Electronic Research Archive

ERA, 33(5): 3271–3284.
DOI: 10.3934/era.2025144
Received: 24 February 2025
Revised: 15 May 2025
Accepted: 20 May 2025
Published: 26 May 2025

http://www.aimspress.com/journal/ERA

Research article

Generative adversarial network for inverse design of airfoils with flow control devices

Alejandro Ballesteros-Coll[1], Koldo Portal-Porras[2], Unai Fernandez-Gamiz[1,*], Iñigo Aramendia[2] and Daniel Teso-Fz-Betoño[2]

[1] Department of Energy Engineering, University of the Basque Country (UPV/ EHU), Nieves Cano, 12, Vitoria-Gasteiz 01006, Spain
[2] Department of Electrical Engineering, University of the Basque Country (UPV/EHU), Nieves Cano, 12, Vitoria-Gasteiz 01006, Spain

* **Correspondence:** Email: unai.fernandez@ehu.eus.

Abstract: Deep learning has recently gained prominence in fluid dynamics due to advances in computational power, algorithm development, and data availability. While most applications have focused on modeling and control, its potential for design and optimization remains relatively unexplored. In this study, a conditional generative adversarial network (CGAN) was developed for inverse design of airfoils with flow control devices. This CGAN receives the characteristics of the flow (Reynolds number and angle of attack) and the desired aerodynamic characteristics (drag and lift coefficients). Based on those inputs, the CGAN generates an airfoil that fulfills the defined specifications. With this objective, numerical simulations of 4-digit NACA airfoils with variable flap configurations and flow conditions were conducted, in order to obtain the necessary aerodynamic data for training and testing the CGAN. The results demonstrate that the proposed CGAN generates airfoil geometries with high accuracy and efficiency, showing minor deviations from real geometries and achieving aerodynamic performances that approach the desired ones for all the flow conditions considered. Additionally, the model is able to generalize to extreme cases not seen during training, which significantly broadens its application range. This approach offers a significant reduction in computational time compared to traditional iterative optimization methods, making it suitable for rapid design exploration and real-time applications.

Keywords: flow control devices; inverse design; generative adversarial network; computational fluid dynamics; deep learning

1. Introduction

In recent years, an exponential growth in the use of deep learning–based methods has occurred. This is attributable to various technological and scientific advances, including increased computational capacity, the maturation of algorithms, and the availability of massive datasets for model training. As a result, deep learning has emerged as a powerful tool across multiple fields and sectors.

In the domain of fluid dynamics, numerous authors have employed these methods to address various tasks related to the modeling, control, and design of fluid dynamic systems. Most research of deep learning applied to fluid dynamics focuses on system modeling, demonstrating the validity of these methods for a wide variety of systems, such as airfoils [1–3], flow control devices [4,5], vehicles [6,7], thermal systems [8], chemical reactions [9], and many others. Beyond modeling, multiple authors [10–14] have leveraged these methods for aerodynamic performance enhancement through control strategies; many others, such as Lee et al. [15], Wang et al. [16], and Mehrjardi et al. [17,18], have focused their efforts on the design of fluid dynamic systems, aiming to optimize the performance of these systems. However, the application of deep learning to system design remains a relatively unexplored and evolving area of research.

Within the field of fluid dynamic system design, inverse aerodynamic design is particularly noteworthy. Inverse aerodynamic design consists of determining the shape of the geometry based on information about the desired aerodynamic behavior. As explained in Yilmaz and German [19], traditional approaches rely on potential flow and surface pressure distribution to infer geometry. In contrast, modern techniques aim to generate designs based on broader performance metrics and more complex flow characteristics. These methods typically adopt an optimization-based approach, wherein a cost function is defined to optimize one or more performance measures, subject to constraints that account for the physical and practical limitations of the system under study.

Inverse aerodynamic design presents numerous challenges, including the following [19]:

- The representation and parameterization of the surface shape through discretization or basis functions that allow sufficient freedom to develop good designs, while avoiding poorly thought-out designs.
- The choice of suitable cost functions that strike the right balance between the objectives of the designer, which are often contradictory and require trade-offs.
- The identification of all constraints necessary to achieve realistic solutions, including those tacit to human experience and difficult to codify, to avoid the tendency of optimizers to exploit weaknesses in the problem formulation rather than achieve realistic results.
- The detection and treatment of situations where no solution exists, or more than one solution satisfies the specified performance criteria and constraints.

Deep learning has opened new possibilities for addressing key limitations of traditional inverse design methods, including the need for time-consuming iterative simulations, the difficulty of optimizing under complex physical constraints, and the lack of flexibility in capturing highly nonlinear relationships between performance and geometry. Although in some studies, such as that of Sekar et al. [20], alternative architectures for inverse design have been explored, generative models such as generative adversarial networks (GANs) are particularly well-suited for this type of problem, since they offer a promising alternative by learning these relationships directly from data, enabling fast and accurate design generation once trained. As explained by Mehrjardi et al. [17,18], in a GAN framework, the generator produces geometries that satisfy the desired performance criteria, while the discriminator

ensures that the generated geometries resemble real-world designs. This combination encourages the generator to produce geometries that not only meet performance targets but also conform to learned geometric constraints, enhancing the quality of the samples created by the generator and avoiding the generation of unrealistic outputs.

For instance, Oh et al. [21] proposed a GAN-based design automation framework that generates a diverse set of structurally optimized designs, with a particular emphasis on aesthetic quality and design novelty. This work serves as a benchmark for the topic under discussion, but it lacks aerodynamic considerations. In contrast, Shu et al. [22] presented a method based on a GAN for 3D geometry generation, with a self-updating training scheme where performance-evaluated designs are reintroduced into the training set. This approach addresses functional feasibility through physics-based validation, yet it does not explicitly tackle the inverse design problem.

Regarding problems closely related to the present study, some authors have used GANs for inverse airfoil design. For example, Achour et al. [23] used a Conditional GAN (CGAN) for airfoil optimization based on aerodynamic coefficients and airfoil area; Chen et al. [24] and Tan et al. [25] designed CGAN architectures with interpolation layers for better geometry representation; Wang et al. [26] proposed a generative model for supercritical airfoil generation, based on the wall Mach number distribution; and Yilmaz and German [19] performed inverse airfoil design based on polar data using another CGAN.

This study advances the field by proposing the use of a CGAN for the design of 4-digit NACA airfoils with Trailing Edge (TE) flaps, considering both flow conditions (Reynolds number and angle of attack) and desired aerodynamic performance (drag and lift coefficients). In comparison with existing works that mainly focus on airfoil shape design based on desired aerodynamic performance, this study integrates flow conditions and geometric modifications by means of flow control devices into the design process, offering a novel approach to aerodynamic optimization. By incorporating TE flaps into the generated geometries, this work extends the applicability of GAN-based inverse design to more complex and functionally adaptive aerodynamic configurations.

Traditional inverse design methods rely on iterative optimization loops that require multiple evaluations of CFD models, which can be computationally expensive and time-consuming. In contrast, a data-driven approach, such as the one proposed in this study, can generate candidate geometries in a single forward pass through a trained model, enabling near-instantaneous design generation. This is especially advantageous in scenarios requiring rapid design exploration, real-time system adaptation, or integration into control loops, where speed and flexibility are critical.

The remainder of the manuscript is organized as follows: Section 2 explains the methodology followed to perform the investigation, describing the conducted numerical simulations, which are used to train the CGAN, and the architecture of this network; Section 3 shows the results obtained in this study, comparing the geometries generated by the CGAN with the real ones; and Section 4 summarizes the main findings of this research.

2. Methodology

2.1. Numerical simulations

In this study, 4-digit NACA airfoils with a Trailing Edge (TE) flap were considered. The front part of the airfoil is defined as the fixed part and remains static for all the cases; the rear part of the

airfoil is defined as the flap, whose rotation angle (θ) depends on the case. Figure 1 shows a schematic view of the tested case.

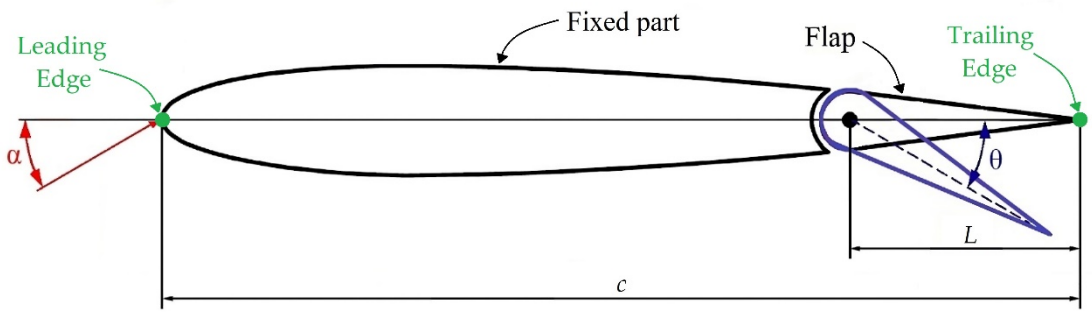


Figure 1. Dimensions and modifiable parameters of the tested case.

Different simulations of the case under discussion were carried out, with the objective of obtaining a diverse dataset for training the CGAN. In these simulations, a chord length of $c = 1\,m$, flap rotation angles between $\theta = 0°$ and $\theta = 30°$, and flap lengths between $L = 0.2c$ and $L = 0.3c$ were considered. Regarding the flow, Reynolds numbers between $Re = 0.1 \cdot 10^6$ and $Re = 2 \cdot 10^6$ and angles of attack of the flow between $\alpha = -5°$ and $\alpha = 5°$ were considered. XFOIL [27] software was used to perform the simulations, from which drag (C_D) and lift (C_L) coefficients were calculated. For training purposes, only the cases whose solution achieved convergence were selected, for a total of 47,645 cases.

Following the study of Achour et al. [23], a cosine distribution was used for the representation of the airfoil, so that the most significant areas—the Leading Edge (LE, which refers to the front-most portion of the airfoil) and the Trailing Edge (TE, which refers to the rear portion of the airfoil)—contained more sampling points for a more accurate representation. This is important because these regions exhibit the most complex geometric curvature and have a critical influence on aerodynamic performance. Mathematically, the x-coordinates were defined as shown in Eq (1).

$$x_i = \frac{1}{2}\left(1 - \cos\left(\frac{i\pi}{N-1}\right)\right),\ i = 0, 1, 2, \ldots, N-1 \tag{1}$$

where N represents the total number of discretization points.

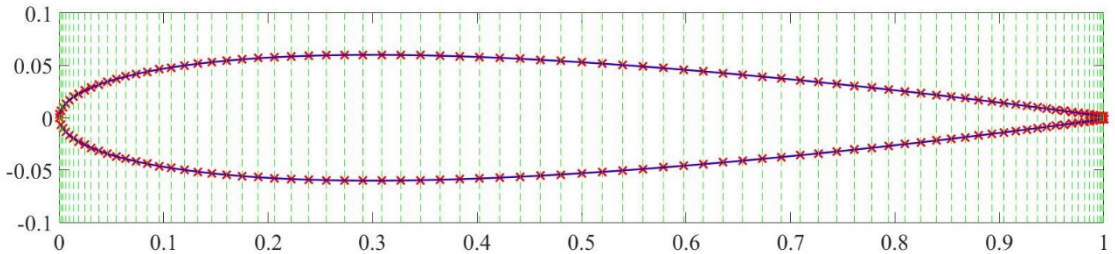


Figure 2. Example of the points selected (red crosses) for the representation of an airfoil.

With the selected representation, the x-coordinates were kept constant for all cases, while the y-coordinates were specific to each geometry. This simplifies the design and training of the CGAN, since it only has to generate the y-coordinates, and also avoids the generation of physically implausible outputs, such as self-intersecting surfaces. The selected distribution contains 80 x-coordinates, for a total of 160 points. Figure 2 shows an example of the points extracted for the airfoil representation.

2.2. CGAN architecture

For the generation of airfoils with flow control devices, a CGAN architecture was considered. The generator of this CGAN generates the coordinates of the airfoil from a noise signal, the flow characteristics (α and Re), and the desired aerodynamic characteristics (C_D and C_L). As previously explained, the x-coordinates were kept constant for all cases, so it only generates the y-coordinates. The discriminator labels the coordinates passed to it as *real* or *generated*, receiving the data of the real airfoils, the generated airfoils, and the characteristics of each case. Figure 3 shows the CGAN architecture used for this study. For designing and training, the proposed CGAN MATLAB software [28] was used, with its Deep Learning Toolbox [29].

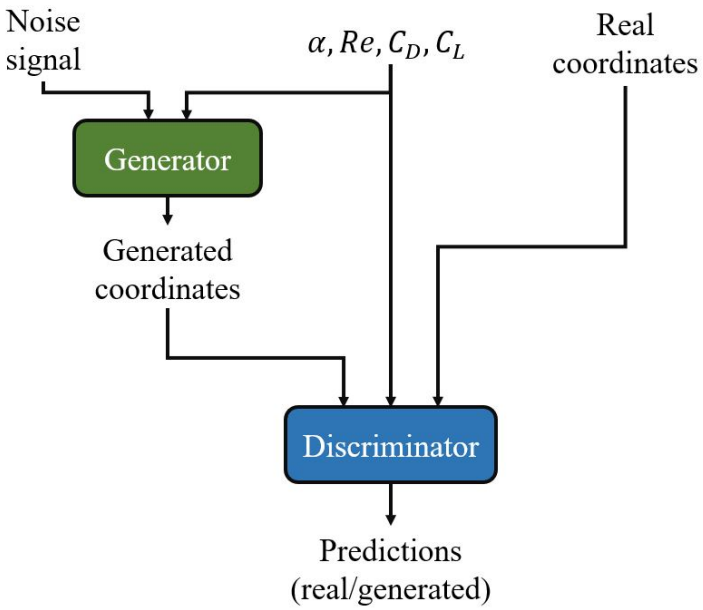


Figure 3. Architecture of the used CGAN.

The generator is a convolutional neuronal network (CNN) with two input layers, one for the noise signal and another one for the problem characteristics. Both inputs are projected so that the dimensions match the features of the network, and then the obtained layers are concatenated. The size of the layers is then increased by one-dimensional transposed convolutions, doubling the size of the data while halving the number of filters until the dimensions of the airfoil are reached. These transposed convolutions are performed with a kernel of 5 in the first layers and a kernel of 3 in the last layers, and all of them are followed by ReLU activation functions. Figure 4 shows the architecture of the generator.

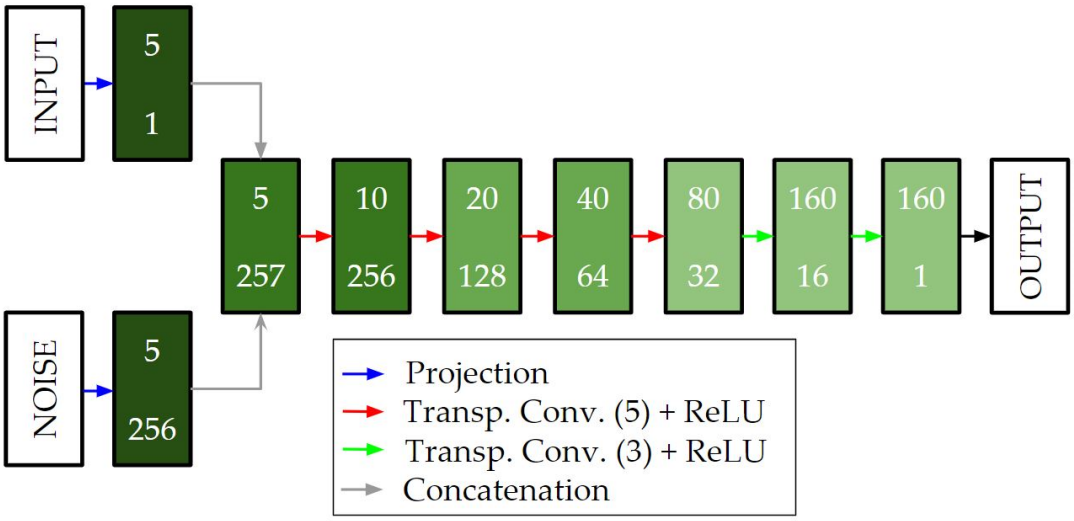


Figure 4. Architecture of the generator.

The discriminator is also a CNN that receives two inputs, the airfoil coordinates and the problem features. These inputs are concatenated after projecting the problem features, in order to match the dimensions of the layers. After the concatenation, a dropout layer sets input elements to zero with a probability of 0.25, in order to avoid overfitting. Subsequently, convolutions are performed, with a kernel of 5, followed by leaky ReLU activation functions with a slope of 0.2. In contrast to the generator, in this case the size of the data is halved, while the quantity of filters is doubled, aiming to reduce the data until a single scalar value is obtained. In the last layer, a convolution with a kernel of 10 is performed, and the sigmoid activation function is used, so that the output determines the probability that the airfoil is real (value close to 1) or generated (value close to 0). Figure 5 shows the architecture of the discriminator.

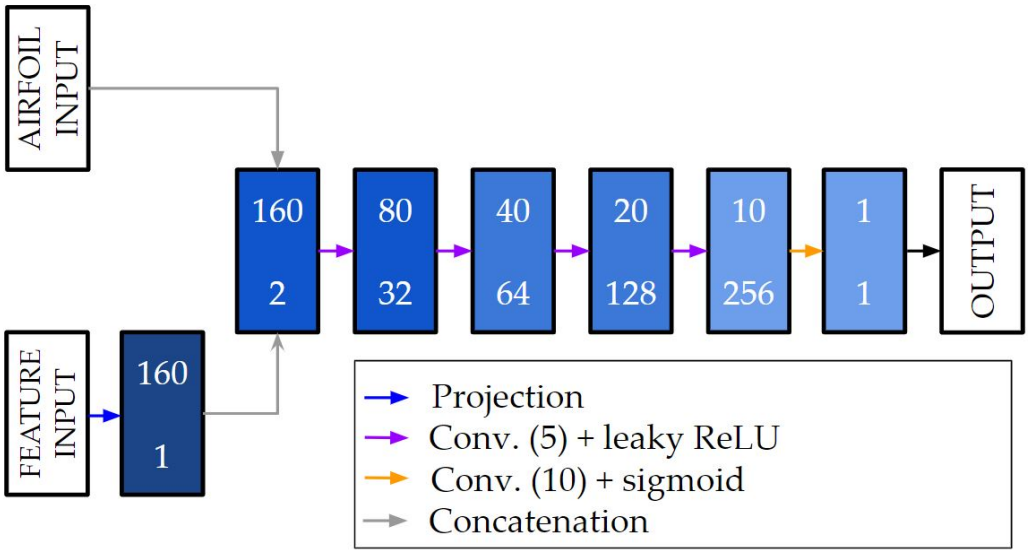


Figure 5. Architecture of the discriminator.

To train the CGAN, the data was split into 90% training and 10% testing, a batch size of 128 was selected, and the Adam optimizer [5] was used with a learning rate of 0.0002. Training was performed until the generator and discriminator losses reached convergence, with the generator loss being lower than the discriminator loss. Aiming to avoid overfitting, L2 regularization with a weight decay of 0.001 was applied, and losses were monitored on the test-set.

2.3. Generalization

An additional dataset was created to test the capacity for generalization of the proposed neural network. The objective of this dataset was to test the ability of the CGAN to transfer the acquired knowledge to conditions not seen during training. To this end, the cases considered in this dataset include extreme conditions not covered by the original dataset, these being $-7° \leq \alpha < -5°$, $5° < \alpha \leq 7°$, $Re = 50{,}000$, and $Re = 5{,}000{,}000$. The combination of these conditions results in the dataset for generalization, for a total of 42,092 cases.

3. Results

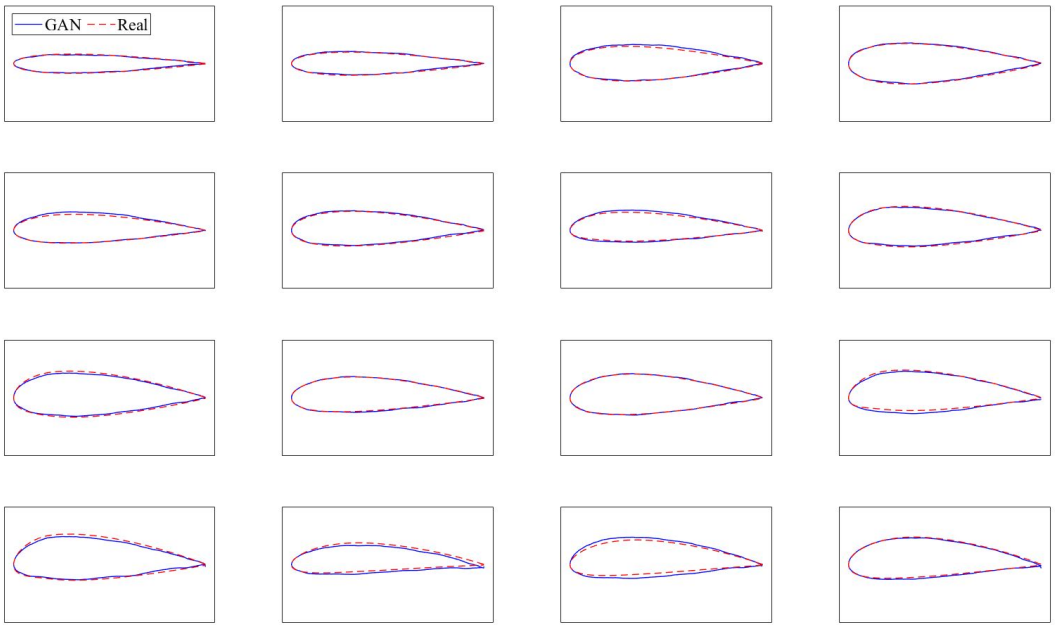
A qualitative comparison of the results, an aerodynamic performance analysis, and a computational performance analysis were carried out in order to assess the ability of CGAN to generate airfoils with flow control devices. For these analyses, only the cases of the test-set and generalization-set are considered, which are the cases that the neural network has not seen during training.

3.1. Qualitative comparison

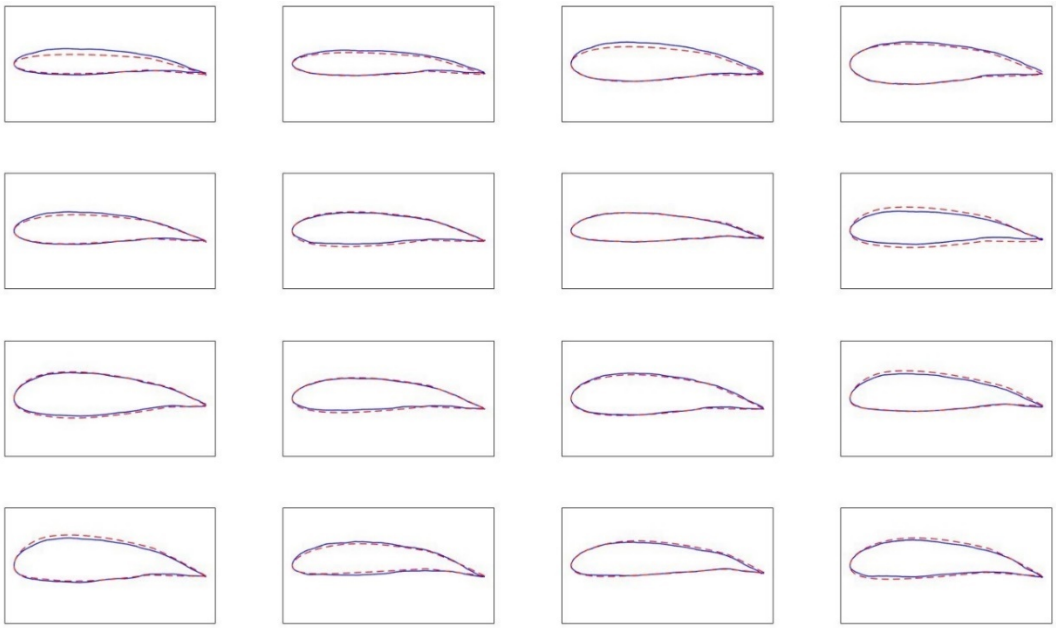
A qualitative comparison of the airfoils generated by the CGAN and the real ones was carried out in order to evaluate the capacity of this network to generate this kind of geometry. In this comparison, different random cases of the test-set, with different flap angles, were considered. Figure 6 shows the comparison of real and generated geometries.

The results show that the designed CGAN is able to generate geometries of high similarity with the real ones, adequately representing the main characteristics of the airfoils. The generated airfoils present certain discontinuities, which can be easily solved by different techniques, such as the interpolation used by Chen et al. [24]. The thickness of the generated airfoils is slightly higher than the real airfoils for the thinner airfoils, and slightly lower for the thicker airfoils. In general terms, the camber is well defined, both in value and position.

Concerning the flap, a correct representation of the flap is observed for all samples. However, in some cases, the start of the flap is not properly defined, being a continuation of the airfoil. This is more noticeable for low flap angles.



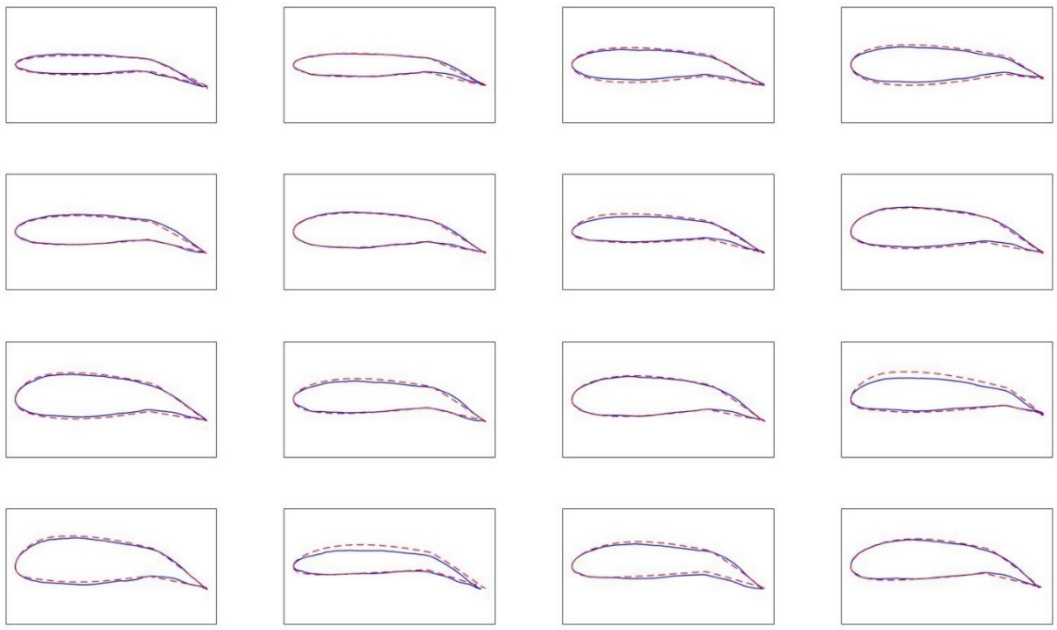
(a)



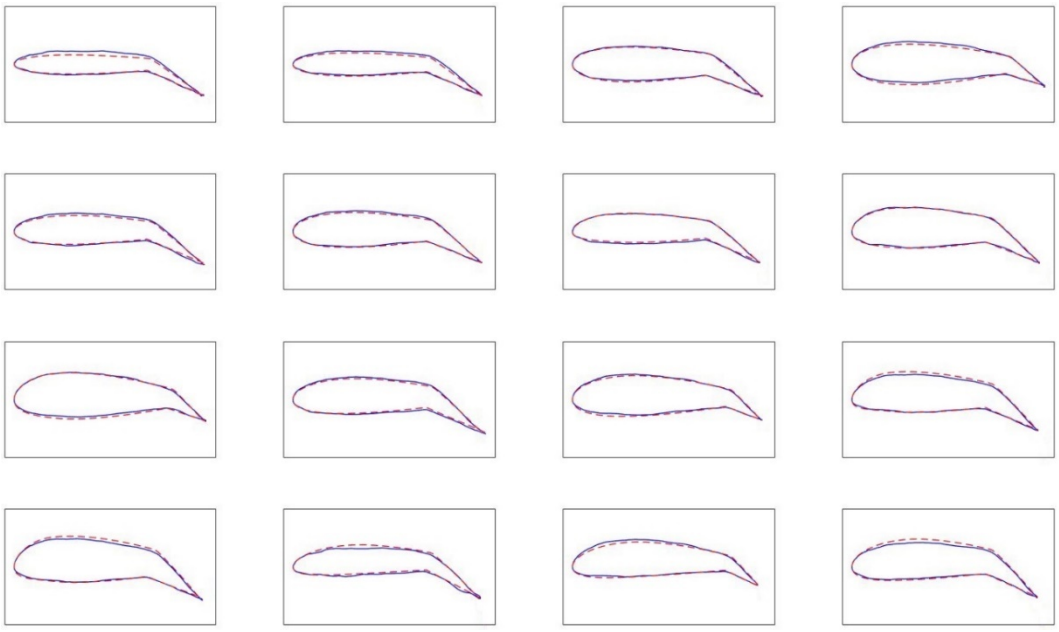
(b)

Continued on next page

Figure 6. Comparison of the real airfoils and the ones generated by the CGAN. (a) $\theta = 0°$; (b) $\theta = 10°$; (c) $\theta = 20°$; (d) $\theta = 30°$.



(c)



(d)

Figure 6. Comparison of the real airfoils and the ones generated by the CGAN. (a) $\theta = 0°$; (b) $\theta = 10°$; (c) $\theta = 20°$; (d) $\theta = 30°$.

3.2. Aerodynamic performance of generated airfoils

The aerodynamic coefficients provided by the CGAN-generated airfoils were analyzed, since the

main objective of the neural network is to generate geometries that meet the established aerodynamic requirements. This analysis focuses on the relative error (ε_r), calculated following Eq (2), of the coefficients provided by the generated airfoils compared to the reference coefficients defined as input to the network, evaluating the error for different flow conditions with the objective of identifying the scenarios in which the CGAN performs better and worse. Tables 1 and 2 show the relative error of the C_D and C_L, respectively, of the generated airfoils.

$$\varepsilon_r = \frac{|C_{real} - C_{generated}|}{C_{real} + k} \cdot 100 \tag{2}$$

where C represents the aerodynamic coefficients (C_D or C_L, depending on the error that is calculated) and k is an adjusting scalar (with a value of 0.001) that is added for cases in which C is equal to 0, in order to avoid infinite relative errors.

Table 1. Mean relative error ± standard deviation of relative error (both in %) of the C_D of generated airfoils.

α	Re						
	5×10^4	10^5	2.5×10^5	5×10^5	10^6	2×10^6	5×10^6
$[-7°,-5°)$	3.65 ± 0.51	3.38 ± 0.95	4.80 ± 1.38	7.87 ± 0.91	14.07 ± 3.7	18.36 ± 3.11	23.42 ± 5.8
$[-5°,-3°)$	3.69 ± 0.8	2.99 ± 0.42	3.11 ± 1.11	9.19 ± 1.5	13.79 ± 2.95	16.81 ± 3.41	22.17 ± 4.24
$[-3°,-1°)$	4.08 ± 1.19	2.94 ± 0.74	4.19 ± 0.65	10.40 ± 2.78	16.98 ± 2.57	18.62 ± 3.87	24.03 ± 4.67
$[-1°, 1°)$	5.76 ± 2.12	3.87 ± 1.24	3.02 ± 0.54	7.69 ± 2.12	16.46 ± 3.82	17.16 ± 3.43	23.91 ± 5.13
$[1°, 3°)$	4.42 ± 0.76	3.44 ± 0.98	2.41 ± 0.59	5.89 ± 0.95	15.45 ± 3.64	14.94 ± 3.51	21.52 ± 3.42
$[3°, 5°]$	4.99 ± 0.83	3.62 ± 0.91	3.65 ± 1.12	5.72 ± 1.14	11.95 ± 1.96	14.30 ± 3.61	22.27 ± 3.89
$(5°, 7°]$	5.96 ± 1.78	4.04 ± 0.56	3.96 ± 1.03	5.28 ± 0.76	10.34 ± 3.26	14.70 ± 3.12	24.51 ± 5.02

Table 2. Mean relative error ± standard deviation of relative error (both in %) of the C_L of generated airfoils.

α	Re						
	5×10^4	10^5	2.5×10^5	5×10^5	10^6	2×10^6	5×10^6
$[-7°,-5°)$	3.62 ± 0.68	6.33 ± 1.14	5.78 ± 1.29	7.54 ± 1.39	8.66 ± 1.6	9.97 ± 2.02	10.46 ± 2.11
$[-5°,-3°)$	8.21 ± 1.85	9.82 ± 1.74	11.41 ± 2.46	12.13 ± 2.3	12.41 ± 2.55	14.93 ± 2.79	13.73 ± 2.39
$[-3°,-1°)$	17.21 ± 3.17	15.74 ± 3.48	15.55 ± 3.08	14.46 ± 2.57	17.60 ± 3.41	18.08 ± 3.81	19.15 ± 3.88
$[-1°,1°)$	20.81 ± 4.02	13.53 ± 2.71	14.47 ± 2.96	14.54 ± 2.43	15.74 ± 2.93	16.44 ± 3.19	18.13 ± 3.44
$[1°, 3°)$	9.42 ± 1.91	3.70 ± 0.81	5.04 ± 1.08	5.93 ± 1.18	6.13 ± 1.25	7.34 ± 1.52	8.42 ± 1.7
$[3°, 5°]$	4.86 ± 0.93	1.98 ± 0.44	1.99 ± 0.41	2.69 ± 0.58	2.99 ± 0.63	3.50 ± 0.81	5.68 ± 1.09
$(5°, 7°]$	4.40 ± 0.97	4.27 ± 0.84	3.58 ± 0.77	3.49 ± 0.71	4.58 ± 0.96	4.89 ± 1.03	5.33 ± 1.02

The results indicate that the aerodynamic characteristics of the airfoils generated by the CGAN closely approximate the desired properties in all cases, exhibiting low relative errors in both aerodynamic coefficients.

For C_D, the relative error increases with Re. This behavior is attributed to the fact that, as previously discussed, the geometric feature most affected during airfoil generation is thickness. Consequently, since C_D tends to decrease with increasing Re, the relative error becomes more

sensitive to slight variations in thickness at high Re values. Under the same Re conditions, variations in α do not significantly affect the relative error.

In contrast, C_L is primarily influenced by α. The largest relative errors for C_L occur at low α (particularly between -3° and 1°), where C_L approaches zero. In this range, minor geometric deviations can lead to disproportionately large relative errors. Although increasing Re within the same α range also leads to an increase in relative error, its effect on C_L is considerably less pronounced than on C_D.

Regarding the generalization-set, the results demonstrate that the proposed CGAN successfully generalizes to extreme conditions not encountered during training, maintaining low relative errors comparable to those observed in the test set. For varying values of α, the network demonstrates good generalization performance across both extremes. In the case of different Re values, the network generalizes significantly better at low Re than at high Re, which is consistent with the results obtained with the original test-set.

3.3. Performance analysis

A comparison of the computational time required by each method was carried out. In this case, a fair comparison between the computational time required by CFD-based tools and the proposed CGAN cannot be made, since CFD simulations are responsible for performing forward design, i.e., providing results from a given geometry, while CGAN is responsible for performing the inverse design, i.e., providing a geometry for given characteristics. Therefore, as the number of CFD simulations required to obtain the design that meets the desired characteristics is variable, the time and computational resources required by CFD are case-dependent.

However, CGAN is able to provide extremely fast designs, generating the 4765 geometries corresponding to the test set in 0.8435 s, which is equivalent to 1.77×10^{-4} s per geometry, whereas obtaining CFD results of a single case using XFOIL [27] requires an average of 0.54 s. In both cases, a single core of the Intel Xeon Gold 5120 CPU was used.

4. Conclusions

In this study, the use of a CGAN was proposed for the inverse design of airfoils with flow control devices, particularly 4-digit NACA airfoils with a TE flap. The designed CGAN receives the flow characteristics (Re and α) and the desired aerodynamic characteristics (C_D and C_L) and designs a geometry that meets the provided input characteristics.

The results show that the CGAN is able to generate geometries with a strong resemblance to the real ones, being able to design the airfoil and flap properly, with very similar aerodynamic properties, and being extremely fast. Additionally, the proposed CGAN is able to generalize adequately to extreme situations not seen during training, making it feasible to use this approach for a wider variety of situations, apart from the ones in which training conditions are considered. Regarding the considered flow parameters, the CGAN struggles to obtain the desired C_D for high Re cases and the desired C_L for α values close to 0; these are the cases in which higher relative errors are obtained. Nevertheless, these errors are considered acceptable given the conditions encountered in these scenarios.

Despite the advances made in this study, several promising directions remain open for future research. One avenue involves extending the current CGAN framework to support more complex geometries, such as multi-element airfoils or other fluid dynamic systems. This would require adapting

the network to work with more sophisticated data representations, such as point clouds or mesh-based surfaces. Additionally, incorporating physics-informed loss functions (considering geometric or aerodynamic constraints) could improve the physical plausibility of the generated designs. Another important direction for future research is the integration of geometric regularization layers within the generator architecture, instead of just smoothing the output values. These approaches would enforce smoothness and continuity in the output geometries by design, thus addressing issues related to performance sensitivity caused by shape discontinuities. Such representations also allow for more compact and structured encoding of airfoil geometries, potentially improving both model interpretability and training efficiency. All these potential research directions, alongside many others, highlight the promising and impactful nature of inverse design and optimization of fluid dynamic systems using deep learning tools.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors are grateful for the support provided by SGIker of UPV/EHU. This research was developed under the frame of the Joint Research Laboratory on Offshore Renewable Energy (JRL-ORE). This work has been partially supported by the Government of the Basque Country, program: Elkartek MOREDIGITAL; Grant No.: KK-2024/00117 and DeepBask Grant No.: KK-2024/00069. U.F. G. was supported by the Mobility Lab Foundation, a governmental organization of the Provincial Council of Araba and the local council of Vitoria-Gasteiz and by Government of the Basque Country, ITSAS-REM Grant No.: IT1514-22.

Conflict of interest

Unai Fernandez-Gamiz is a guest editor for the special issue of the ERA and was not involved in the editorial review or the decision to publish this article. All authors declare that there are no competing interests.

References

1. N. Thuerey, K. Weißenow, L. Prantl, X. Hu, Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows, *AIAA J.*, **58** (2020), 25–36. https://doi.org/10.2514/1.J058291
2. H. Chen, L. He, W. Qian, S. Wang, Multiple aerodynamic coefficient prediction of airfoils using a convolutional neural network, *Symmetry*, **12** (2020), 544. https://doi.org/10.3390/sym12040544
3. E. Yilmaz, B. German, A convolutional neural network approach to training predictors for airfoil performance, in *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA, 2017. https://doi.org/10.2514/6.2017-3660

4. K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, A. Ballesteros-Coll, A. Zulueta, CNN-based flow control device modelling on aerodynamic airfoils, *Sci. Rep.*, **12** (2022), 8205. https://doi.org/10.1038/s41598-022-12157-w
5. K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, R. Garcia-Fernandez, X. Uralde-Guinea, CNN-based vane-type vortex generator modelling, *Eng. Appl. Comput. Fluid Mech.*, **18** (2024), 2300481. https://doi.org/10.1080/19942060.2023.2300481
6. S. J. Jacob, M. Mrosek, C. Othmer, H. Köstler, Deep learning for real-time aerodynamic evaluations of arbitrary vehicle shapes, *SAE Int. J. Passeng. Veh. Syst.*, **15** (2022). https://doi.org/10.4271/15-15-02-0006
7. R. Garcia-Fernandez, K. Portal-Porras, O. Irigaray, Z. Ansa, U. Fernandez-Gamiz, CNN-based flow field prediction for bus aerodynamics analysis, *Sci. Rep.*, **13** (2023), 21213. https://doi.org/10.1038/s41598-023-48419-4
8. B. Du, P. D. Lund, J. Wang, Combining CFD and artificial neural network techniques to predict the thermal performance of all-glass straight evacuated tube solar collector, *Energy*, **220** (2021), 119713. https://doi.org/10.1016/j.energy.2020.119713
9. J. Ren, H. Wang, K. Luo, J. Fan, A priori assessment of convolutional neural network and algebraic models for flame surface density of high Karlovitz premixed flames, *Phys. Fluids*, **33** (2021), 036111. https://doi.org/10.1063/5.0042732
10. J. Rabault, M. Kuchta, A. Jensen, U. Réglade, N. Cerardi, Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control, *J. Fluid Mech.*, **865** (2019), 281–302. https://doi.org/10.1017/jfm.2019.62
11. F. Ren, J. Rabault, H. Tang, Applying deep reinforcement learning to active flow control in weakly turbulent conditions, *Phys. Fluids*, **33** (2021), 037121. https://doi.org/10.1063/5.0037371
12. D. Fan, L. Yang, Z. Wang, M. S. Triantafyllou, G. E. Karniadakis, Reinforcement learning for bluff body active flow control in experiments and simulations, *Proc. Natl. Acad. Sci. U.S.A.*, **117** (2020), 26091–26098. https://doi.org/10.1073/pnas.2004939117
13. B. Z. Han, W. X. Huang, C. X. Xu, Deep reinforcement learning for active control of flow over a circular cylinder with rotational oscillations, *Int. J. Heat Fluid Flow*, **96** (2022), 109008. https://doi.org/10.1016/j.ijheatfluidflow.2022.109008
14. K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, R. Garcia-Fernandez, S. E. Berrizbeitia, Active flow control on airfoils by reinforcement learning, *Ocean Eng.*, **287** (2023), 115775. https://doi.org/10.1016/j.oceaneng.2023.115775
15. G. Lee, Y. Joo, S. U. Lee, T. Kim, Y. Yu, H. G. Kim, Design optimization of heat exchanger using deep reinforcement learning, *Int. Commun. Heat Mass Transfer*, **159** (2024), 107991. https://doi.org/10.1016/j.icheatmasstransfer.2024.107991
16. Y. Wang, W. Wang, G. Tao, H. Li, Y. Zheng, J. Cui, Optimization of the semi-sphere vortex generator for film cooling using generative adversarial network, *Int. J. Heat Mass Transfer*, **183** (2022), 122026. https://doi.org/10.1016/j.ijheatmasstransfer.2021.122026
17. S. A. A. Mehrjardi, A. Khademi, M. Fazli, Optimization of a thermal energy storage system enhanced with fins using generative adversarial networks method, *Therm. Sci. Eng. Prog.*, **49** (2024), 102471. https://doi.org/10.1016/j.tsep.2024.102471

18. S. A. A. Mehrjardi, A. Khademi, S. M. M. Safavi, Machine learning approach to balance heat transfer and pressure loss in a dimpled tube: Generative adversarial networks in computational fluid dynamics, *Therm. Sci. Eng. Prog.*, **57** (2025), 103116. https://doi.org/10.1016/j.tsep.2024.103116
19. E. Yilmaz, B. German, Conditional generative adversarial network framework for airfoil inverse design, in *AIAA Aviation 2020 Forum*, AIAA, 2020. https://doi.org/10.2514/6.2020-3185
20. V. Sekar, M. Zhang, C. Shu, B. C. Khoo, Inverse design of airfoil using a deep convolutional neural network, *AIAA J.*, **57** (2019), 993–1003. https://doi.org/10.2514/1.J057894
21. S. Oh, Y. Jung, S. Kim, I. Lee, N. Kang, Deep generative design: Integration of topology optimization and generative models, *J. Mech. Des.*, **141** (2019), 111405. https://doi.org/10.1115/1.4044229
22. D. Shu, J. Cunningham, G. Stump, S. W. Miller, M. A. Yukish, T. W. Simpson, et al., 3D design using generative adversarial networks and physics-based validation, *J. Mech. Des.*, **142** (2019), 071701. https://doi.org/10.1115/1.4045419
23. G. Achour, W. J. Sung, O. J. Pinon-Fischer, D. N. Mavris, Development of a conditional generative adversarial network for airfoil shape optimization, in *AIAA Scitech 2020 Forum*, Orlando, FL, 2020. https://doi.org/10.2514/6.2020-2261
24. W. Chen, K. Chiu, M. Fuge, Aerodynamic design optimization and shape exploration using generative adversarial networks, in *AIAA Scitech 2019 Forum*, San Diego, CA, 2019. https://doi.org/10.2514/6.2019-2351
25. X. Tan, D. Manna, J. Chattoraj, L. Yong, X. Xinxing, D. M. Ha, et al., Airfoil inverse design using conditional generative adversarial networks, in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, IEEE, (2022), 143–148. https://doi.org/10.1109/ICARCV57592.2022.10004343
26. J. Wang, R. Li, C. He, H. Chen, R. Cheng, C. Zhai, et al., An inverse design method for supercritical airfoil based on conditional generative models, *Chin. J. Aeronaut.*, **35** (2022), 62–74. https://doi.org/10.1016/j.cja.2021.03.006
27. M. Drela, XFOIL: An analysis and design system for low Reynolds number airfoils, in *Low Reynolds Number Aerodynamics. Lecture Notes in Engineering* (eds. T. J. Mueller), Springer, Berlin, Heidelberg, **54** (1989). https://doi.org/10.1007/978-3-642-84010-4_1
28. MATLAB, Mathworks. Available from: https://es.mathworks.com/products/matlab.html.
29. Deep Learning Toolbox, Mathworks. Available from: https://es.mathworks.com/products/deep-learning.html.

AIMS Press