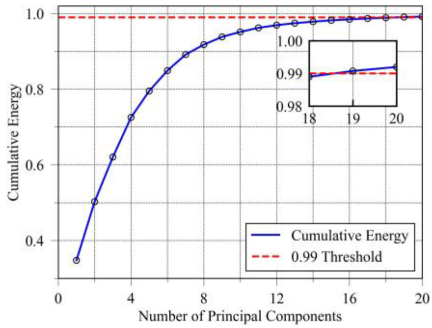

Figure 9. Cumulative Energy vs. Number of Principal Components.

After training, 16 pressure coefficient distributions were generated through random sampling using the DDPM. These generated pressure distributions were then mapped to geometric parameters using a specialized pressure-to-geometry mapping network. Subsequently, the obtained geometric parameters were fed into CFD solver to produce the corresponding real pressure coefficient distributions. As illustrated in Figure 10, the generated pressure distributions closely match the real pressure distributions derived from CFD simulations. This agreement indicates the effectiveness and reliability of the entire inverse design framework. The results demonstrate the potential of this method to accurately predict and generate pressure coefficient distributions, thereby establishing a robust foundation for future inverse aerodynamic optimization design.

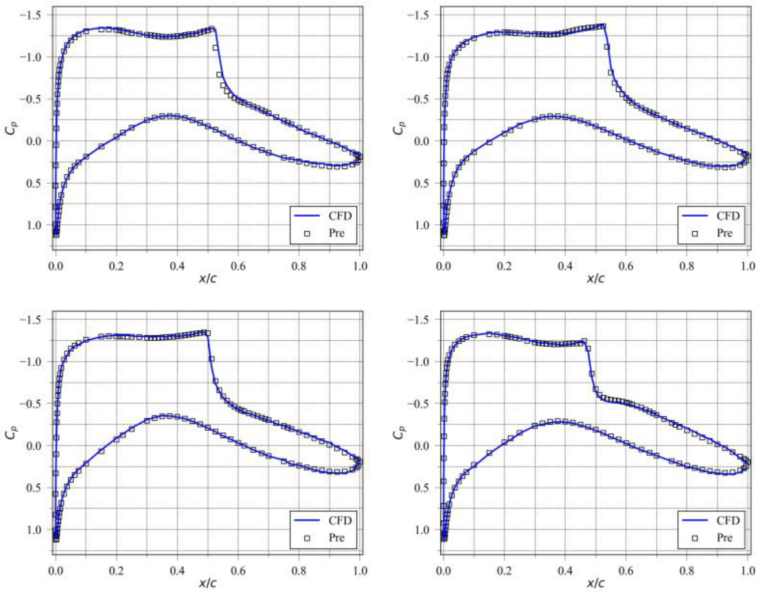

Figure 10. Comparison of the true and predicted pressure coefficient distributions for the generated samples.

Additionally, Figure 11 compares the predicted and real lift coefficients and drag coefficients obtained from the CFD solver. The plots show a strong correlation between the predicted and real values, with R^2 values of 0.946 and 0.956, demonstrating the high accuracy of the predictive model.

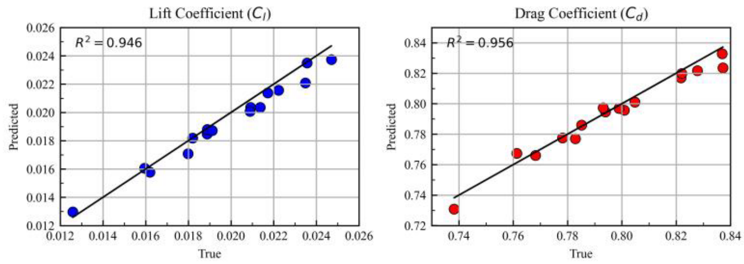

Figure 11. Comparison of Predicted and Real Lift and Drag Coefficients for the generated samples.

4.2. Effects of the Training Set Size

This section studies the impact of the number of training samples on the performance of the DDPM model. Since the focus is on the DDPM model, the Pressure distribution-geometry and Pressure distribution-aerodynamic performance models still use those trained with 2400 samples, while the DDPM is trained with 1200 samples.

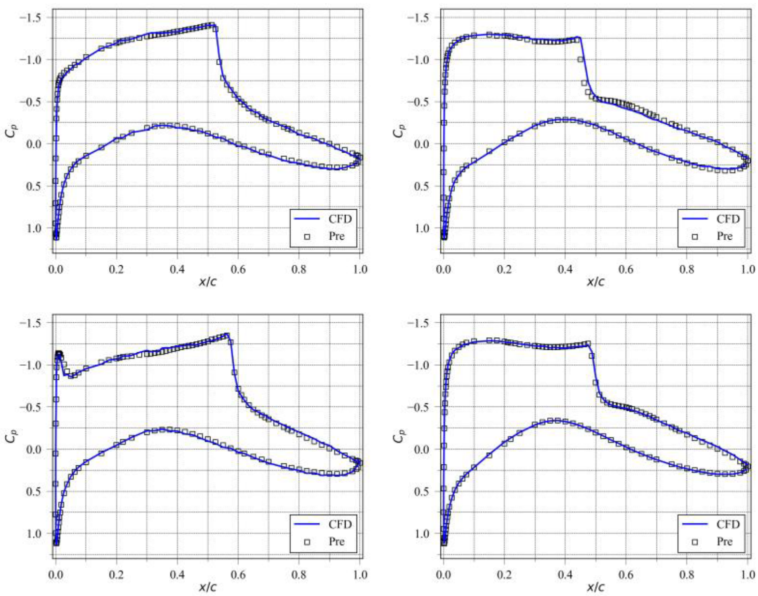

Figure 12. Comparison of the true and predicted pressure coefficient distributions for the generated samples with 1200 training samples.

With the DDPM model trained on only 1200 samples, the generated pressure coefficient distributions were transformed into geometric parameters using the previously trained pressure-to-geometry network, and then the resulting geometries were input into a CFD solver to produce real pressure distributions. Figure 12 shows that the generated distributions closely resemble the real distributions, albeit with a slight decrease in precision compared to the results from 2400 samples.

In Figure 13, the comparison between predicted and real lift and drag coefficients is presented, with R^2 values of 0.836 and 0.873, respectively. These values indicate a strong correlation, though they are somewhat lower than the values obtained using the larger training set. This highlights the model's robustness and effectiveness even with fewer training samples.

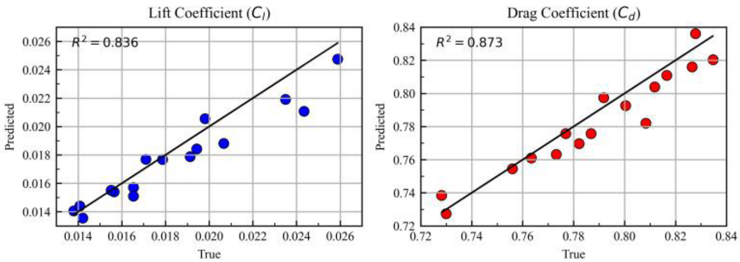

Figure 13. Comparison of Predicted and Real Lift and Drag Coefficients for the generated samples with 1200 training samples.

5. Conclusion

This paper investigates the application of inverse design in airfoil development, focusing on the use of DDPM to generate pressure coefficient distribution data. Two neural networks are employed: one maps these pressure coefficients to geometric parameters, and the other maps the pressure coefficients to lift and drag coefficients. The generated samples are validated through CFD simulations, demonstrating the effectiveness and feasibility of this approach. This study establishes a robust foundation for future inverse aerodynamic optimization design, significantly enhancing the efficiency of the design process and ensuring that the generated airfoil shapes meet desired aerodynamic performance criteria.

References

[1] Q. Du, T. Liu, L. Yang, L. Li, D. Zhang, Y. Xie, Airfoil design and surrogate modeling for performance prediction based on deep learning method, Phys. Fluids. 34 (2022) 15111.

[2] B. Mufti, A. Bhaduri, S. Ghosh, L. Wang, D.N. Mavris, Shock wave prediction in transonic flow fields using domain-informed probabilistic deep learning, Phys. Fluids. 36 (2024) 16121.

[3] J. Tian, F. Qu, D. Sun, Q. Wang, Novel Pressure-Based Optimization Method Using Deep Learning Techniques, AIAA J. 62 (2024) 708–724. https://doi.org/10.2514/1.J063222.

[4] J. WANG, R. LI, C. HE, H. CHEN, R. CHENG, C. ZHAI, M. ZHANG, An inverse design method for supercritical airfoil based on conditional generative models, Chinese J. Aeronaut. 35 (2022) 62–74.

[5] B. Kulfan, J. Bussoletti, "Fundamental" Parameteric Geometry Representations for Aircraft Component Shapes, in: 11th AIAA/ISSMO Multidiscip. Anal. Optim. Conf., 2006: p. 6948.

[6] B.M. Kulfan, Universal parametric geometry representation method, J. Aircr. 45 (2008) 142–158.

[7] P.H. Cook, M.A. McDonald, M.C.P. Firmin, Aerofoil rae 2822-pressure distributions, and boundary layer and wake measurements. experimental data base for computer program assessment, AGARD Rep. AR. 138 (1979) 47.

[8] W. Haase, F. Bradsma, E. Elsholz, M.A. Leschziner, D. Schwamborn, EUROVAL–An European Initiative on Validation of CFD Codes. Notes on Numerical Fluid Mechanics, Vol. 42, Vieweg, Braunschweig/Wiebad. (1993).

[9] J. Ho, A. Jain, P. Abbeel, Denoising Diffusion Probabilistic Models, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Adv. Neural Inf. Process. Syst., Curran Associates, Inc., 2020: pp. 6840–6851.