Physics-Enhanced Neural Networks for Aerodynamic Shape Optimization of UAV Wings

Chris Pliakos[*] and Giorgos Efrem[†]
Aristotle University of Thessaloniki, 54124 Thessaloniki, Central Macedonia, Greece.

Grigorios Dimitriadis[‡]
University of Liège, 4000 Liège, Walloon Region, Belgium.

Pericles Panagiotou[§]
Aristotle University of Thessaloniki, 54124 Thessaloniki, Central Macedonia, Greece.

Aerodynamic shape optimization plays a crucial role in the development of UAV wings, directly impacting mission efficiency, endurance, and overall flight performance. Traditional optimization methods rely heavily on high-fidelity CFD simulations which, due to their high computational cost, limit extensive exploration. Another alternative is the use of computationally efficient, yet physically simplified, low-fidelity methods that poorly represent nonlinear aerodynamic phenomena such as aerodynamic stall. To address these limitations, a multi-fidelity neural network architecture is developed, combining large amounts of low-fidelity data with fewer high-fidelity evaluations. To strengthen predictive accuracy, this neural network framework is enhanced through domain knowledge-driven feature engineering and physics-informed soft constraints, specifically targeting the relatively undersampled stall region. Additionally, Monte Carlo dropout is incorporated to estimate uncertainty, enabling more confident and reliable design decisions. The integration of aerodynamic principles ensures realistic lift curves exhibiting proper stall behavior and physically plausible drag polars, thereby achieving robust, physically consistent predictions across the operational envelope of fixed-wing UAVs.

I. Nomenclature

a	=	angle of attack
C_L	=	Coefficient of lift
C_D	=	Coefficient of drag
C_M	=	Coefficient of pitching moment
$C_{D,i}$	=	Coefficient of induced drag
$C_{D,0}$	=	Coefficient of zero lift drag
NN_{LF}	=	Low-Fidelity neural network
NN_{HF}	=	High-Fidelity neural network
NN_{MFNN}	=	Multi-Fidelity neural network
R^2	=	Coefficient of determination
Re	=	Reynolds Number

II. Introduction

Unmanned aerial vehicles (UAVs) are rapidly transforming modern aerospace applications, playing a pivotal role in domains ranging from surveillance and logistics to environmental monitoring and defense [1]. As the range and

[*] PhD Student, Mechanical Engineering Department, AIAA University Student Member, pliakosc@meng.auth.gr
[†] Mechanical Engineer, Mechanical Engineering Department, egiorgosm@meng.auth.gr
[‡] Professor, Aerospace and Mechanical Engineering Department, Senior Member AIAA, gdimitriadis@uliege.be
[§] Assistant Professor, Mechanical Engineering Department, AIAA Member, peripan@meng.auth.gr

complexity of UAV missions grows, so does the need for highly efficient, optimized aerodynamic configurations tailored to specific performance objectives. Designing such configurations requires the capability to rapidly and accurately evaluate aerodynamic behavior across broad design spaces.

Traditionally, aerodynamic shape optimization (ASO) has relied on a combination of high-fidelity Computational Fluid Dynamics (CFD), particularly Reynolds-Averaged Navier–Stokes (RANS) simulations, and low-fidelity methods such as vortex lattice or panel methods [2,3]. While high-fidelity simulations offer accurate flow predictions, they are computationally intensive – especially when using adjoint-based gradient methods – making them impractical for wide design space exploration. In contrast, low-fidelity approaches are computationally efficient but inherently limited in modeling viscous effects and non-linear flow phenomena, particularly at high angles of attack. To address these trade-offs, machine learning (ML) has emerged as a powerful surrogate modeling strategy, capable of approximating complex aerodynamic responses at a fraction of the cost. As discussed by Brunton and Kutz [4], ML models act as nonlinear, data-adaptive mappings that can augment or even replace traditional solvers in early-stage design. More broadly, they emphasize that ML is fundamentally transforming the practice of aerospace engineering, offering new paradigms for modeling, control, and design across the entire discipline.

This paradigm shift is actively reshaping the ASO landscape by enabling faster, scalable, and more flexible design workflows. To further improve the robustness and accuracy of surrogate models – particularly in under-sampled or highly nonlinear regions – researchers have turned to multi-fidelity modeling strategies. Works by Forrester et al. [5] and Kennedy et al. [6] introduced variable-complexity and co-Kriging models, which integrate fast but coarse predictions (e.g., from panel solvers) with sparse high-fidelity CFD data. Recent advancements have extended these concepts into deep learning architectures. Yang et al. proposed a composite multi-fidelity neural network (MFNN) for optimizing the NASA CRM model, using large amounts of coarse-mesh CFD data in conjunction with fewer fine-mesh runs [7]. Similarly, Du et al. investigated MFNN performance across different fidelity ratios for airfoil optimization [8]. In the UAV domain Zadeh et al. applied MFNN frameworks to aerodynamic shape optimization, demonstrating that integrating fidelity levels enables rapid yet reliable performance prediction [9].

Despite these successes, most surrogate models for aerodynamic optimization remain purely data-driven and often lack generalization capability outside the training domain – particularly in flow regimes characterized by strong nonlinearity, such as post-stall behavior. To mitigate these limitations, recent studies have explored integrating physical knowledge into machine learning frameworks. For example, Eivazi et al. utilizes physics-informed neural networks to satisfy the RANS equation for airfoils [10]. Zobeiry et al. adopt the theory-guided machine learning paradigm to engineer input features that embed known physical correlations, improving generalization in composite material modeling [11,12]. In contrast, Zhao et al. modify the model architecture directly by introducing lattice layers that enforce monotonicity in the lift curve within the linear regime, applied to morphing wing aircraft [13].

In this work, a multi-fidelity neural network framework that incorporates aerodynamic theory to enhance prediction quality in aerodynamic shape optimization of UAV wings is presented. The model is trained using data from both a low-fidelity panel method and a high-fidelity RANS solver. Approximately 150,000 low-fidelity wing samples are used, about 20% of which also have corresponding high-fidelity data. To address limitations in purely data-driven models physics-based loss terms are introduced, based on classical aerodynamic relationships. These terms help ensure that the network output remains physically consistent, even in regions where training data is sparse or highly nonlinear. The proposed framework enables accurate and physically meaningful predictions of lift and drag.

III. Problem Statement

The objective of this work is to present the impact of enhancing and informing a neural network as part of a data-driven multi-fidelity framework with underlying physics and aerodynamic theory. Neural networks and machine learning techniques are known for their strong interpolation capacity, but they exhibit inferior performance in extrapolation – particularly in regions near or outside the boundaries of the design space or when the design space is under-sampled, as often occurs near stall. Common datasets include many aerodynamic evaluation points in the linear region, where lift progresses monotonically with the angle of attack, but far fewer points in the stall and post-stall areas. This study focuses on the application of physical laws, derived from aerodynamics and fluid dynamics, within neural networks in the form of loss function components. The integration of physical laws follows techniques described in the Physics-Informed Neural Networks (PINNs) literature [14]. However, in this work, although the flow field is modeled to generate the high-fidelity dataset, partial differential equations are not directly used in the loss formulation, as the neural networks operate on integrated aerodynamic quantities. The targeted aerodynamic geometries are wings from all UAV families operating in the low subsonic regime, i.e., with Mach numbers less than 0.3. The goal is to investigate the potential improvement in neural network inference, especially at high angles of attack, and the physical realism of the predicted polars and sweeps. For this purpose, two types of datasets are generated, a low-fidelity one using the OpenVSP VSPAERO panel solver, and a high-fidelity one using the RANS solver of Ansys Fluent. Both datasets are automatically generated using an in-house automated framework tailored for these UAVs, as described in Section IV.B. The quantities of interest are the aerodynamic coefficients of lift, drag, and moment. The neural networks used are simple Multi-Layer Perceptron (MLP) architectures, discussed in Section IV.C.

IV. Methodology

A. Geometry Parametrization

The geometry parametrization strategy employed in this study builds on a previously established framework for UAV wing modeling [15]. The formulation is designed to enable scalable geometry generation suitable for large-scale computational fluid dynamics (CFD) simulations and machine learning applications. Each wing is described through two complementary components, planform parametrization and airfoil shape parametrization, which together provide sufficient geometric expressiveness while maintaining a manageable number of design variables.

The wing planform is defined using 11 high-level geometric parameters distributed across two connected trapezoidal sections. This configuration supports a wide range of UAV wing designs, including both conventional planforms and more integrated geometries such as blended-wing-body (BWB) and flying wings configurations, while maintaining a tractable design space. The parametrization, illustrated in Fig. 1, includes spanwise lengths (b_1, b_2), sweep $(\Lambda_{1@c/4}, \Lambda_{2@c/4})$, twist $(i_{1@c/4}, i_{2@c/4})$ and dihedral (γ_1, γ_2) angles all referenced to a fixed origin at the leading edge of the centerline chord. Chord lengths (c_1, c_2, c_3) are specified independently to avoid over-constraining the geometry. It is worth noting that considering UAV design, the wing incidence angle must typically be defined. However, in this work, the incidence is fixed to zero, effectively aligning it with the angle of attack (AoA), to avoid introducing an additional design variable.

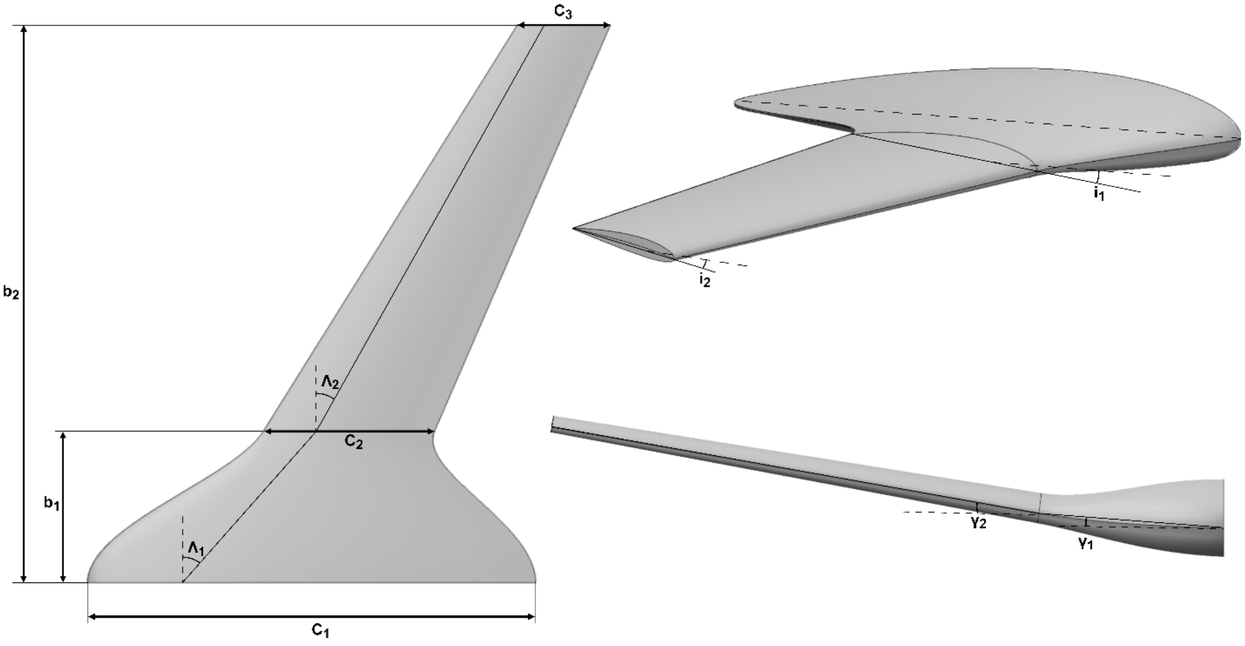


Fig. 1 Wing planform parametrization and the available design variables.

Complementing the two-section planform, each wing is assigned three distinct airfoil profiles located at the centerline (root), the section break (kink), and the tip. Airfoils are represented using a data-driven generative model based on Bézier-GAN (BGAN), originally introduced by Chen et al. [16] and adapted here for fixed-wing UAV wing applications. The BGAN model is trained on the UIUC airfoil database [17] and encodes each airfoil into an 8-dimensional latent vector that captures key geometric features such as thickness, camber, reflex, and curvature. With three airfoils per wing and eight latent parameters per airfoil, the airfoil definition contributes 24 variables to the overall geometry. Combined with the 11 planform parameters, each wing is fully described by a set of 35 independent design variables.

B. Design Space & Data Generation

The ranges of each geometric design variable are consistent with the operational envelope regarding the UAV families in [15], spanning fixed-wing UAV classes from Micro-Mini to HALE-Strike in accordance with NATO classification standards.

Wingspan limits are obtained according to Mitridis et al. [18], in their study they conducted a comprehensive analysis of 202 UAV platforms to investigate trends across NATO classification levels. By compiling a detailed database of technical and performance specifications, they identified correlations between key design parameters such as wingspan with gross take-off weight (GTOW), where GTOW is directly associated with the NATO classification. Considering Figure 6 in their work, a near-linear relationship between wingspan and GTOW is observed across the primary UAV classes, with most configurations falling within the $1-20\ meter$ span range. Outliers at the upper end, primarily associated with MALE and HALE-Strike UAVs, define the maximum bounds of the design space, while only a single Micro-Mini is recorded below the 1-meter threshold. The limits derived from this study provide a physically grounded basis for the present work's total wingspan ($2 \times b_2$) space. Chord length limits are also based on the smallest and largest chord lengths found in the database of Mitridis et al. excluding the outliers mentioned above. The limits for sweep, twist, and dihedral angles are defined based on values commonly observed in the literature [19]. Considering the low speeds ($M \leq 0.3$) at which these UAVs operate (Figure 21 in [18]) it is questionable whether or not wing sweep should be included in the design space. Although sweep is usually used to reduce the adverse effects of transonic - supersonic flow, in the case of BWB configurations it is necessary to achieve longitudinal stability [20].

Design variable constraints have been defined to exclude unusual wing geometries, such as wings with taper ratio values larger than 1, wings with very thin sections or wings with large sweep differences between the sections (such shapes can be observed in Figure 3 of [15]). Table 1 provides a comprehensive overview of the wing's planform limits and constraints, enabling a complete definition of the proposed framework's design space. These constraints are applied during the sampling phase of the design space where the fully independent variables are sampled according to the absolute limits and the per sample allowed ranges of the dependent variables are adjusted dynamically per sample.

Table 1 Design variables used to parametrize the wing planform and their respectable range.

Planform Parameter	Description	Units	Limits
c_1	Centerline chord length	m	$c_1 \in [0.2, 10]$
c_2	Root chord length	m	$c_2 \in [0.3 \times c_1, c_1]$
c_3	Tip chord length	m	$c_3 \in [0.3 \times c_2, c_2]$
b_1	c_2 spanwise location	m	$b_1 \in [0.2 \times b_2, 0.7 \times b_2]$
b_2	c_3 spanwise location (semispan)	m	$b_2 \in [1, 10]$
Λ_1	Sweep angle of section 1$@c/4$	deg	$\Lambda_1 \in [0, 60]$
Λ_2	Sweep angle of section 2$@c/4$	deg	$\Lambda_2 = f(\Lambda_1) \in [0, 60]$
i_1	Twist angle of $c_2@c/4$	deg	$i_1 \in [-5, 5]$
i_2	Twist angle of $c_3@c/4$	deg	$i_2 \in [-7, 7]$
γ_1	Dihedral angle of section 1	deg	$\gamma_1 \in [-10, 10]$
γ_2	Dihedral angle of section 2	deg	$\gamma_2 \in [-10, 10]$

Although the constraint-based sampling strategy ensures broad coverage of the geometric design space while filtering out unrealistic wing configurations, an inconsistency emerges when comparing the distribution of sampled aspect ratios (ARs) with those reported in the literature [21]. AR is widely recognized as a key classification metric in aircraft design due to its strong association with aerodynamic performance and efficiency [19]. To evaluate how well the generated samples align with real-world UAV designs, available statistical references are consulted. However, detailed distributions of low-level geometric parameters such as sweep or chord gradients are largely unavailable across public UAV datasets. Among the few reliable sources, Verstraete et al. [21] provide a broad statistical analysis of UAV sizing trends, regarding wingspan and wing area data. These two parameters enable the back-calculation of AR, serving as a benchmark for assessing the realism of the sampled design space.

To address the mismatch between the sampled and real-world AR distributions, a sampling refinement algorithm is implemented. The initial observation reveals that samples generated using traditional Latin Hypercube Sampling (LHS) methods exhibit a skewed AR distribution, with a median value significantly lower than that observed in operational UAV datasets. To resolve this, a probabilistic approach is employed using a Weibull probability density function (PDF), whose parameters are derived through least-squares fitting to the AR data extracted from [21]. This approach successfully reshapes the sampled AR distribution to closely match that of real UAV configurations. The final sample size of 10,000 wing configurations is chosen as a practical estimate, ensuring sufficient coverage of the vast design space to support robust machine learning model training.

The 24 latent vectors used to define 3 airfoil geometries in this framework are sampled using Latin Hypercube Sampling (LHS) over the range $[0, 1]$ for each dimension. To perform aerodynamic simulations, geometric definitions alone are insufficient – each wing sample must also be associated with specific free-stream conditions. Given the extensive range of geometric design space, different UAV classifications are expected to operate under different conditions. As demonstrated by the statistical analysis of GTOW as a function of cruise speed in [18], each class operates under unique conditions without a uniform trend applying to all classifications. To simplify data generation the operational altitude is kept constant for all wing samples at sea level, while different velocity ranges per UAV class are defined. Table 2, summarizes the wingspan and cruise speed ranges for three major UAV classes.

Table 2 Velocity sampling ranges with respect to UAV classification according to wingspan.

Class	Category	Wingspan [m]	Cruise speed [m/s]
Class I	Micro – Mini	≤ 8	$6 \leq V_\infty \leq 42$
Class II	Tactical	$8 \leq b \leq 18$	$20 \leq V_\infty \leq 42$
Class III	MALE – HALE	≥ 18	$27 \leq V_\infty \leq 103$

Given the constraint and AR "guided" geometric sampling discussed above, each sample is classified according to the tabulated wingspan ranges, and using LHS, its velocity is drawn from the velocity range associated with that class. This classification and sampling process is illustrated in Fig. 2, where the red dashed boxes represent the defined operational envelopes for each UAV class and the discrete clustering of samples reflects the applied class-based velocity assignment across the full dataset of 10,000 wing geometries.

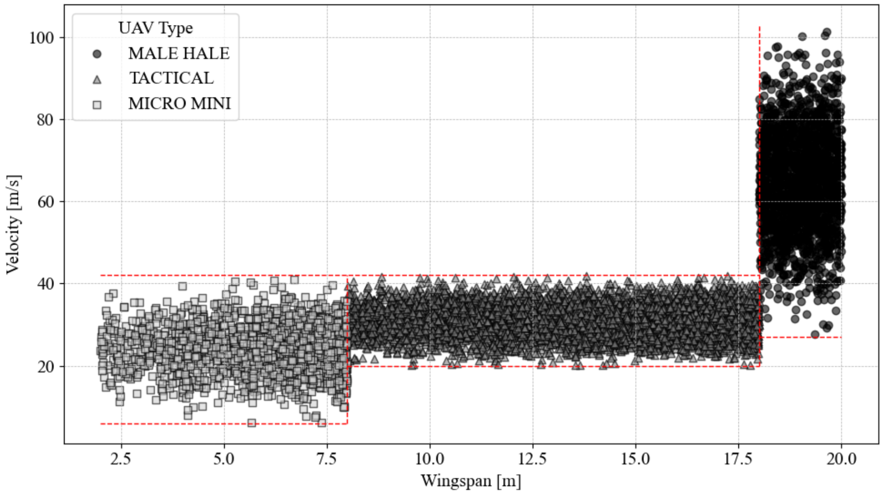


Fig. 2 Sampled velocity distribution per UAV geometric sample.

Each wing geometry in the dataset is evaluated using the OpenVSP VSPAERO panel solver at 15 angles of attack, uniformly covering the range from −5° to 20°, resulting in a dataset of approx. 150,000 samples. A subset of 20% is selected for high-fidelity evaluation using a RANS-based CFD pipeline under the same geometric and flow conditions. Low-fidelity simulations are performed in VSPAERO panel solver with a computational grid of 35 chordwise and 70 spanwise elements. The grid is weighed by span for each section and features element clustering near the leading and trailing edges, as recommended by validation studies [22]. High-fidelity data are generated using the automated CFD pipeline introduced in [15], which integrates geometry generation, meshing, and solver execution. The CFD setup follows established best practices for subsonic external aerodynamics, including convergence monitoring, residual control, and boundary-layer mesh refinement to maintain appropriate near-wall resolution for turbulence modeling, ensuring numerical robustness across all evaluated designs. Prior to training, the combined dataset is filtered using an interquartile range (IQR) method to remove approximately 1% of samples with outlier aerodynamic behavior. To prevent data leakage, the dataset is split into training, validation, and test subsets using an 80/10/10 ratio. Input features are then normalized using statistics derived solely from the training set.

C. Machine Learning Architectures

Machine learning techniques in aerodynamic shape optimization commonly employ neural networks. In this work, two (2) multi-layer perceptron neural networks constitute the core of the architecture, where the first one regards the low fidelity panel data, and the second one the high fidelity CFD data. They are coupled in a serial multi-fidelity fashion (also referred to as a '2-step' approach). The low-fidelity neural network (NN_{LF}) is trained independently, without access to high-fidelity data. Subsequently, the high-fidelity network (NN_{HF}) is trained using both CFD data and the predictions produced by NN_{LF} for the corresponding cases. The resulting input vector to NN_{HF} is illustrated in Eq. (1).

$$x_{HF} = [wing\ geometry\ design\ variables, flow\ conditions, C_L^{LF}, C_D^{LF}, C_M^{LF}\ \ ]^T \tag{1}$$

The rationale behind this multi-fidelity structure is based on the observation that low-fidelity data, which can be produced in large volumes to span the design space, retain physical consistency in regions of limited flow separation - particularly within the linear regime [23]. Although high-fidelity CFD data are more accurate, they are also computationally expensive and sparsely distributed. The function approximated by the multi-fidelity neural network (MFNN) is represented by Eq. (2), where $\rho(x)$ is a learned scaling function expressing the correlation between fidelity levels,$\{y_{LF}, y_{HF}\}$ and $\varepsilon(x)$ is the noise associated with the learning process.

$$\mathcal{F}_{MFNN} = \mathcal{F}_{HF} = \rho(x)\, y_L + \varepsilon(x) \tag{2}$$

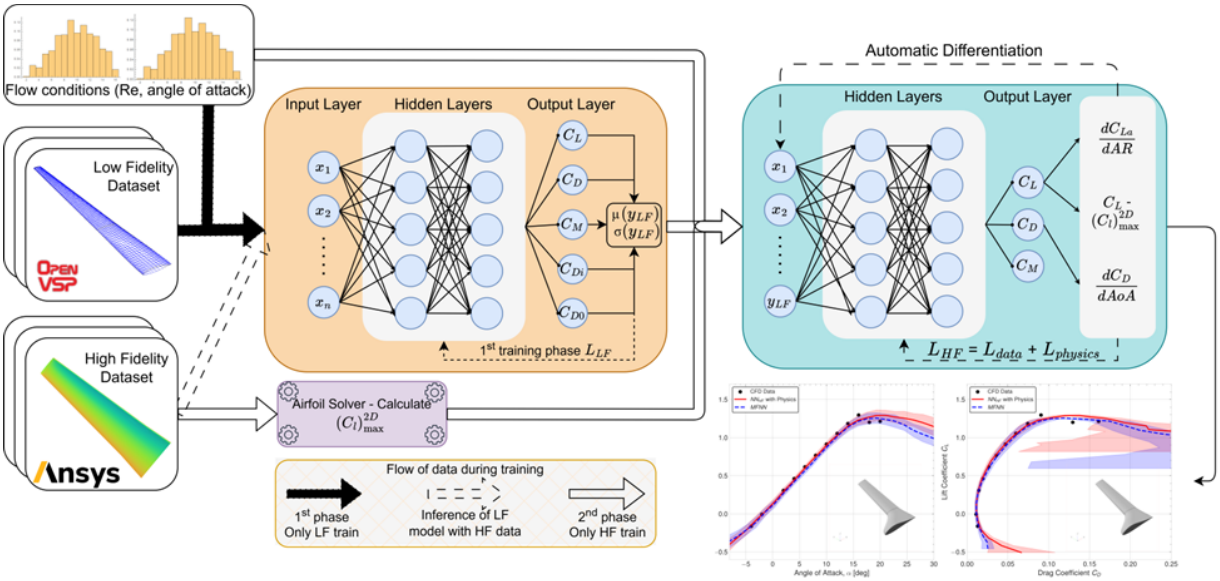


Fig. 3 Multi-fidelity architecture used in this work, and all necessary components.

This class of serial multi-fidelity neural networks belongs to a broader family of multi-fidelity modeling approaches and is characterized by its stability during training, as the two neural networks are trained individually, in contrast with architectures that require a joint optimization of both the loss function for the NN_{LF} and the NN_{HF} [24]. In those cases, a careful balance of weighting each loss function must be supervised, leading to a multi-objective optimization problem. The architecture of this work is shown in Fig. 3, with elements discussed in Section IV.D.

Once trained, neural networks produce deterministic outputs; a given input always yields the same output, as the weights and biases are fixed. However, in tasks such as aerodynamic shape optimization, probabilistic predictions - providing both the expected value (mean) and the uncertainty (standard deviation) - can offer significant advantages. This enables informed decision-making by identifying areas of high uncertainty, guiding the optimization algorithm toward wing designs where the neural network confidently predicts aerodynamic behavior. Moreover, a validation strategy employing high-cost CFD analyses can be steered toward selecting wings exhibiting lower uncertainty.

Monte Carlo Dropout (MCD) is selected as the uncertainty quantification technique by utilizing dropout layers during both training and inference [25]. This results in stochastic neuron activation during each forward pass, producing variable outputs for the same input. MCD approximates Bayesian inference and deep ensembles, while remaining computationally efficient and architecture-agnostic. In this work it is applied both in the NN_{LF} and NN_{HF}, yielding six outputs per model - three predictive means (μ) and three associated standard deviations (σ), as defined in Eq. (3) and Eq. (4).

$$y_{LF} = [\mu(C_L^{LF}), \mu(C_D^{LF}), \mu(C_M^{LF}), \sigma(C_L^{LF}), \sigma(C_D^{LF}), \sigma(C_M^{LF})] \tag{3}$$

$$y_{MFNN} = y_{HF} = [\mu(C_L^{HF}), \mu(C_D^{HF}), \mu(C_M^{HF}), \sigma(C_L^{HF}), \sigma(C_D^{HF}), \sigma(C_M^{HF})] \tag{4}$$

The number of forward passes (inferences) to compute the mean and standard deviation is set to 100, following best practices established in the original MCD work [25]. The dropout rate in both NN_{LF} and NN_{HF} is set 0.2 across all hidden layers. For a single wing polar (30 angle of attack points), the complete inference – comprising 100 forward passes through NN_{LF} to estimate the low-fidelity mean and standard deviation, followed by 100 evaluations of NN_{HF} – requires 0.13 seconds on an i7 Intel CPU. When batched across 100 wings, the total time is 0.276 seconds, corresponding to approximately 2.8 milliseconds per aerodynamic case. The remaining hyper-parameters for the networks resulted from a preliminary tuning study using the Optuna library and the Hyperopt algorithm [26]. All model development, training, and inference routines are implemented in Python using the PyTorch deep learning framework [27]. More specifically, the NN_{LF} features 2 hidden layers with 176 neurons each, the Sigmoid Linear Unit (SiLU) as the activation function, and is trained with Mean Squared Error (MSE) as the loss function. A learning rate scheduler reduces the learning rate by 10% if no improvement is observed over ten consecutive epochs. The NN_{HF} (and thus the MFNN) differs from the NN_{LF} primarily in its architecture, comprising three hidden layers instead of two. NN_{LF} is trained for 80 epochs, while NN_{HF} for 120, with total training time on the order of 5 minutes per model.

D. Injection of Physics

As mentioned in Section IV.C, neural networks trained solely on data perform well in interpolation tasks when ample data are available. In practice, engineering applications often contain much less data than large machine learning projects from companies such as Google and OpenAI [28]. This is due to factors such as limited computational resources, the absence of benchmark datasets for calibrating automated frameworks, and the high specificity of each engineering problem. For these reasons, the injection and enhancement of neural networks with domain knowledge and physics theory is crucial. Machine learning models for engineering applications should not only regress and classify within the training domain but also respect the underlying physical laws, providing guidance in out-of-distribution areas. In this work, physics is embedded in two ways, through feature engineering and through loss terms. Techniques inspired by PINNs work, such as automatic differentiation and collocation point sampling, are applied.

Domain knowledge in this work derives from fundamental aerodynamic and fluid-mechanics principles. A notable example is NeuralFoil, which substitutes the low-fidelity XFOIL solver with a MLP trained on millions of airfoil simulations. Notably, NeuralFoil enhances model performance by preprocessing both input and output data to reflect established aerodynamic symmetries [29]. Specifically, it imposes even and odd symmetry constraints on the outputs - such as averaging to preserve symmetry in C_D and differencing for C_L - through evaluations of both the original and mirrored airfoil profiles at angles ±α. Motivated by this approach, the present methodology includes a trigonometric feature expansion of the angle of attack, as detailed in Eq. (5):

$$x_{AngleOfAttack} = x_a = \{a_{rad}, \sin(a), \cos(a), \sin^2(a), \cos^2(a)\} \tag{5}$$

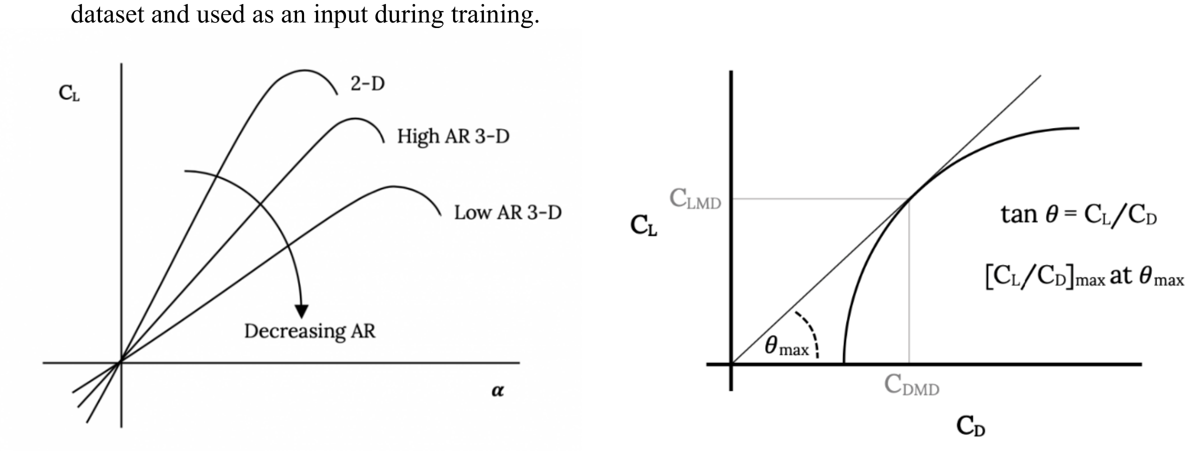
This directly provides the network with the mathematical components needed to capture the periodic, near-stall and post-stall behavior of lift and drag observed in experiments and detailed in Truong's work [30].

- Both Reynolds number and drag coefficient C_D are transformed into logarithmic space (i.e. the networks sees $\log_{10}(Re)$ and predicts $\log_{10}(C_D)$). This is done to reflect the well-established power-law relationship between drag and Reynolds number. This transformation converts the complex curve of the power law into a simple linear relationship, a scaling consistent with classical boundary layer models, such as Falkner–Skan and Schlichting, and essential for physically sound prediction in data-sparse regions [19].

All these transformations are applied consistently to both NN_{LF} and NN_{HF}, ensuring the models "see" physics informed representations of the core aerodynamic variables.

In addition to input/output transformations, physical principles are also used to constrain the network's learned behavior. A key difference between 2D and 3D aerodynamics lies in how lift is generated and limited by three-dimensional flow effects such as wingtip vortices and spanwise flow.

- The slope of the lift coefficient curve (C_{L_a}) decreases as the aspect ratio ($AR = {b^2}/{S}$) decreases. This is because lower AR wings produce stronger tip vortices, which induce downwash that reduces the effective angle of attack, thereby reducing lift. As a result, for the same geometric angle of attack, a low-AR wing generates less lift than a high-AR or 2D section. This relationship is visualized in Fig. 4.a. To enforce this trend in NN_{HF}, the C_{La} is computed using automatic differentiation since C_L is an output and angle of attack is an input. A second derivative with respect to aspect ratio, i.e. ${dC_{L_a}}/{dAR}$ is also computed automatically to evaluate whether the learned model respected this physical monotonicity.
- Another important physical relationship is that the maximum lift coefficient of a 3D wing $C_{L,max}$ is always lower than the maximum lift coefficient of its corresponding 2D airfoil ($C_{l,max}^{2D}$). This is again due to 3D effects like the influence of spanwise pressure gradients. In the case of multi-airfoil wings, such as those in this work, the Mean Aerodynamic Chord (MAC) is used as a reference airfoil. For each wing, a full angle-of-attack sweep is performed on the MAC airfoil using the publicly available NeuralFoil "*xxxlarge*" model, "at the Reynolds number specific to that wing. The extracted $C_{l,max}^{2D}$ is stored as part of the high-fidelity dataset and used as an input during training.



a) C_L vs α decreasing slope with decreasing AR.

b) C_L vs C_D – commonly referred as drag polar plot

Fig. 4 Fundamental aerodynamics theory for 2D and 3D lift surfaces[**].

Additionally, Fig. 4.b depicts the convexity of the drag polar and the monotonic increase of the drag coefficient as the angle of attack increases beyond a certain threshold. To reflect this aerodynamic behavior, a conservative cutoff of 10 degrees is selected, beyond which a physical loss penalty is applied to enforce increasing C_D. Like the previously discussed constraints, this rule is implemented as a soft penalty during NN_{HF} training. As with all penalty-based approaches, these physical laws are not treated as hard constraints - therefore, their satisfaction is not guaranteed and instead depends on the optimization dynamics of backpropagation. The final loss function is shown in Eq. (6).

$$\mathcal{L}_{HF} = \mathcal{L}_{data} + \lambda_{C_L}\mathcal{L}_{CL} + \lambda_{AR}\mathcal{L}_{AR} + \lambda_{C_D}\mathcal{L}_{C_D} \tag{6}$$

Note that, the terms shown in Eq. (6) are calculated as follows (Eqs. (6.a), (6.b) and (6.c)).

$$\mathcal{L}_{CL} = \frac{1}{N}\sum_{i=1}^{N}\left[\max\left(0, \widehat{C_L}(x_i) - C_{l,max}^{2D}(g_i)\right)\right]^2 \; where \; g_i = MAC \tag{6.a}$$

$$\mathcal{L}_{AR} = \frac{1}{N}\sum_{i=1}^{N}\left[\max\left(0, -\frac{\partial^2 \widehat{C_L}(x_i)}{\partial a \times \partial AR}\right)\right]^2 \tag{6.b}$$

[**] Courtesy: Virginia Tech "https://pressbooks.lib.vt.edu/aerodynamics/chapter/chapter-4/", accessed 10th June 2025.

$$\mathcal{L}_{C_D} = \frac{1}{N} \sum_{i:a_i > 10^o}^{N} \left[\max\left(0, -\frac{\partial \widehat{C_D}(x_i)}{\partial a}\right)\right]^2 \tag{6.c}$$

To extend the model's reliability into data-sparse regions, physics-based penalties are applied at high angles of attack where no CFD data exist. This approach mirrors the collocation point philosophy in PINNs. Specifically, during each training epoch, a random subset of 256 wings is sampled and evaluated over the 15°–30° angle of attack range, enforcing the physical constraints discussed earlier. Although these extrapolatory points lack ground-truth labels, the model is still trained to respect aerodynamic behavior through these soft penalties.

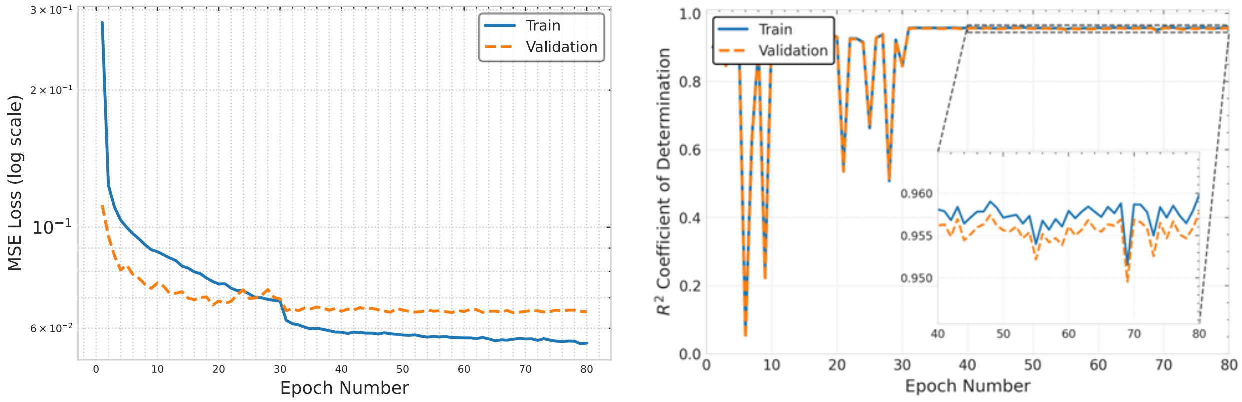
Embedding physical constraints is not only a regularization strategy, but is also essential for aerodynamic shape optimization. In many applications, such as takeoff and landing performance, the design goal often targets maximizing $C_{L,max}$. For such targets to be meaningful, the model must predict a realistic lift curve. That is, the curve must be linear at low angles of attack, peaking once near stall (i.e., unimodal), and then declining smoothly. Likewise, an aerodynamically plausible drag polar - smooth and convex - ensures that the lift-to-drag ratio exhibits a clear optimum, enabling robust optimization of cruise performance or endurance. Without these physically grounded trends, even high-accuracy models can produce misleading or non-actionable predictions when used for design.

V. Results and Discussion

This section presents the performance of the three distinct neural network models developed in this study: the low-fidelity network (NN_{LF}) trained on panel method data, the high-fidelity network (NN_{HF}) trained on CFD data, and the multi-fidelity architecture (MFNN) which combines both. The discussion emphasizes the impact of physics-based loss terms introduced in NN_{HF} and propagated to MFNN. Since physical constraints are not enforced in NN_{LF}, its presentation remains concise, focusing primarily on baseline predictive accuracy. For NN_{HF} and MFNN, results are illustrated using representative aerodynamic plots, such as lift curves and drag polars, alongside standard regression metrics - namely the coefficient of determination (R^2) and Mean Squared Error (MSE).

A. Low Fidelity Neural Network

The NN_{LF} is trained exclusively on low fidelity data generated using the VSPAERO panel solver. The dataset comprises approximately 150,000 samples, which enables the application of a conventional supervised training procedure without the need for additional architectural modifications to enhance performance. Fig. 5 shows the training progress of the loss function (Fig. 5.a) and the coefficient of determination R^2 (Fig. 5.b) over 80 epochs. Notably, R^2 exhibits significant fluctuations during the initial training stages, stabilizing rapidly around epoch 30 - a trend mirrored by a corresponding drop in the MSE loss curve. This early-stage instability is attributed to the drag-related outputs, specifically the challenge of learning a consistent relationship between total drag (C_D) and its components, induced drag ($\mathrm{C_{D,i}}$) and zero-lift drag ($C_{D,0}$). Preliminary experiments attempted to regularize the network by enforcing the additive relationship $C_D = C_{D,i} + C_{D,0}$ in the same way as described in Section IV.D, but this soft constraint did not improve performance and is not included in the final model.



a) Loss history

b) Aggregated R^2 for all aerodynamic coefficients

Fig. 5 Loss history and R^2 score per epoch for the NN_{LF}.

Table 3 summarizes the network's performance across all output quantities. The large dataset allows the model to generalize effectively, particularly in predicting lift (C_L) and pitching moment (C_M). However, notable discrepancies are observed in drag-related metrics. Considering these findings - and the critical role of accurate drag prediction in design studies - the NN_{LF}, although effective in general trend capture, is considered inadequate for standalone use in ASO studies. Nonetheless, since NN_{LF} serves as the low-fidelity component in a broader multi-fidelity framework, its predictive limitations are acceptable, provided that higher-fidelity corrections are applied downstream.

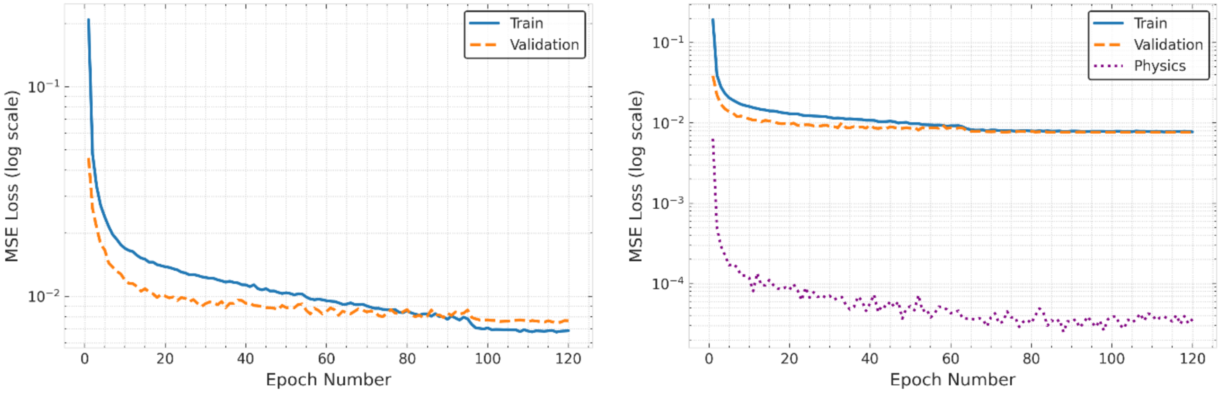
Table 3 Predictive performance metrics for the NN_{LF}.

Metric / Dataset	Train	Validation	Test
MSE (total)	0.0196	0.0283	0.036
R^2 (total)	0.96	0.956	0.95
R^2 (C_L)	0.988	0.988	0.987
R^2 (C_D)	0.946	0.947	0.954
R^2 (C_{Di})	0.926	0.930	0.939
R^2 (C_{D0})	0.936	0.931	0.933
R^2 (C_M)	0.986	0.981	0.984

B. High and Multi Fidelity Neural Networks

1. High Fidelity Neural Network – Physics Integration

Prior to incorporating the NN_{LF} into the multi-fidelity pipeline, an independent evaluation is conducted to assess the impact of physical regularization and feature engineering on the high-fidelity network NN_{HF}. The inputs related to wing geometry, flow conditions and output variables, as well as the network architecture, are kept constant across both cases. The only varying factor is the application of physics-informed enhancements. Figure 6 depicts the training and validation loss histories for the data-only NN_{HF} (Fig. 6.a) and the physics-enhanced NN_{HF} (Fig. 6.b). In both cases, the data loss converges rapidly toward a value near 10^{-2}, with a notable sudden decrease in training loss. For the data-only model, this drop occurs around Epoch 100, while in the physics-informed case it appears earlier, near Epoch 60. The physics loss component also decreases swiftly to approximately 10^{-5}, indicating that the model progressively learns to satisfy the soft physical constraints. An attempt was made to scale the physics loss to match the magnitude of the data loss (i.e., ~10^{-2}); however, this leads to a stronger competition between the two objectives and ultimately degrades overall performance.



a) Loss history of data-only NN_{HF}

b) Loss history of physics-enhanced NN_{HF}

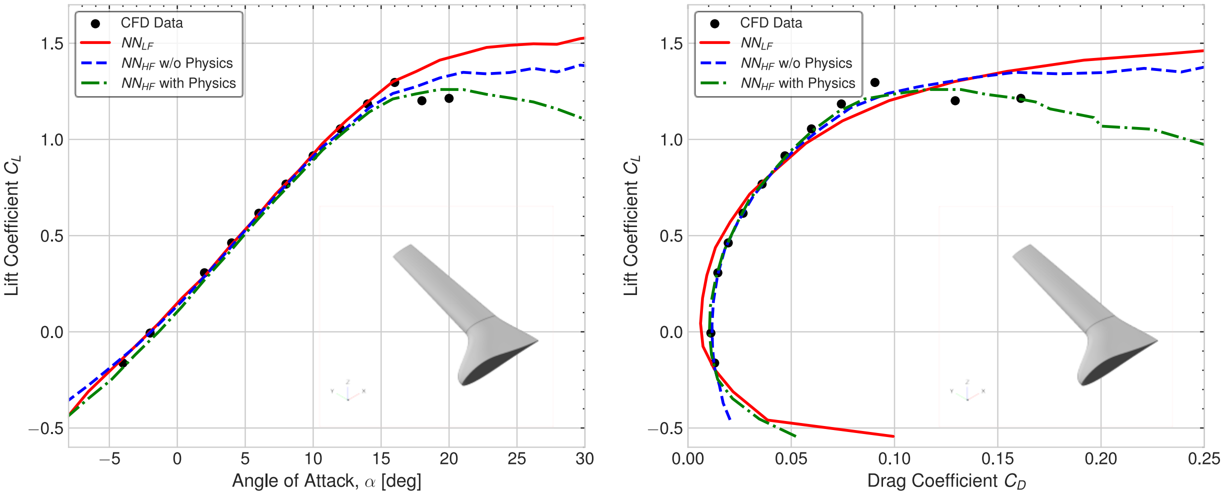
Fig. 6 Comparison between loss history of data-only NN_{HF} and physics-enhanced NN_{HF}.

Table 4 summarizes the performance of the two NN_{HF} models in terms of mean squared error (MSE) and coefficient of determination (R^2). Both networks demonstrate strong predictive capability, achieving high R^2 scores across all aerodynamic coefficients. However, in certain cases - particularly for the lift coefficient (C_L) - the inclusion of physics-based regularization appears to slightly hinder the learning process, likely due to the competing objectives of minimizing both the data loss and the physics loss simultaneously. Additionally, the dataset is dominated by samples in the linear region of the polar, which contributes to the consistently high R^2 values, as the physical constraints primarily influence behavior in the stall and post-stall regions. While statistical performance may suggest that further enhancement is unnecessary, this is misleading in the context of aerodynamic shape optimization, where accurate stall prediction is critical. Since the physics-based constraints specifically target lift and drag characteristics, the following analysis focuses on C_L and drag polars, omitting moment coefficient evaluation.

Table 4 Predictive performance metrics for the data-only NN_{HF}y and physics-enhanced NN_{HF}.

	Data – Only approach			Physics – Enhanced approach		
Metric / Dataset	**Train**	**Validation**	**Test**	**Train**	**Validation**	**Test**
MSE (total)	0.00233	0.00619	0.00645	0.00278	0.00626	0.00638
R^2 (total)	0.979	0.965	0.961	0.978	0.965	0.962
R^2 (C_L)	0.987	0.983	0.983	0.985	0.984	0.983
R^2 (C_D)	0.962	0.951	0.942	0.965	0.952	0.945
R^2 (C_M)	0.987	0.960	0.957	0.985	0.959	0.958

Figure 7 presents the lift curve (Fig. 7.a) across the angle-of-attack range [−5°, 30°] and the corresponding drag polar (Fig. 7.b) for a representative test-case wing, comparing the predictions of three models: NN_{LF}, the data-only NN_{HF}, and the physics-enhanced NN_{HF}. The black markers denote the ground truth CFD data. All models accurately capture the linear regime, including NN_{LF}, which is notable given its lower-fidelity training data. However, the failure of NN_{LF} to capture stall behavior is expected, as panel methods inherently lack the capability to model flow separation and post-stall phenomena. More critically, the data-only NN_{HF} - despite being trained on CFD data and exhibiting high R^2 scores - also fails to replicate the stall region accurately. It continues the upward trend of lift beyond stall, resulting in a monotonic C_L curve. This makes the identification of the true maximum lift coefficient $(C_{L,max})$ infeasible and undermines the model's applicability in design tasks where stall characteristics are pivotal. In contrast, the physics-enhanced NN_{HF} produces a distinct, physically realistic peak in lift, closely aligned with CFD data. The results highlight the importance of physics integration in ensuring realistic and actionable predictions.

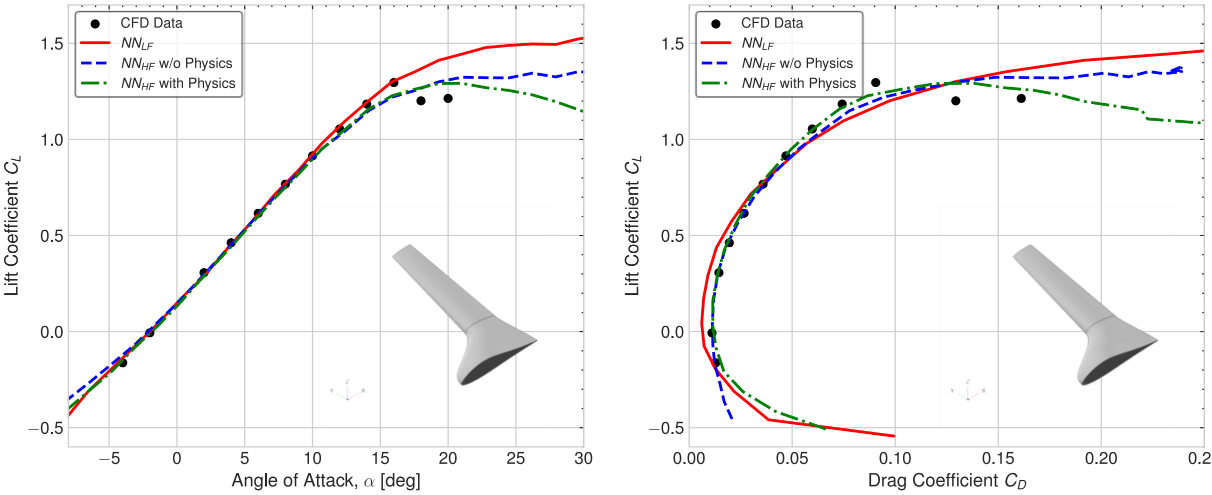


a) Lift curve predictions vs CFD

b) Drag polar predictions vs CFD

Fig. 7 Polar predictions for NN_{LF}, and the two NN_{HF} for a test wing.

Similar trends are observed in the drag polar plot. NN_{LF} under-predicts the zero-lift drag coefficient $C_{D,0}$ resulting in a noticeably less accurate drag curve. In contrast, both NN_{HF} models offer more realistic polars that align more closely with the ground truth. Notably, the physics-enhanced NN_{HF} captures the stall-induced sudden change in the drag curve more convincingly, whereas the data-only NN_{HF} exhibits an early saturation in drag at low angles of attack, deviating from expected aerodynamic behavior. To assess the extrapolative capabilities of the high-fidelity models, an additional experiment is conducted in which the training dataset is truncated to include only angles of attack up to 16°. The remaining high-angle data points $(18° - 20°)$ are held out entirely during training. Under this constraint, in Fig. 8, the physics-enhanced NN_{HF} demonstrates superior performance. It maintains a smooth, physically plausible drag curve (Fig. 8.b) and successfully models the stall region, consistent with the imposed ($^{\partial C_D}/_{\partial \alpha} > 0$) regularization beyond the linear regime. In contrast, the data-only NN_{HF} produces a distorted and non-monotonic drag polar at high angles, highlighting its instability and poor generalization when faced with out-of-distribution inputs.

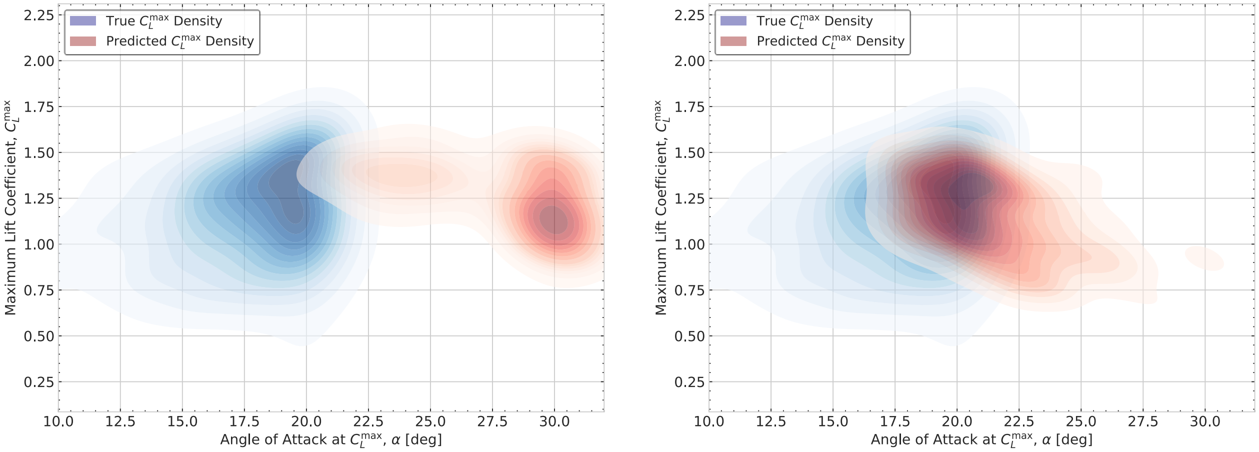


a) Lift curve predictions vs CFD

b) Drag polar predictions vs CFD

Fig. 8 Predictions for NN_{LF}, and the two NN_{HF} for a test wing. NN_{HF} models have been trained with angles of attack up to 16^o.

A broader view of stall behavior across the entire test dataset is provided in Fig. 9, which displays the distributions of predicted maximum lift coefficient ($C_{L,max}$) and corresponding stall angle ($\alpha_{@C_{Lmax}}$) for all wings of the test set. Figure 9.a shows predictions from the data-only NN_{HF}, while Fig. 9.b shows results from the physics-enhanced version. Both models are trained using the extrapolative setup described earlier, where training is limited to angles of attack below 16°. The data-only NN_{HF} consistently fails to capture the stall phenomenon. For the majority of wings, it predicts a $C_{L,max}$ occurring at the upper bound of the inference range (30°), resulting in an unrealistic, ever-increasing lift curve. In contrast, the physics-informed NN_{HF} exhibits a much more accurate distribution, correctly capturing a spread of $C_{L,max}$ values and corresponding stall angles. While deviations of $2° - 4°$ from the true stall point are observed, the model's behavior remains physically consistent across the dataset. These results reinforce the importance of physics-based constraints in extending the neural network's reliability to unseen flow conditions, particularly for ASO scenarios where stall characteristics play a defining role.



a) Data-only NN_{HF}

b) Physics-enhanced NN_{HF}

Fig. 9 True CFD and predicted density for $C_{L,max}$ and respective angle of attack regarding the test set.

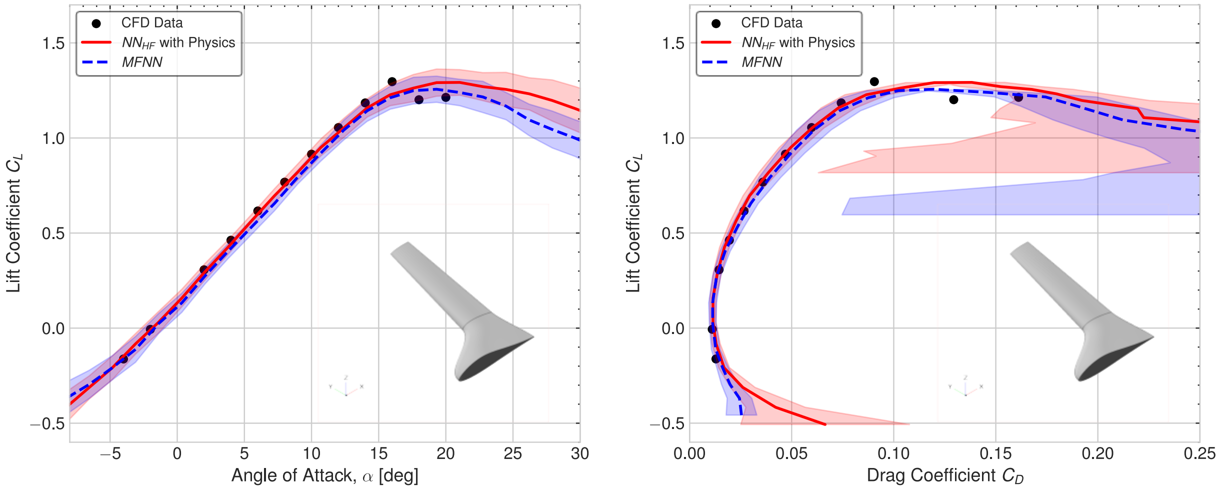
2. *Multi Fidelity Neural Network*

The integration of the NN_{HF} into the multi-fidelity framework follows the architecture illustrated in Fig. 3 and previously outlined in Section IV.C. For this coupled configuration, the physics-enhanced NN_{HF} is selected, having demonstrated superior performance over its data-only counterpart. During training, each high-fidelity sample passes through NN_{LF} in inference mode 100 times to compute the low-fidelity predictions of both the mean and standard deviation, which are then appended to the input vector of NN_{HF}. The training process for the resulting multi-fidelity neural network MFNN employs the same hyper-parameters as those used for the standalone NN_{HF}. Table 5 summarizes the performance of the MFNN model across all subsets. Compared to the physics-enhanced NN_{HF}, MFNN exhibits a modest improvement in R^2 for all targets and a slight reduction in MSE. However, given that R^2 are already near unity, it becomes necessary to examine the predicted C_L behavior more closely – particularly near stall – where purely statistical metrics may obscure critical aerodynamic discrepancies.

Table 5 Predictive performance metrics for the MFNN across all subsets.

Metric / Dataset	Train	Validation	Test
MSE (total)	0.00215	0.00581	0.00630
R^2 (total)	0.982	0.968	0.964
R^2 (C_L)	0.988	0.985	0.984
R^2 (C_D)	0.970	0.956	0.950
R^2 (C_M)	0.989	0.963	0.958

Figure 10 revisits the same representative test wing, comparing the predictions of physics-enhanced NN_{HF} and the MFNN against ground truth CFD data. Both models are trained under the extrapolative scenario, where angle of attack values above 16^o are excluded from the training set. Uncertainty estimates – computes as ±1 standard deviation from MCD over 100 forward passes – also shown. The MFNN exhibits further improvement in the predicted C_L curve near stall, offering a sharper and earlier drop in lift and more accurately identifying the $a_{@C_{L,max}}$. Notably, its uncertainty bounds tightly enclose the CFD data throughout the entire range. In the drag polar plot, MFNN predictions remain closer to the true aerodynamic trend and maintain lower uncertainty at high angles of attack, despite lacking direct exposure to data in that regime. This suggests that low-fidelity information, in combination with physics-based regularization, supports more physically consistent generalization. By contrast, the standalone physics NN_{HF} becomes increasingly uncertain beyond 20^o, and its drag predictions deviate significantly, at times producing implausible or non-monotonic behavior.

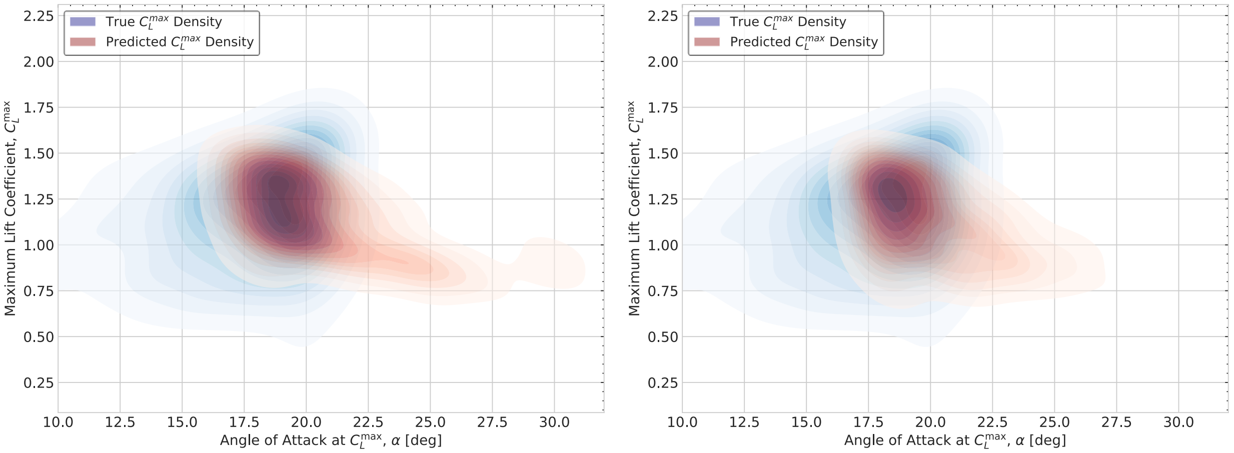


a) Lift curve vs CFD

b) Drag polar vs CFD

Fig. 10 Mean and standard deviation predictions for physics NN_{HF} and MFNN for a test wing. Both models have been trained with angles of attack up to 16^o.

Finally, Fig. 11 illustrates the predictive performance of MFNN in the stall region, expressed as the joint distribution of $C_{L,max}$ and the corresponding stall angle $\alpha_{@C_{L,max}}$, under the extrapolative training scenario.



a) Only feature - engineering physics MFNN

b) Full physics integration along with feature engineering MFNN

Fig. 11 True CFD and predicted density for $C_{L,max}$ and respective angle of attack

Two configurations are compared. More specifically, Fig. 11.a shows results from the MFNN trained with feature engineering alone, while Fig. 11.b includes both feature engineering and the full set of physics-based loss regularizations described in Section IV.D. All other hyper-parameters are held constant. In both cases, the predicted distributions shift closer to the ground truth compared to the physics-only NN_{HF}, highlighting the benefit of incorporating low-fidelity information. However, the model trained with full physics integration clearly demonstrates superior accuracy. It successfully anticipates stall behavior for nearly all test wings, avoiding the unrealistic, monotonically increasing lift curves seen in less constrained models. Furthermore, the spread and magnitude of predicted $C_{L,max}$ values closely match those of the CFD reference, reinforcing the importance of combining multi-fidelity inputs with physics-informed learning to achieve reliable and actionable aerodynamic predictions.

VI. Conclusions

This work demonstrates the significant benefits of enhancing neural networks with physics-based regularization, combining soft constraints through loss function penalties and physics-informed feature engineering. Three neural network architectures are developed and trained on aerodynamic data for UAV wings operating in the low subsonic regime (Mach < 0.3), using low-fidelity simulations from the OpenVSP VSPAERO panel method and high-fidelity data from a RANS-based CFD solver. A low-fidelity neural network (NN_{LF}) is trained independently and then coupled with a high-fidelity neural network (NN_{HF}), forming a serial multi-fidelity model (MFNN). Physics-based regularization is applied to the high-fidelity components to improve generalization and physical realism, especially in underrepresented regions of the design space.

The incorporation of physical constraints significantly improves the neural networks' ability to capture critical aerodynamic phenomena, most notably stall behavior. The physics-enhanced models successfully predict C_L curves that exhibit linearity in the low-angle regime and unimodal behavior near stall, enabling accurate estimation of $C_{L,max}$ and its associated stall angle within $1° - 3°$ of the CFD reference. Notably, these improvements are achieved without modifying the underlying neural network architecture. The proposed method relies solely on fundamental aerodynamic principles and does not require access to flow-field data. Computational overhead remains minimal – complete inference for a full polar with 100 Monte Carlo Dropout samples requires approximately 0.13 seconds per wing – and the implementation is straightforward, using established tools such as automatic differentiation. As such, the methodology presented here offers a practical and robust enhancement to neural network-based aerodynamic modeling, especially in scenarios requiring reliable predictions across the full operational envelope.

Future developments will aim to extend the methodology by incorporating regularization terms for the moment coefficient (C_M) to ensure greater aerodynamic completeness. In addition, hard physical constraints will be introduced through architectural modifications, such as the use of deep lattice layers to enforce monotonicity in the C_L curve. Further investigation will also include ablation studies on the ratio of low to high-fidelity training samples, in order to assess the sensitivity of the model to multi-fidelity composition and identify optimal data allocation strategies.

Acknowledgments

The research work was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the Basic Research Financing (Horizontal support for all Sciences), National Recovery and Resilience Plan (Project Number: 016429, Project Acronym: INDIANA). Results presented in this work have been produced using the Aristotle University of Thessaloniki (AUTh) High Performance Computing Infrastructure and Resources.

References

[1] Shakhatreh, H., Sawalmeh, A. H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N. S., Khreishah, A., and Guizani, M., "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges," *IEEE Access*, Vol. 7, 2019, pp. 48572–48634. doi: 10.1109/ACCESS.2019.2909530

[2] Conlan-Smith, C., Ramos-García, N., Sigmund, O., and Andreasen, C. S., "Aerodynamic Shape Optimization of Aircraft Wings Using Panel Methods," *AIAA Journal*, Vol. 58, No. 9, 2020, pp. 3765–3776. https://doi.org/10.2514/1.J058979

[3] Wang, K., Han, Z., Zhang, K., and Song, W., "Efficient Global Aerodynamic Shape Optimization of a Full Aircraft Configuration Considering Trimming," *Aerospace 2023, Vol. 10, No. 8, 2023, p. 734*. doi: 10.3390/AEROSPACE10080734

[4] Brunton, S. L., Kutz, J. N., Manohar, K., Aravkin, A. Y., Morgansen, K., Klemisch, J., Goebel, N., Buttrick, J., Poskin, J., Blom-Schieber, A. W., Hogan, T., and McDonald, D., "Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning," *AIAA Journal*, Vol. 59, No. 8, 2021, pp. 2820–2847. doi: 10.2514/1.J060131

[5] Forrester, A. I. J., and Keane, A. J., "Recent Advances in Surrogate-Based Optimization," *Progress in Aerospace Sciences*, Vol. 45, Nos. 1–3, 2009, pp. 50–79. doi: 10.1016/J.PAEROSCI.2008.11.001

[6] Kennedy, B. M. C., and O'hagan, A, "Predicting the Output from a Complex Computer Code When Fast Approximations Are Available," *Biometrika*, Vol. 87, No. 1, 2000, pp. 1–13. doi: 10.1093/biomet/87.1.1

[7] Yang, A., Li, J., and Liem, R. P., "Multifidelity Data-Driven Aerodynamic Shape Optimization of Wings with Composite Neural Networks," *AIAA Aviation and Aeronautics Forum and Exposition, AIAA AVIATION Forum 2023*, 2023. doi: 10.2514/6.2023-3470

[8] Du, B., Shen, E., Wu, J., Guo, T., Lu, Z., and Zhou, D., "Aerodynamic Prediction and Design Optimization Using Multi-Fidelity Deep Neural Network," *Aerospace*, Vol. 12, No. 4, 2025, p. 292. doi: 10.3390/AEROSPACE12040292/S1

[9] MOHAMMAD ZADEH, P., and SAYADI, M., "An Efficient Aerodynamic Shape Optimization of Blended Wing Body UAV Using Multi-Fidelity Models," *Chinese Journal of Aeronautics*, Vol. 31, No. 6, 2018, pp. 1165–1180. doi: 10.1016/j.cja.2018.04.004

[10] Eivazi, H., Tahani, M., Schlatter, P., and Vinuesa, R., "Physics-Informed Neural Networks for Solving Reynolds-Averaged Navier-Stokes Equations," *Physics of Fluids*, Vol. 34, No. 7, 2022. doi: 10.1063/5.0095270

[11] Zobeiry, N., and Poursartip, A., "Theory-Guided Machine Learning for Process Simulation of Advanced Composites," 2021.

[12] Zobeiry, N., Reiner, J., and Vaziri, R., "Theory-Guided Machine Learning for Damage Characterization of Composites," *Composite Structures*, Vol. 246, 2020, p. 112407. doi: 10.1016/J.COMPSTRUCT.2020.112407

[13] Zhao, J., Zeng, L., Lin, A., and Shao, X., "Deep Learning Prediction Method for Aerodynamic Forces on Morphing Aircraft Considering Physical Monotonicity," *Advances in Aerodynamics* , Vol. 7, No. 1, 2025, pp. 1–16. doi: 10.1186/s42774-024-00197-x

[14] Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," *Journal of Computational Physics*, Vol. 378, 2019, pp. 686–707. doi: 10.1016/j.jcp.2018.10.045

[15] Pliakos, C., Efrem, G., Terzis, D., and Panagiotou, P., "An Automated Framework for Streamlined CFD-Based Design and Optimization of Fixed-Wing UAV Wings," *Algorithms 2025, Vol. 18, No. 4, 2025, p. 186*. doi: 10.3390/A18040186

[16] Chen, W., Chiu, K., and Fuge, M. D., "Airfoil Design Parameterization and Optimization Using Bézier Generative Adversarial Networks," *AIAA Journal*, Vol. 58, No. 11, 2020, pp. 4723–4735. doi: 10.2514/1.J059317

[17] Selig, M. S., "UIUC Airfoil Coordinates Database," https://m-selig.ae.illinois.edu/ads/coord_database.html. [retrieved 10 June 2025]

[18] Mitridis, D., Kapsalis, S., Terzis, D., and Panagiotou, P., "An Evaluation of Fixed-Wing Unmanned Aerial Vehicle Trends and Correlations with Respect to NATO Classification, Region, EIS Date and Operational Specifications," *Aerospace*, Vol. 10, No. 4, 2023. doi: 10.3390/aerospace10040382

[19] Raymer, D., "Aircraft Design: A Conceptual Approach, Sixth Edition," American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 2018. doi: 10.2514/4.104909

[20] Antoniou, S., Kapsalis, S., Panagiotou, P., and Yakinthos, K., "Parametric Investigation of Leading-Edge Slats on a Blended-Wing-Body UAV Using the Taguchi Method," *Aerospace 2023, Vol. 10, No. 8, 2023, p. 720*. doi: 10.3390/AEROSPACE10080720

[21] Verstraete, D., Palmer, J. L., and Hornung, M., "Preliminary Sizing Correlations for Fixed-Wing Unmanned Aerial Vehicle Characteristics," *Journal of Aircraft*, Vol. 55, No. 2, 2017, pp. 715–726. doi: 10.2514/1.C034199

[22] Sheridan, C. N. D., Pham, D. D. V, and Whiteside, S. K. S., “Evaluation of VSPAERO Analysis Capabilities for Conceptual Design of Aircraft with Propeller-Blown Wings,” *AIAA AVIATION Forum*, 2021. doi: 10.2514/6.2021-2510

[23] Adler, E. J., Christison Gray, A., and Martins, J. R. R. A., “TO CFD OR NOT TO CFD? COMPARING RANS AND VISCOUS PANEL METHODS FOR AIRFOIL SHAPE OPTIMIZATION,” *33rd ICAS Congress 2022*, 2022.

[24] Guo, M., Manzoni, A., Amendt, M., Conti, P., and Hesthaven, J. S., “Multi-Fidelity Regression Using Artificial Neural Networks: Efficient Approximation of Parameter-Dependent Output Quantities,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 389, 2022, p. 114378. doi: 10.1016/J.CMA.2021.114378

[25] Gal, Y., and Ghahramani, Z., “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning,” 33rd International Conference on Machine Learning, ICML 2016, Vol. 3, 2015, pp. 1651–1660.

[26] Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M., “Optuna: A Next-Generation Hyperparameter Optimization Framework,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 2623–2631. doi: 10.1145/3292500.3330701

[27] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *Advances in Neural Information Processing Systems*, Vol. 32, 2019.

[28] Alzubaidi, L., Bai, J., Al-Sabaawi, A., Santamaría, J., Albahri, A. S., Al-dabbagh, B. S. N., Fadhel, M. A., Manoufali, M., Zhang, J., Al-Timemy, A. H., Duan, Y., Abdullah, A., Farhan, L., Lu, Y., Gupta, A., Albu, F., Abbosh, A., and Gu, Y., “A Survey on Deep Learning Tools Dealing with Data Scarcity: Definitions, Challenges, Solutions, Tips, and Applications,” *Journal of Big Data*, Vol. 10, No. 1, 2023, pp. 1–82. doi: 10.1186/S40537-023-00727-2

[29] Sharpe, P., and Hansman, R. J., “NeuralFoil: An Airfoil Aerodynamics Analysis Tool Using Physics-Informed Machine Learning,” 2025.

[30] Truong, V. K., “An Analytical Model for Airfoil Aerodynamic Characteristics over the Entire 360° Angle of Attack Range,” *Journal of Renewable and Sustainable Energy*, Vol. 12, No. 3, 2020. doi: 10.1063/1.5126055