Reinforcement Learning for Practical Airfoil Design Optimization

Andrew Smith

I. Problem Statement

AIRFOIL shape optimization is a fundamental challenge in aerodynamics due to the nonlinear and high-dimensional characteristics of fluid mechanics. The goal is to design two-dimensional airfoil shapes that maximize desirable aerodynamic metrics such as lift-to-drag ratio while minimizing penalties such as excessive drag. Traditional optimization methods, including gradient-based algorithms, evolutionary algorithms, and particle swarm optimization, have achieved varying degrees of success. However, these approaches are often computationally expensive when coupled with high-fidelity solvers, prone to convergence on local optima, and data-inefficient in that they do not build on prior evaluations to guide future optimization.

Geometric parameterization adds another layer of complexity. While classical families like the NACA 4-digit series provide simplicity, they confine optimization to a limited design space. Bézier curves, by contrast, provide a flexible and smooth representation of airfoil geometries, allowing for localized and interpretable shape modifications. This flexibility introduces a richer search space but also necessitates an optimization method capable of navigating it effectively.

Reinforcement learning (RL) is particularly well suited to this problem because it frames airfoil optimization as a sequential decision-making process. An RL agent can iteratively modify control points of a Bézier curve, evaluate aerodynamic performance using a solver, and use feedback to refine its policy. Unlike supervised learning, RL does not depend on pre-existing datasets that might bias outcomes or limit generative potential. Instead, it learns from direct interaction with the environment, building knowledge through trial and error. RL also naturally accommodates continuous action spaces, which is essential for smooth geometric modifications in Bézier curve parameterizations.

II. Motivation

Reducing aerodynamic drag and increasing efficiency are critical for improving aircraft performance and sustainability. Studies estimate that reducing drag can cut fuel burn by up to 20 to 25 percent, with substantial implications for both cost savings and carbon emissions in the aviation industry. In the context of the global push for sustainable aviation, efficient airfoil optimization is both timely and impactful.

Beyond aviation, efficient aerodynamic design benefits renewable energy systems such as wind turbines, where optimized blade profiles can significantly enhance power generation. The relevance of airfoil optimization therefore extends across multiple sectors where fluid–structure interactions play a central role.

A. Smith was with the Department of Mechanical Engineering, Rochester Institute of Technology, Rochester, NY, 14623 USA e-mail: (as5652@rit.edu).

III. Environment Details

A. State Space

The state space consists of eight continuous values representing the y-coordinates of the fixed x-coordinate Bézier control points defining the airfoil geometry. Together, these values describe the current airfoil shape and thickness distribution. The state vector fully encodes the design parameters relevant to the aerodynamic optimization task and is normalized to a bounded range to improve numerical conditioning during learning. The x-coordinates remain constant and uniformly spaced along the chord from 0 to 1, allowing the state representation to focus exclusively on vertical shape modulation. The initial airfoil state is defined by the vector

$$[0.1,\ 0.1,\ 0.1,\ 0.1,\ -0.1,\ -0.1,\ -0.1,\ -0.1]$$

which represents a symmetric baseline profile with mirrored displacements on the upper and lower surfaces.

B. Action Space

The action space is continuous and eight-dimensional, reflecting the agent's ability to adjust each Bézier control point vertically at every time step. Each element of the action vector corresponds to an incremental displacement applied to the y-coordinate of a control point. These updates allow for controlled modifications to airfoil camber and local thickness distributions. Action magnitudes are bounded to prevent overly aggressive geometry changes that could destabilize learning or produce non-physical shapes. Additional geometric constraints are enforced after each action to ensure smoothness, prevent self-intersections, maintain a closed trailing edge, and enforce minimum thickness requirements. This study explores a single action per episode restricted to $\pm$ 0.15 for each control point. Distance between control points with the same x-coordinate cannot fall below 0.025.

C. XFOIL Solver

Aerodynamic evaluation is conducted using the XFOIL solver, a panel-method-based computational tool widely used for two-dimensional airfoil analysis. For each candidate airfoil geometry generated by either RL agents or the genetic algorithm (GA), the discretized Bézier coordinates are formatted into an input file and passed to XFOIL via automated Python system calls.

The primary input parameters to the solver include the airfoil geometry coordinates, a prescribed angle of attack or range of angles of attack, and a fixed Reynolds number chosen to represent the operating flight regime. Additional solver settings control boundary-layer modeling and convergence tolerance to ensure consistent analysis across simulations. For this study, angles of attack range from 5 to 7 deg and a Reynold's number of 1,000,000 is used.

XFOIL outputs aerodynamic performance coefficients, most notably the lift coefficient (C_L) and drag coefficient (C_D), which are parsed from the solver output files by the Python interface. These coefficients are used to compute the lift-to-drag ratio (L/D), which serves as the principal performance metric and reward signal for RL training and for GA fitness evaluation.

D. Reward Function

The reward function is directly tied to the aerodynamic optimization objective and is computed using XFOIL simulations invoked through Python. After the agent applies an action to modify the Bézier control points defining the airfoil shape, the updated geometry is passed to XFOIL, which evaluates aerodynamic performance under specified flow conditions. The reward is defined as the resulting lift-to-drag ratio (L/D) extracted from XFOIL's output. Positive reward values reinforce shape modifications that improve aerodynamic efficiency, while lower reward values penalize regressions away from optimal designs. For alternative optimization tasks, such as minimizing the drag coefficient or maximizing the lift coefficient, the reward function may be redefined using the corresponding output parameters reported by XFOIL. In cases where generated airfoil geometries are non-physical or lead to solver convergence failures, the simulation is skipped and a reward of zero is assigned, discouraging exploration into impractical or unstable regions of the design space.

IV. Algorithms

A. Soft Actor-Critic (SAC)

A fundamental actor–critic algorithm that learns both a stochastic policy and a value function while explicitly encouraging exploration through an entropy-maximization objective. In this work, the Stable-Baselines3 implementation of SAC is used. The continuous action space of Bézier control point adjustments can be handled directly, eliminating the need for discretization. The actor network outputs a probability distribution over control point movements, while twin critic networks estimate expected returns for sampled actions. This approach provides a robust baseline for comparison and builds intuition for how policy-based RL can interact with aerodynamic optimization while balancing performance and exploration.

B. Deep Deterministic Policy Gradient (DDPG)

A deterministic actor–critic algorithm designed for continuous control tasks. In this work, the Stable-Baselines3 implementation of DDPG is used. The continuous action space of Bézier control point adjustments is handled directly through a parameterized actor network that outputs precise control actions, while a critic network estimates the expected return for state–action pairs. Target networks and experience replay are employed to stabilize training. This approach provides a complementary baseline to compare against stochastic methods and builds intuition for how deterministic policy learning can be applied to aerodynamic optimization tasks requiring fine-grained, continuous control.

C. Genetic Algorithm (GA)

A population-based optimization method inspired by natural evolution. To apply a GA implemented using PyGAD, each candidate solution encodes a set of Bézier control point parameters representing an airfoil shape. The population is evolved over successive generations using selection, crossover, and mutation operators, with fitness defined by aerodynamic performance metrics. This approach provides a strong, reinforcement-learning-free baseline for comparison and helps contextualize the effectiveness of policy-based and value-based RL methods in aerodynamic optimization.

V. Results

A. Soft Actor-Critic (SAC)

The SAC algorithm was evaluated using the custom airfoil optimization environment integrated with XFOIL. Because each environment step required a full aerodynamic simulation, computational expense was the dominant constraint governing experimental design. As a result, the total number of training interactions was limited to 4,096 steps, which is significantly lower than typical deep RL experiments. Hyperparameter selection therefore prioritized rapid learning, stability, and efficient use of limited data rather than long-horizon asymptotic performance.

To improve gradient estimation with the sparse and noisy reward signals produced by aerodynamic simulations, the batch size was increased to 128 samples per update, allowing the critic networks to average over more diverse transitions and improve stability relative to smaller default minibatches.

The replay buffer size ensured that all collected experiences could be reused efficiently due to the low total interaction count. This mitigated early overfitting to recent samples and promoted better generalization across the small number of available aerodynamic evaluations. Learning was delayed by setting the learning start parameter to 512 steps, preventing premature updates before the replay buffer contained a sufficiently diverse dataset of airfoil shapes.

Despite the extreme limit on total training interactions, SAC demonstrated stable learning behavior and consistent improvement in lift-to-drag ratios across evaluation runs, reaching a peak of 121.59. The algorithm's entropy-regularized objective encouraged continuous exploration of the airfoil design space, enabling the agent to identify higher-performing geometries even with a small sample budget. These results indicate that SAC is well-suited to simulation-expensive optimization problems, as it effectively balances exploration and exploitation while maintaining sample efficiency under strict computational constraints.

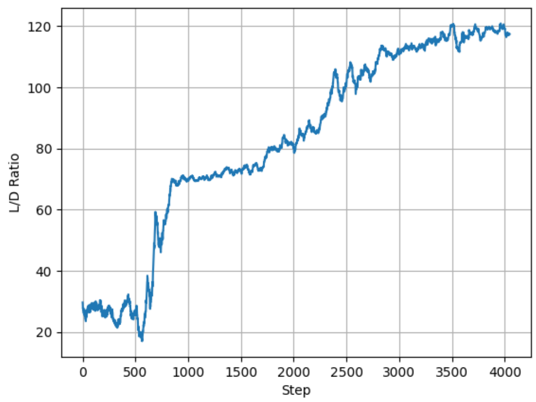

Fig. 1: Convergence of airfoil performance metrics using SAC.

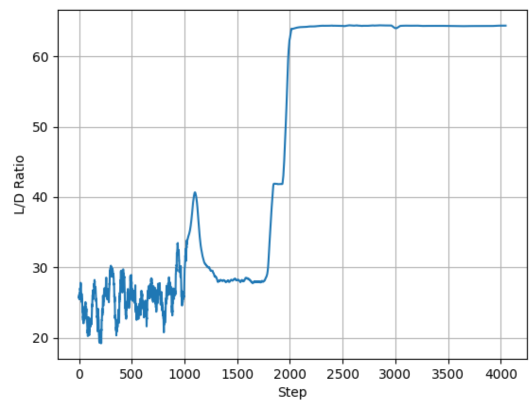

Fig. 2: Convergence of airfoil performance metrics using DDPG.

B. Deep Deterministic Policy Gradient (DDPG)

The DDPG algorithm was evaluated using the same custom XFOIL-integrated airfoil optimization environment as SAC. As with all experiments, the dominant constraint was computational expense, since each environment interaction required a full aerodynamic solver call. Consequently, the total training budget was restricted to 4,096 environment steps, necessitating hyperparameter choices that favored data efficiency, training stability, and rapid early learning over long-run convergence performance.

Training updates were delayed until 1,024 environment steps had been collected, a larger warm-up period than is typically used. This decision ensured that the replay buffer contained sufficiently diverse experiences before learning commenced, improving the stability of critic learning in the early stages. Given the limited number of total interactions, this tradeoff reduced noisy updates and promoted more reliable policy improvement once learning began.

The batch size was increased to 128 transitions per update. Using larger minibatches improves gradient estimation stability when rewards are noisy and sparse, as is common in aerodynamic simulations subject to solver convergence variability. This choice helped prevent large oscillations in critic estimates while allowing the actor to receive smoother learning signals with fewer total update steps.

All other DDPG hyperparameters were maintained at their default values in order to minimize unnecessary tuning and isolate the impact of core algorithmic differences rather than configuration optimization. Training was performed with one gradient update per environment interaction and default target network update parameters, providing a consistent comparison framework with SAC while keeping total computational cost low.

Although DDPG is known to be less exploration-efficient than entropy-regularized methods, its deterministic policy allowed it to rapidly exploit high-performing regions of the airfoil design space once identified. Under the limited training budget, DDPG achieved steady improvements in lift-to-drag ratio, reaching a peak of 66.72, though its convergence behavior exhibited higher sensitivity to initialization than SAC. This aligns with known characteristics of the algorithm: strong local optimization once promising solutions are discovered but increased risk of premature convergence due to less inherent exploration. These results underline the tradeoff between deterministic exploitation and stochastic exploration in simulation-expensive aerodynamic optimization tasks.

C. Genetic Algorithm (GA)

A population-based GA implemented using PyGAD was employed as a baseline optimization method for the airfoil design problem. As with the RL experiments, computational expense was the primary limiting factor, since each fitness evaluation required an independent XFOIL simulation. Consequently, the GA configuration was chosen to balance population diversity with a manageable number of total aerodynamic evaluations.

The evolutionary process was run for 64 generations with a population size of 64 individuals per generation, resulting in a maximum of 4,096 candidate airfoil evaluations. This total was deliberately matched to the training budgets used for SAC and DDPG, enabling a fair comparison across methods under identical computational constraints. Each individual encoded an eight-dimensional vector corresponding to the Bézier control point y-coordinates defining an airfoil geometry.

Tournament selection was employed as the parent selection strategy with a tournament size of four. This approach balances selection pressure and population diversity by favoring higher-performing candidates without eliminating weaker individuals too aggressively. Maintaining moderate selection pressure was especially important given the small optimization budget, as excessive exploitation could lead to premature convergence to suboptimal geometries without sufficient exploration of alternative airfoil shapes.

A crossover probability of 0.85 was applied, promoting strong genetic mixing between parent solutions to generate novel shape combinations that inherit beneficial traits from multiple candidates. This high crossover rate encourages broad search early in optimization, helping the algorithm efficiently explore the geometric design space despite the limited number of generations.

Mutation was applied with a probability of 0.08, introducing random perturbations to control point values to preserve population diversity and allow escape from local optima. The chosen mutation rate was large enough to ensure exploration of the design space while remaining small enough to avoid destabilizing high-quality solutions during later generations.

Elitism was enabled by preserving the top two individuals from each generation unchanged. This ensured that the best-performing airfoils discovered were retained throughout the evolutionary process, preventing regression in solution quality while evolutionary search continued around these elite candidates.

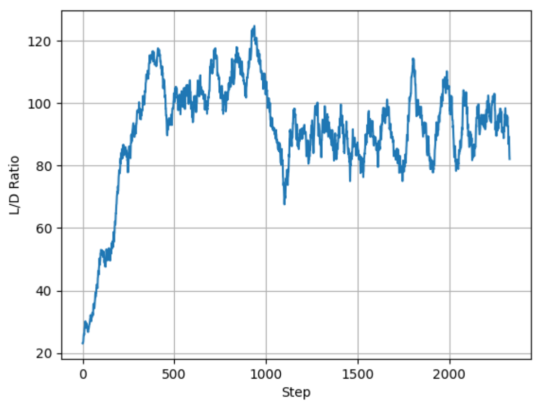

Fig. 3: Convergence of airfoil performance metrics using GA.

Under these settings, the GA demonstrated improvement in aerodynamic performance across generations, achieving a peak lift-to-drag ratio of 124.38. Its population-based exploration strategy enabled broad coverage of the design space from the initial symmetric baseline, though convergence was slower and more stochastic relative to SAC due to the absence of gradient-based learning. Nonetheless, the GA provided a strong baseline for comparison, highlighting the fundamental tradeoff between evolutionary global search methods and RL approaches for computationally expensive aerodynamic optimization problems.

VI. Discussion

A. Summary of Results

The performance of the optimization algorithms was evaluated in terms of the lift-to-drag ratio (L/D) achieved under the given simulation conditions. The initial symmetric Bézier airfoil had a baseline L/D of 24.24, reflecting its unoptimized geometry. All algorithms demonstrated significant improvements over this starting point, highlighting their ability to explore the design space and enhance aerodynamic efficiency even within the constrained parameterization.

Among the RL approaches, SAC achieved an L/D of 121.59, markedly higher than the initial airfoil and roughly double the performance achieved by DDPG, which reached an L/D of 66.72. The superior performance of SAC can be attributed to its stochastic policy and entropy-regularized objective, which promoted consistent exploration of promising geometries despite the limited number of environment interactions. DDPG, by contrast, employed a deterministic policy that focused on exploitation, resulting in faster convergence toward local optima but reduced exploration capability. These characteristics likely contributed to its comparatively lower L/D value.

The GA achieved the highest L/D of 124.38; however, the algorithm did not converge fully to this value within the allocated 64 generations and population size. The population-based search enabled broader exploration across the design space and facilitated discovery of high-performing solutions, though the limited computational budget prevented the refinement of the top candidates to fully exploit the potential aerodynamic gains.

When compared to established industry-standard airfoils under identical flow conditions, the optimized designs approach but do not surpass top-performing benchmarks. For reference, the SG6043 airfoil achieves an L/D of 128.79 and the NACA 6409 achieves 144.85 at the same angle of attack and Reynolds number. The gap between the optimized Bézier airfoils and these standards can be attributed to multiple factors, including the limited degrees of freedom in the Bézier parameterization (only eight points moving vertically within a ± 0.15 range), the coarse control over chordwise and trailing-edge shaping, and the restricted computational budget limiting the total number of simulation evaluations.

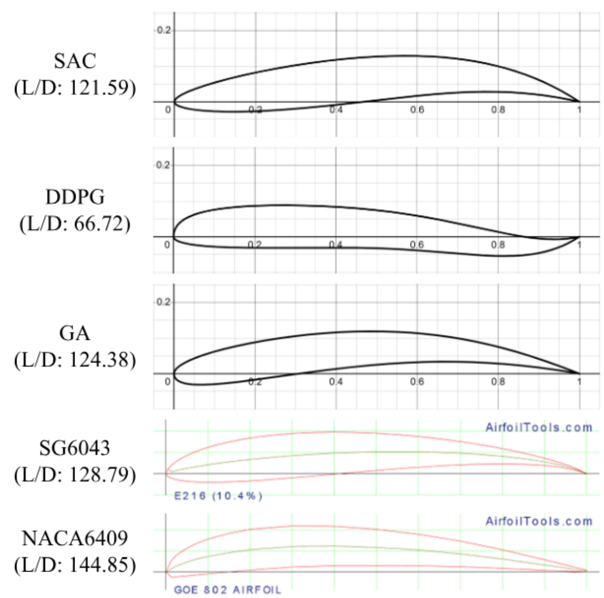

Fig. 4: Comparison between airfoil geometries and corresponding L/D ratios produced by optimization methods and industry standard designs.

Despite these constraints, the algorithms demonstrated the ability to substantially improve aerodynamic efficiency from the baseline symmetric airfoil. SAC and GA, in particular, reached L/D values approaching industry-standard performance, indicating that even with low-dimensional control and limited simulation resources, both RL and evolutionary strategies can produce high-quality candidate airfoil designs suitable for conceptual and early-stage aerodynamic optimization. The results highlight the importance of algorithmic exploration-exploitation balance, parameterization flexibility, and computational budget in determining achievable performance.

B. Practical Implications

The results of this study demonstrate that RL and GAs can substantially improve airfoil aerodynamic efficiency even under tight computational constraints and low-dimensional parameterizations. Starting from a symmetric Bézier airfoil with an L/D of 24.24, SAC achieved 121.59, DDPG reached 66.72, and GA achieved 124.38, approaching the performance of industry-standard airfoils such as the SG6043 ($L/D = 128.79$) and NACA 6409 ($L/D = 144.85$) at the same angles of attack and Reynolds number.

These findings suggest that machine learning approaches, particularly SAC, are capable of producing high-quality candidate designs with limited simulation budgets. The stochastic exploration in SAC allowed consistent identification of high-performing airfoils, while GA leveraged population diversity to discover competitive solutions. Although DDPG was less effective in this low-sample regime due to its deterministic policy, it remains a promising method for fine-tuning designs when near optimal regions.

For aerospace engineering practice, these results indicate that both RL and evolutionary algorithms can serve as rapid prototyping tools in preliminary airfoil design, providing near-industry-standard performance without the need for high-fidelity CFD evaluation at every iteration. This could accelerate conceptual design loops and reduce early-stage development costs.

C. Limitations

Several limitations of the present study should be acknowledged, primarily driven by the need to balance experimental scope with computational feasibility. The most significant constraint arises from the airfoil parameterization used to define the design space. Only eight Bézier control points were employed, and their x-coordinates were held fixed and uniformly distributed along the chord. Furthermore, control point modifications were restricted solely to vertical (y-direction) movement, and each coordinate was bounded within a range of ± 0.15 relative to the initial symmetric state. While this configuration enables controlled and stable learning, it substantially limits the complexity and diversity of airfoil geometries that can be generated. Important geometric features, such as leading-edge radius variation, chordwise camber shifts, thickness redistribution, or trailing-edge shaping, cannot be fully explored under these constraints. Consequently, the optimization is confined to relatively smooth camber and thickness perturbations rather than representing the full space of practical aerodynamic designs.

The dimensionality reduction imposed by this control scheme also restricts the potential performance gains achievable by all optimization methods tested. Though beneficial for sample-efficient learning under scarce simulation budgets, it inherently biases results toward algorithm behavior within a narrow design subspace rather than capturing their capabilities under higher-dimensional, more realistic optimization problems. Extending the control space to include additional points or allowing limited chord-wise positioning would increase representational power at the cost of higher exploration complexity and simulation expense.

A second major limitation arises from the computational budget constraints. Each optimization step required a full invocation of the XFOIL solver, significantly limiting the total number of allowable environment interactions. All RL agents and GA runs were therefore restricted to 4,096 total evaluations. While this ensured fair comparison across methods, it prevented the algorithms from reaching full convergence and limited assessment primarily to early-stage learning performance and sample efficiency. Algorithms that typically benefit from longer training horizons may have been disproportionately constrained by this restriction.

Additionally, the aerodynamic evaluations relied exclusively on XFOIL, which employs a two-dimensional panel method with boundary-layer corrections. While XFOIL provides fast and widely accepted preliminary aerodynamic predictions, it does not fully capture complex flow phenomena such as three-dimensional effects, flow separation, stall behavior, or transitional boundary-layer dynamics in high-angle-of-attack regimes. Therefore, the optimized airfoils should be viewed as candidates for conceptual design rather than validated high-fidelity aerodynamic solutions.

The experiments were also conducted under fixed flow conditions, utilizing a single Reynolds number and angle-of-attack setting. This simplifies the optimization process but limits the generality of the resulting airfoils. Real-world aerodynamic designs typically require robust performance across a range of operating conditions, which was not investigated within the scope of this study.

Finally, the reward formulation was restricted to a single scalar objective, lift-to-drag ratio (L/D). While this metric captures overall aerodynamic efficiency, it neglects other important design considerations such as structural feasibility, stall margin, pitching moment, and manufacturability constraints. Incorporating multi-objective optimization would offer a more realistic and comprehensive assessment of the design space.

Despite these limitations, the experimental setup provides a valuable and controlled platform for evaluating algorithm behavior under realistic computational constraints. The results therefore offer meaningful insight into the comparative sample efficiency, stability, and exploration performance of RL and evolutionary strategies applied to simulation-expensive aerodynamic optimization tasks.

D. Future Work

Several avenues exist to extend and enhance the present study, addressing both methodological limitations and opportunities for higher-fidelity optimization. Regarding the airfoil representation, future work could explore increasing the number of Bézier control points to allow finer resolution of geometric modifications, as well as relaxing the restriction of fixed x-coordinates to enable chordwise movement and more complex shape variations. Alternative spacing strategies for control points could also improve the expressiveness of the parameterization. The action and observation constraints could be systematically varied to study their effects on exploration efficiency and solution quality, including loosening or tightening the upper and lower bounds of control-point displacements and experimenting with different normalization strategies. In terms of RL hyperparameters, further investigation could examine the impact of batch size, the number of learning steps per update, and the learning start threshold on sample efficiency and convergence speed. Automated hyperparameter optimization methods, such as Bayesian optimization or population-based training, could help identify configurations that maximize performance under computational constraints.

Simulation parameters also present opportunities for extension. Expanding the range of operating conditions, including different angles of attack, Reynolds numbers, or iteration counts, could improve the robustness of optimized airfoils across realistic flight envelopes. Alternative optimization strategies could be explored as well; Proximal Policy Optimization (PPO) could provide additional stability and robustness in stochastic settings, while Twin Delayed Deep Deterministic Policy Gradient (TD3) may improve performance over DDPG by mitigating critic overestimation and handling continuous control more effectively. Finally, the geometric parameterization itself could be extended beyond Bézier curves. Using standard NACA 4-digit airfoils, Class-Shape Transformation (CST) methods, or Parametric Equation Concept (PARSEC) formulations may provide additional degrees of freedom, enforce inherent geometric constraints, and facilitate direct comparisons with industry-standard airfoils. Pursuing these directions would allow for more expressive designs, higher sample efficiency, and optimization results that are more generalizable and applicable to real-world aerospace applications.

VII. Conclusion

This study demonstrated the potential of reinforcement learning and evolutionary algorithms for aerodynamic airfoil optimization under computationally constrained conditions. Starting from a symmetric Bézier airfoil, both SAC and GA significantly improved lift-to-drag ratios, approaching the performance of industry-standard airfoils such as the SG6043 and NACA 6409. SAC's stochastic policy enabled efficient exploration, while GA leveraged population diversity to identify high-performing designs. DDPG, while less effective in this low-sample regime, highlighted the tradeoff between deterministic exploitation and exploration. Despite limitations in control-point flexibility, simulation fidelity, and single-condition optimization, the results indicate that machine learning and evolutionary strategies can serve as practical tools for rapid, sample-efficient airfoil design. Future work extending control-point degrees of freedom, exploring alternative parameterizations, and incorporating multi-objective or multi-condition optimization could further improve design quality and generalization, bridging the gap between conceptual design and industry-grade performance.

References

[1] O. J. P. F. D. N. M. Thomas P. Dussauge, Woong Je Sung, "A reinforcement learning approach to airfoil shape optimization," *Scientific Reports*, 2023.

[2] A. N. Amanda Lampton and J. Valasek, "Reinforcement learning of morphing airfoils with aerodynamic and structural effects," *Aerospace Computing, Information, and Communication*, 2012.

[3] A. N. Dev Rajnarayan and J. Mehr, "Universal airfoil parametrization using b-splines," *Faculty Publications*, 2018.

[4] H. Y. Bo Lian and J. Wang, "Performance analysis of three heuristic algorithms for airfoil design optimization," *International Journal of Green Energy*, 2022.

[5] J. W. Haris Moazam Sheikh, Sangjoon Lee and P. S. Marcus, "Airfoil optimization using design-by-morphing," *Journal of Computational Design and Engineering*, 2023.

[6] G. P. P. Scavella and C. Greco, "Deep reinforcement learning-based airfoil design and optimization: An aerodynamic analysis," *Aerospace Science and Technology*, 2025.

[1] [2] [3] [4] [5] [6]