OXFORD CDE

Journal of Computational Design and Engineering, 2026, **13**, 108–124

DOI: 10.1093/jcde/qwaf124
Advance access publication date: 12 November 2025
Research Article

Airfoil optimization using Design-by-Morphing with minimized design-space dimensionality

Sangjoon Lee [1] and Haris Moazam Sheikh [2,*]

[1]Center for Turbulence Research, Stanford University, Stanford, CA 94305, USA
[2]Department of Aeronautics and Astronautics, University of Southampton, Southampton SO17 1BJ, UK
*Correspondence: h.m.sheikh@soton.ac.uk

Abstract

Effective airfoil geometry optimization requires exploring a diverse range of designs using as few design variables as possible. This study introduces AirDbM, a Design-by-Morphing (DbM) approach specialized for airfoil optimization that systematically reduces design-space dimensionality. AirDbM selects an optimal set of 12 baseline airfoils from the UIUC airfoil database, which contains over 1600 shapes, by sequentially adding the baseline that most increases the design capacity. With these baselines, AirDbM reconstructs 99% of the database with a mean absolute error below 0.005, which matches the performance of a previous DbM approach that used more baselines. In multi-objective aerodynamic optimization, AirDbM demonstrates rapid convergence and achieves a Pareto front with a greater hypervolume than that of the previous larger-baseline study, where new Pareto-optimal solutions are discovered with enhanced lift-to-drag ratios at moderate stall tolerances. Furthermore, AirDbM demonstrates outstanding adaptability for reinforcement learning (RL) agents in generating airfoil geometry when compared to conventional airfoil parameterization methods, implying the broader potential of DbM in machine learning-driven design.

Keywords: Design-by-Morphing, airfoil, design-space dimensionality, optimization, reinforcement learning

Nomenclature

Alphabets and greek letters

- $\mathcal{A}$: Arbitrary airfoil shape
- $\mathcal{B}_i$: i-th baseline airfoil shape
- c: Airfoil chord length (m)
- c_s: Speed of sound (m s^{-1})
- d: Drag force exerted on an airfoil per unit span (N m^{-1})
- l: Lift force exerted on an airfoil per unit span (N m^{-1})
- $\mathcal{M}$: Morphed airfoil shape
- N: Design-by-Morphing shape normalization factor
- S: Airfoil shape similarity measure
- U: Freestream flow speed (m s^{-1})
- w_i: Design-by-Morphing weight factor with respect to the ith baseline
- x: Horizontal Cartesian coordinate
- y: Vertical Cartesian coordinate
- α: Airfoil angle of attack (°)
- α_s: Airfoil stall angle, the first local maximum of α with respect to l (°)
- $\Delta\alpha$: Stall tolerance, the range of α between α_s and the maximum l/d point (°)
- ν: Fluid kinematic viscosity (m^2 s^{-1})
- ρ: Fluid density (kg m^{-3})

Dimensionless groups

- C_d: Drag coefficient, $2d/(\rho U^2 c)$
- C_l: Lift coefficient, $2l/(\rho U^2 c)$
- Re: Reynolds number based on airfoil chord length, Uc/ν
- Ma: Mach number, U/c_s

1. Introduction

In aerodynamic design, airfoil shape optimization remains a fundamental and active challenge that requires the exploration of a design space comprising diverse and valid airfoil configurations (Drela, 1998; Lyu *et al.*, 2015; Skinner & Zare-Behtash, 2018; Martins, 2022). Evaluating the performance of each design involves analyzing multiple dynamic metrics (e.g. lift, drag, and stall angle) across varying flight conditions (e.g. wind speed and angle of attack), often demanding computationally intensive simulations to identify optimal candidates. As in typical optimization processes, the first step is to define a design space that captures a broad range of airfoil shapes using a finite set of parameters, enabling systematic exploration and refinement toward optimal solutions (Sobester & Barrett, 2008; Masters *et al.*, 2017).

A number of studies have explored airfoil shape parameterization methods, including—but not limited to—PARSEC (Sobieczky, 1999), Class-Shape Transformation (CST; Kulfan & Bussoletti, 2006), Hicks-Henne bump functions (Hicks & Henne, 1978), Bézier curves (Gordon & Riesenfeld, 1974; Derksen & Rogalsky, 2010), or Non-Uniform Rational B-Splines (NURBS) (Piegl & Tiller, 1996; Lepine *et al.*, 2000), which form the basis of Free-Form Deformation (FFD; Sederberg & Parry, 1986). Recently, with advances in machine learning, deep generative models have also been explored for airfoil shape parameterization via nonlinear dimensionality reduction (Chen *et al.*, 2020; Xie *et al.*, 2024). Each method provides a systematic approach to achieving design flexibility based on distinct mathematical principles. The choice of parameterization significantly affects the diversity of airfoil shapes within the constructed design space. If the global design space of airfoil shapes (repre-

Received: July 3, 2025. **Revised:** October 21, 2025. **Accepted:** October 21, 2025
© The Author(s) 2025. Published by Oxford University Press on behalf of the Society for Computational Design and Engineering. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

senting the maximum possible diversity) were known, the optimal method would be the one that constructs this space with the fewest parameters, thereby mitigating the *curse of dimensionality* (Sobester & Barrett, 2008; Viswanath *et al.*, 2011; Serani & Diez, 2024).

In general, the maximum level of design diversity is unattainable. Many aerodynamic and hydrodynamic shape design problems—such as high-speed train aerodynamics (Oh *et al.*, 2018), riblet surface design for drag reduction (Bai *et al.*, 2016; Lee *et al.*, 2024), or hydrokinetic turbine draft tube optimization (Sheikh *et al.*, 2022)—even suffer from a lack of design diversity for several reasons. First of all, these problems inherently involve highly nonlinear dynamics in design evaluation, making the correlation between geometry and performance non-intuitive and difficult to predict. Additionally, many of the practical or commercially adopted designs, often developed through costly trial-and-error processes (because of the first reason), remain proprietary and are not publicly available (Benjamin & Iaccarino, 2025). As a result, design exploration in such problems is significantly restricted by a lack of a rich, comprehensive, and centralized design database, leaving room for novel designs that have yet to be explored.

In this respect, Design-by-Morphing (DbM) has been proposed to offer a universal strategy to these challenges across different design problems by enabling extensive design space exploration based on a limited set of baseline designs (Oh *et al.*, 2018; Sheikh *et al.*, 2022, 2023; Lee *et al.*, 2024). Rather than relying on predefined shape parameterizations which are mostly problem-specific (e.g. PARSEC for airfoils), DbM generates intermediate forms by morphing between selected baselines, facilitating the constraint-free and continuous creation of new designs. The weight factors assigned to the baselines replace traditional shape parameters, meaning that the number of baselines determines the dimensionality of the design space. DbM also allows for *extrapolative* morphing, wherein negative weight factors can be applied to some or all baselines while handling non-feasible geometries (e.g. self-intersections), which increases the ability to encompass novel shapes to expand the design space beyond the interpolative morphing of the baseline design set. Another major advantage of this framework comes from its inherent interpretability; the weight factors explicitly quantify the geometric influence of each baseline airfoil, enabling intuitive design adjustments that are directly comprehensible.

Although DbM offers clear theoretical advantages in design space exploration, it is important to rigorously assess its practical effectiveness in comparison to established shape parameterization methods. The airfoil design and optimization problem serves as an ideal benchmark for this purpose, given the availability of several conventional airfoil shape parameterization techniques that provide meaningful standards for comparison. Moreover, unlike many other aerodynamic and hydrodynamic shape design problems, airfoil design also benefits from a rich, publicly available database containing more than 1600 airfoil shapes as of now (Selig, 2024). This extensive repository (hereinafter denoted as the UIUC database), a result of over a century of modern airfoil development (Bilstein, 1989; Anderson, 1997), offers a diverse set of tested and proposed design alternatives. While the UIUC database may not perfectly represent the global airfoil design space, its breadth and diversity make it sufficiently comprehensive to be regarded as approximately global for practical purposes.

From this perspective, DbM for airfoil optimization was evaluated in the authors' previous study (Sheikh *et al.*, 2023). DbM was applied to reconstruct the entire UIUC database, with its performance compared against PARSEC, NURBS, and the Hicks-Henne approach. The results confirmed DbM's competitiveness in airfoil shape generation and demonstrated the importance of extrapolative morphing in expanding the design space. However, in that study, the baseline selection of 25 airfoil shapes relied on designer judgment, raising questions about DbM's sensitivity to baseline selection (though partial mitigation was achieved through subset analysis, variations across distinct baseline sets remained unexamined). Such manual curation risks unintended biases in design-space coverage. Note that this concern arises specifically when a diverse, representative baseline dataset exists (e.g. more than hundreds of design points); in contexts lacking comprehensive design sets (less than 10 design points), all existing designs can simply be used as baselines for DbM (e.g. Sheikh *et al.*, 2022).

Given the presence of comprehensive accessible designs, if DbM is able to achieve the same design generation capacity with their small subset, it can further benefit optimization by reducing the dimensionality of the design space, resulting in more time-efficient optimum search and faster convergence. We address this issue by introducing a systematic baseline identification process that mitigates designer bias while maximizing design-space representativeness. Focusing on a specific case of DbM for airfoil design and optimization, the current study aims to provide the following contributions:

- Developing an effective approach to identify reduced baseline sets for DbM while maintaining its airfoil design generation capacity.
- Presenting an optimal baseline set for DbM with reduced design-space dimensionality, which rivals the precedent with a larger number of airfoil baselines.
- Quantifying improvements in airfoil shape design and optimization using a new DbM with reduced design-space dimensionality.

We first briefly revisit the application of DbM in 2D airfoil design, describing the details of morphing, the similarity measure between airfoil shapes, and the airfoil reconstruction problem with additional clarifications from our previous study. Several approaches for identifying optimal baseline selections (assuming the global design space is represented by the UIUC database) are then discussed, along with an analysis of which approach is the most feasible given limited computing resources. Next, using the baseline set revealed through this approach, example cases of airfoil optimization are conducted with the objectives of maximizing the lift-to-drag ratio and stall angle tolerance, quantifying convergence acceleration and solution enhancements. Finally, we demonstrate DbM's adaptability in reinforcement learning environments for airfoil geometry generation, enabling designers to achieve faster learning rates and higher accuracy than conventional airfoil parameterization methods.

2. Design-by-Morphing for Airfoil Optimization

2.1. Formulation for airfoil morphing

In this section, we review and summarize the 2D airfoil design process using DbM, which was described in Sheikh *et al.* (2023). While the core procedure remains identical, we provide additional details that were not thoroughly covered in the previous study to enhance the robustness of this design technique. For this purpose, we first outline the general steps of DbM before applying them specifically to airfoil morphing.

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

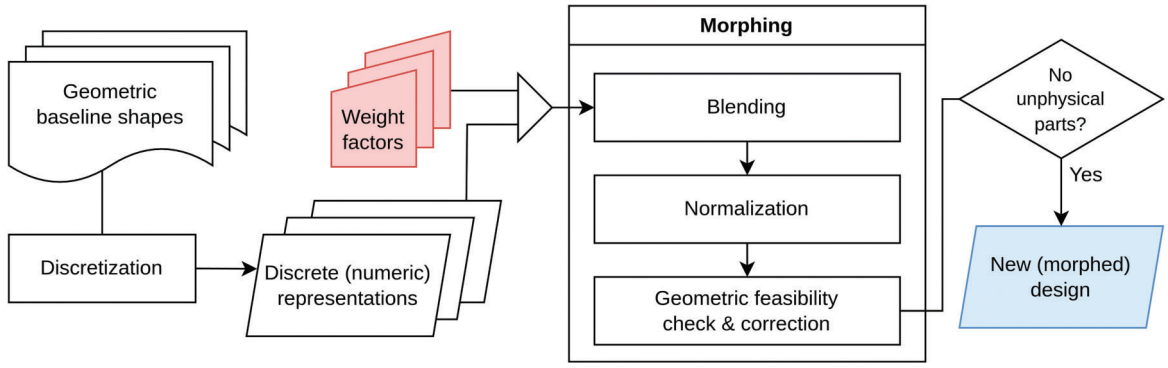


Figure 1: General flowchart of DbM to get a new design by *morphing* baseline shapes.

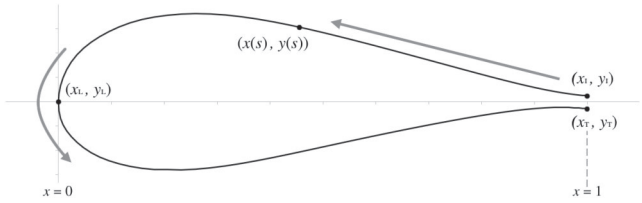


Figure 2: Selig coordinate format for airfoil geometry of a unit chord length (see Selig, 2024).

Figure 1 presents a generalized flowchart of DbM, outlining the sub-processes involved in outputting a newly morphed design from selected baselines, where the user specifies weight factors for each baseline as input. This flowchart assumes that the baseline designs have already been selected; the reduction of design-space dimensionality (i.e. using fewer baselines) is not the focus here but will be addressed later (Section 3).

In the pre-morphing stage, baseline shapes, originally defined in geometric form, must be converted into a discrete numerical representation in a consistent format to enable basic arithmetic operations (e.g. addition and scalar multiplication) for computational processing. This concept is widely studied in computer animation for object transformation, where various techniques have been developed (Parent, 2012). We refer to this process as *discretization* to emphasize the mapping of geometric shapes into a consistent numerical form, such as control points or grids.

Next, the morphing stage consists of three sub-processes: (1) *blending*, where the baseline shapes are combined according to the input weight factors; (2) *normalization*, which scales the blended shape to fit within typical scale of the problem; and (3) *geometric feasibility check and correction*, which adjusts and removes any unphysical parts, mostly represented by self-intersections. If no unphysical parts remain in the final shape, the process is complete, and a new design is achieved. Notably, the overall process resembles the metamorphosis of irregularly shaped (e.g. non-rectangular) objects in computer graphics and, in 2D, several practical approaches to these general procedures have been considered and developed (Sederberg & Greenwood, 1992; van den Bergh *et al.*, 2002).

When it comes to airfoil morphing, one of the most widely used formats for describing airfoil geometry is the Selig coordinate format, or simply the Selig format. Named after Selig, this format is used to store airfoil data in a structured manner. As shown in Figure 2, it consists of a list of (x, y) coordinate pairs that define the airfoil geometry non-dimensionalized by chord length c. The coordinates are arranged sequentially, starting from the upper trailing edge ($x = x_I = 1$), following the upper surface toward the leading edge ($x = x_L = 0$), and then continuing along the lower surface back to the lower trailing edge ($x = x_T = 1$). Using this format, any arbitrary airfoil shape $\mathcal{A}$ can be represented as a parametric curve with respect to a variable s, defined as

$$\mathcal{A}(s) \equiv \big(x(s),\ y(s)\big) \quad 0 \le s \le 2, \tag{1}$$

where $x(s) \equiv |1 - s|$ and $y(s)$ depends on the specific airfoil geometry (which thereby defines it). Discretizing s into equispaced points s_j for $j = 0,\ 1,\ \cdots,\ F$, such that $0 = s_0 < s_1 < \cdots < s_F = 2$ with $s_j = 2j/F$, we obtain $\mathcal{A}$'s discrete numerical representation as the following $(F+1)$-dimensional vector (referred to as the Selig-format vector henceforth):

$$\vec{\mathcal{A}} \equiv \Big[\ y(s_0)\ y(s_1)\ \cdots\ y(s_F)\ \Big]^T \in \mathbb{R}^{F+1}. \tag{2}$$

We assume that F is sufficiently large so that the airfoil's shape is well preserved, with minimal loss of geometric detail between consecutive discretized points. In practice, $F = 200$ is found to be large enough to represent every airfoil in the UIUC database. In Equation 2, $y(s_0) = y(0) = y_I$, $y(s_{F/2}) = y(1) = y_L$, and $y(s_F) = y(2) = y_T$ (in order for $F/2$ to be integer, let's assume F to be even).

Given n baseline airfoil shapes $\mathcal{B}_1, \mathcal{B}_2, \cdots \mathcal{B}_n$, each can be expressed in Selig-format vector form as $\vec{\mathcal{B}}_1, \vec{\mathcal{B}}_2, \cdots \vec{\mathcal{B}}_n$ in $\mathbb{R}^{F+1}$. Since these baseline shapes correspond one-to-one and $\mathbb{R}^{F+1}$ is a vector space that is equipped with well-defined addition and scalar multiplication, we can formally define the process of airfoil morphing. Topologically, the existence of a one-to-one mapping is ensured by the homeomorphism of these shapes, which is a prerequisite for performing DbM, as noted in Sheikh *et al.* (2023). Our previous study highlighted the homeomorphism of 2D closed shapes. However, in consideration of the fact that the geometric representation an airfoil here is a closed curve as in Equation 1, the relevant homeomorphism to be correctly highlighted should be that of a 1-manifold with boundary, topologically equivalent to closed intervals. For mathematical rigor, in the case of airfoils with zero trailing edge thickness ($y(0) - y(2) = 0$), let zero be interpreted as *almost zero* ($y(0) - y(2) = 0^+$) to preserve the same homeomorphism (by conceptually separating the endpoints).

Overall, morphing of the n baseline airfoil shapes with given weight factors $w_1, w_2, \cdots, w_n \in [-1, 1]$ is expressed as follows:

$$\vec{\mathcal{M}} = \mathcal{F}\left(\frac{1}{N}\sum_{i=1}^{n} w_i \vec{\mathcal{B}}_i\right), \tag{3}$$

where $N = N(w_1, w_2, \cdots, w_n)$ is a normalization factor that scales the blended shape, and $\mathcal{F}$ represents a set of adjustment opera-

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

tions to correct unphysical geometries. A linear blending formula is chosen as it represents the simplest form of blending. However, the choice of blending is not necessarily limited to linear methods, and users may also explore nonlinear blending approaches. Similarly, the normalization factor can be determined in various ways, but we adhere to the original formulation

$$N = \sum_{i=1}^{n} w_i \tag{4}$$

to maintain consistency with the original study for comparative purposes. As for checking and correcting geometric feasibility, the removal of self-intersections is essential. Additional treatments may be applied depending on what design features users consider unphysical (e.g. holes). In the present airfoil morphing process, we focus solely on treating self-intersections. If no self-intersections are present, $\mathcal{F}$ is the identity map. Otherwise, in general, we adopt the self-intersection removal procedure for airfoil shapes as introduced in section 2.2 of Sheikh *et al.* (2023), in which zero-thickness points due to self-intersections are locally stiffened and then smoothed. Starting with the Selig-format vector representation, we can inexpensively detect self-intersections in an airfoil shape using a simple sign-change checker. The detection algorithm is presented in Algorithm 1.

Algorithm 1 Detect a self-intersecting airfoil shape given as a Selig-format vector

Require: $\vec{\mathcal{A}} = [\, y(s_0)\ \ y(s_1)\ \ \cdots\ \ y(s_F)\,]^T$ where F is even (e.g., $F = 200$)
Ensure: True if $\vec{\mathcal{A}}$ represents a self-intersecting shape, else False
  $\vec{\mathcal{A}}_u \Leftarrow [\, y(s_{F/2-1})\ \ \cdots\ \ y(s_0)\,]^T$ // slice the first half of $\vec{\mathcal{A}}$, and flip it
  $\vec{\mathcal{A}}_l \Leftarrow [\, y(s_{F/2+1})\ \ \cdots\ \ y(s_F)\,]^T$ // slice the second half of $\vec{\mathcal{A}}$
  Allocate $\vec{b}$ (integer array) of size $F/2$
  for i $\Leftarrow$ 1 to F/2 **do** // one-based indexing is assumed
    x $\Leftarrow$ $\vec{\mathcal{A}}_u$[i] - $\vec{\mathcal{A}}_l$[i]
    $\vec{b}$[i] $\Leftarrow$ Sign of x (1 for positive values, -1 for negative values, and 0 for zero)
  end for
  for j $\Leftarrow$ 1 to F/2 − 1 **do**
    if ($\vec{b}$[j] * $\vec{b}$[j+1] < 0) .OR. ($\vec{b}$[j] == 0) **then**
      return True
    end if
  end for
  return False

2.2. Shape similarity measure

For two arbitrary airfoil shape geometries $\mathcal{A}_1$ and $\mathcal{A}_2$ in the parametric curve form with respect to s as in Equation 1, their similarity, denoted as $S(\mathcal{A}_1, \mathcal{A}_2)$, can be quantified by measuring the mean absolute error (MAE) between these two airfoils along the upper and lower surfaces, respectively, and then summing the results. That is,

$$S(\mathcal{A}_1, \mathcal{A}_2) \equiv \underbrace{\frac{\int_0^1 |y_1(s) - y_2(s)|ds}{\int_0^1 ds}}_{\text{Upper curve}} + \underbrace{\frac{\int_1^2 |y_1(s) - y_2(s)|ds}{\int_1^2 ds}}_{\text{Lower curve}}, \tag{5}$$

where $y_1(s)$ and $y_2(s)$ are the y-coordinates of $\mathcal{A}_1(s)$ and $\mathcal{A}_2(s)$, respectively. Since both $\int_0^1 ds$ and $\int_1^2 ds$ evaluate to unity, Equation 5 simplifies to

$$S(\mathcal{A}_1, \mathcal{A}_2) = \int_0^2 |y_1(s) - y_2(s)|ds. \tag{6}$$

Here, $\int_0^2 (y_1(s) - y_2(s))ds = 0$ is assumed to provide a consistent vertical alignment. This formulation is equivalent to the airfoil shape similarity measure (as MAE) proposed by Sheikh *et al.* (2023). It is important to note that various similarity measures can be defined as long as they form a convergent series in which the similarity value approaches a certain limit (mostly zero) as one shape becomes identical to the other; Equation 6 evidently satisfies this fundamental requirement.

Taking one step further from Sheikh *et al.* (2023), let us derive a discretized formula that is effectively equivalent to Equation 6. Considering an equispaced $(F+1)$-point discretization of s, we use numerical integration based on the trapezoidal rule to obtain an approximate form of Equation 6, given by

$$\begin{aligned} S(\mathcal{A}_1, \mathcal{A}_2) &\simeq \frac{2}{F}\sum_{i=1}^{F}\frac{1}{2}\left(|y_1(s_{i-1}) - y_2(s_{i-1})| + |y_1(s_i) - y_2(s_i)|\right) \\ &= \sum_{i=1}^{F-1}\frac{2}{F}|y_1(s_i) - y_2(s_i)| \\ &\quad + \frac{1}{F}\left(|y_1(s_0) - y_2(s_0)| + |y_1(s_F) - y_2(s_F)|\right). \end{aligned} \tag{7}$$

For a more compact expression, we may factor the endpoint terms into the summation by multiplying them by 2, resulting in

$$S'(\mathcal{A}_1, \mathcal{A}_2) = \frac{2}{F}\sum_{i=0}^{F}|y_1(s_i) - y_2(s_i)|. \tag{8}$$

S' can be interpreted as a modified version of S that places slightly more weight on endpoint error evaluation. In the context of airfoils, this adjustment emphasizes matching the trailing edge, which can be rationalized since airfoil dynamics are considerably influenced by edge geometries. The right-hand side of Equation 8 corresponds to the mean absolute difference between the Selig-format vectors of $\mathcal{A}_1$ and $\mathcal{A}_2$ (scaled by $2(F+1)/F$, or approximately 2 when F is much greater than 1). Using the ℓ^1-norm notation, we express the similarity measure for Selig-format vectors as

$$S'(\vec{\mathcal{A}}_1, \vec{\mathcal{A}}_2) = \frac{2}{F}\left\|\vec{\mathcal{A}}_1 - \vec{\mathcal{A}}_2\right\|_1, \tag{9}$$

which we use as the airfoil shape similarity measure that is effectively equivalent to the integral form in Equation 6.

2.3. Airfoil reconstruction problem

Suppose that we aim to reconstruct a known airfoil shape, $\vec{\mathcal{A}}_t$, using the DbM process with given n baseline shapes $\vec{\mathcal{B}}_1, \vec{\mathcal{B}}_2, \cdots \vec{\mathcal{B}}_n$. It is additionally assumed that $\vec{\mathcal{A}}_t$ is distinct from each $\vec{\mathcal{B}}_i$ for any $i = 1, 2, \cdots, n$; otherwise, the reconstruction is trivial. Recalling Figure 1, DbM employs n input morphing weight factors, $w_1, w_2, \cdots, w_n$, to output a morphed airfoil shape $\vec{\mathcal{M}}$ as shown in Equation 3. Defining the weight vector $\vec{w} \equiv [w_1\ w_2\ \cdots\ w_n]^T \in \mathbb{R}^n$, we can formulate the problem of finding $\vec{w}$ as a single-objective optimization problem with continuous variables in standard form:

$$\underset{\vec{w}\,\in\,\mathbb{R}^n}{\arg\min}\quad S'(\vec{\mathcal{M}}(\vec{w}),\ \vec{\mathcal{A}}_t)\quad \text{subject to}\quad \|\vec{w}\|_\infty \le 1, \tag{10}$$

where $\|\cdot\|_\infty$ represents the ℓ^∞-norm.

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

If there exists a set of m airfoil shapes to be reconstructed, denoted as $\vec{\mathcal{A}}_{t,i}$ for $i = 1, \cdots, m$, solving the optimization problem in Equation 10 m times for each $\vec{\mathcal{A}}_{t,i}$ yields m weight vectors $\vec{w}_{opt,i}$. These weight vectors generate morphed shapes that best approximate their respective target airfoil shapes. In each case, the proximity of $S'(\vec{\mathcal{M}}(\vec{w}_{opt,i}), \vec{\mathcal{A}}_{t,i})$ to zero indicates how accurately the morphing of the n baseline shapes reconstructs $\vec{\mathcal{A}}_{t,i}$. Consequently, the sum of these similarity measures, denoted as $S^{\ddagger}$, i.e.

$$S^{\ddagger} \equiv \sum_{i=1}^{m} S'(\vec{\mathcal{M}}(\vec{w}_{opt,i}), \vec{\mathcal{A}}_{t,i}), \tag{11}$$

can serve as an indicator of the reconstruction capability of the set of airfoil shape baselines, $\vec{\mathcal{B}}_1, \vec{\mathcal{B}}_2, \cdots \vec{\mathcal{B}}_n$, for the target airfoil set, $\vec{\mathcal{A}}_{t,1}, \vec{\mathcal{A}}_{t,2}, \cdots \vec{\mathcal{A}}_{t,m}$.

3. Baseline Shape Selection

The central challenge addressed in this section—and a pivotal question for this study—is: How can we identify an optimal minimal set of baseline airfoil shapes that effectively represents the diversity of a larger collection considered globally representative? Selecting such a compact baseline set is conceptually analogous to principal component analysis (PCA), where lower-dimensional subspaces capture significant data variations (see Li *et al.*, 2022, exhibiting the use of PCA to the UIUC database). In our context, the selected baseline airfoils serve a role similar to principal components, enabling reconstruction of diverse airfoil designs through weighted morphing.

However, DbM's baseline selection process is distinct from PCA. While PCA generates abstract principal components through linear combinations, DbM preserves original baseline airfoils as interpretable building blocks. This approach maintains physical intuition by allowing designers to work with recognizable geometries rather than abstract eigencomponents whose meaning is obscured by PCA's rotational transformations. Furthermore, DbM introduces essential non-linearities through geometric feasibility corrections. Although linear blending is employed in this study (as in Equation 3), the method permits extension to non-linear blending strategies, necessitating non-linear dimensionality reduction approaches unlike PCA. This structural flexibility is to enhance DbM's design generation capabilities while preserving geometric interpretability.

With these distinctions established, our objective is to identify a minimal subset from the global airfoil set $\{\mathcal{A}_1, \cdots, \mathcal{A}_m\}$ that minimizes the total reconstruction error $S^{\ddagger}$ Equation 11 when used as DbM baselines. This maximizes the representational power of a compact design space while maintaining DbM's core advantages of physical interpretability and constrained dimensionality.

3.1. Description of problem

To improve the practicality of DbM for airfoil design and optimization, we identify a set of baseline shapes that can effectively span the diversity of possible airfoil geometries. In order to define and quantify the coverage of the airfoil design space, it is necessary to first establish a comprehensive target set that reasonably represents the global airfoil shape design space.

The UIUC airfoil database (Selig, 2024) provides a broad and diverse repository of airfoil geometries, accumulated through more than a century of aerodynamic development. Given its scope, historical depth, and inclusion of a multitude of tested, proposed and optimized designs, the UIUC database can be regarded as approximately global for practical purposes (while no finite database can perfectly capture the infinite possibilities of airfoil geometries, we affirm that the database serves as a sufficiently comprehensive surrogate).

With the UIUC database taken as the global target set of airfoil shapes to be reconstructed by the DbM framework, we frame the identification of optimal baseline shapes as the problem of finding a subset of airfoil shapes whose morphing combinations can best approximate the entire UIUC database, thereby maximizing design-space coverage while minimizing design-space dimensionality. Mathematically, we let the set of available airfoil shapes in the UIUC database, after consistently discretizing them in the Selig-format vector form, be denoted by $\mathbb{A} \equiv \{\vec{\mathcal{A}}_{DB,1}, \vec{\mathcal{A}}_{DB,2}, \cdots, \vec{\mathcal{A}}_{DB,m}\}$, where m (or $\#\mathbb{A}$) is 1644 as of the present collection. We seek to select a subset of n baselines, denoted as $\mathbb{B} \equiv \{\vec{\mathcal{B}}_1, \vec{\mathcal{B}}_2, \cdots, \vec{\mathcal{B}}_n\} \subset \mathbb{A}$, such that the reconstruction capability measure $S^{\ddagger}$ of $\mathbb{B}$ over $\mathbb{A}$ is minimized.

The corresponding optimization problem can be generally expressed as

$$\min_{\mathbb{B} \subseteq \mathbb{A}} \quad S^{\ddagger} \quad \text{subject to} \quad \#\mathbb{B} = n, \tag{12}$$

where n is the number of baseline airfoil shapes allowed in $\mathbb{B}$. When n equals to m, the problem takes a trivial and global solution, $\mathbb{B} = \mathbb{A}$ (which evidently yields $S^{\ddagger} = 0$). Preferentially, $n \ll m$ to promote significant dimensionality reduction.

In Equation 12, n (or $\#\mathbb{B}$) acts as a control parameter balancing the complexity and expressiveness of the design space. A larger n increases the representational power but also the design-space dimensionality, whereas a smaller n reduces the dimensionality at the cost of the airfoil reconstruction capacity. Therefore, solving Equation 12 additionally aims to identify the smallest possible n (or equivalently, the most compact set of baselines) that still achieves an acceptable level of reconstruction performance over the database. However, determining the acceptability is mostly done *a posteriori*; thus, in the subsequent discussion, we presume that n is given, for example, $n = 10$.

3.2. Approaches for the subset selection

Using the concept of feature selection (Guyon & Elisseeff, 2003) or, similarly, factor screening (Serani & Diez, 2024), we identify the most influential elements among a large set of airfoil geometries. In other words, each individual airfoil shape in $\mathbb{A}$ is treated as a distinct feature that contributes to the construction of the overall design space. Since not all airfoil shapes are essentially unique—some may offer redundant contributions to the representational capability—an effective selection process should aim to retain only the most informative baselines in $\mathbb{B}$ while safely eliminating superfluous ones.

3.2.1. *Exhaustive search*

The most straightforward approach to baseline subset selection is to exhaustively compare all subsets. All possible combinations of n baselines are enumerated from $\mathbb{A}$, and each candidate subset is evaluated based on its reconstruction capability measure $S^{\ddagger}$. The subset that minimizes $S^{\ddagger}$ is chosen as the optimal set.

For the UIUC database containing $m = 1,644$ airfoil shapes, the number of possible subsets of $\mathbb{A}$ for even modest values of n becomes astronomical. For instance, selecting $n = 10$ baselines would require evaluating approximately $\binom{1644}{10} \approx 3.9 \times 10^{25}$ candidate subsets. As each candidate subset's evaluation towards $S^{\ddagger}$ even necessitates solving Equation 10 $m = 1,644$ times, such a number is computationally infeasible to process.

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

While exhaustive search guarantees identification of the globally optimal baseline sets for a given n, the combinatorial explosion in the number of subsets renders this approach unconditionally impractical for any realistic subset size. Thus, alternative strategies that significantly reduce the computational burden must be sought.

3.2.2. Backward search

One alternative is the backward search strategy. This method starts with the entire database as the initial baseline set, i.e. $\mathbb{B} = \mathbb{A}$. At each iteration, a single baseline is eliminated from $\mathbb{B}$ based on its relative contribution to the overall reconstruction capability, thereby reducing the size of the baseline set by one sequentially.

At the first iteration, all m subsets of size $(m-1)$ are considered, where solving Equation 12 requires only a single subordinate optimization in Equation 10 for the eliminated airfoil of each subset. For each candidate subset, we evaluate $S^{\ddagger}$, and the subset that yields the smallest increase in $S^{\ddagger}$ from its initial value (zero, when $\mathbb{B} = \mathbb{A}$) is chosen. Then, in subsequent iterations, the elimination decision is guided differently to reduce computational cost. After solving Equation 10 for each target airfoil based on the current baseline set, we compute the non-trivial morphing weight factors assigned to each baseline. The baseline whose total contribution, summed across all reconstructions, is then eliminated. In other words, we sequentially remove the baseline that contributes least to reconstructing the target airfoil set according to the absolute sum of its weight factors. This process is repeated iteratively until the number of remaining baselines reaches the desired subset size n.

The total number of times the subordinate optimization for airfoil reconstruction Equation 10, must be conducted throughout the process is given by

$$m + \sum_{k=2}^{m-n+1} k = \frac{1}{2}\left\{m^2 + m(5-2n) + (n-3)n\right\},$$

when counting only non-trivial cases (where a target airfoil is not included in the current baseline set). For $m = 1,644$ and $n = 10$, this amounts to approximately 1.3×10^6 optimal AirDbM weight factor searches. While this backward search strategy is inherently a greedy algorithm for solving Equation 12 (i.e. making local optimal choices at each elimination), it offers a computationally tractable compromise between reconstruction accuracy and cost.

Despite substantially reducing the number of evaluations compared to exhaustive search, this strategy still incurs a computational cost that nearly scales $\mathcal{O}(m^2)$. As m increases, the total number of subordinate optimizations becomes quadratically prohibitive. In the present case with $m = 1,644$, the total number of evaluations remains an impractical computational burden.

3.2.3. Forward search

Another approach is the forward search strategy, which we ultimately adopt in this study. Unlike backward search, forward search progressively builds the baseline set by sequentially adding airfoil shapes from the full database. The process starts with an empty set and, at each iteration, adds a single baseline that is expected to contribute most to improving the overall reconstruction of $\mathbb{A}$.

At the first iteration, all m subsets of size 1, or equivalently, all m individual airfoils are considered. For each, solving Equation 12 simply requires summing the $(m-1)$ airfoil shape similarity measures with respect to all other $(m-1)$ airfoils, without any optimization, as no morphing needs to occur. The airfoil shape that yields the smallest $S^{\ddagger}$ is selected as the first baseline element. Then, in subsequent iterations, an airfoil shape that is least well reconstructed via the current baseline set (i.e. the one with the largest S' value obtained from Equation 10) is added, until the desired subset size n is reached.

Now, the total number of times Equation 10 must be non-trivially solved is given by

$$\sum_{k=2}^{n-1}(m-k) = \frac{1}{2}(n-2)(2m-n-1),$$

as the first iteration with a single baseline shape does not involve DbM weight optimization. For $m = 1,644$ and $n = 10$, this results in 13108 evaluations, which finally becomes a computationally tractable number. Moreover, each reconstruction is performed with a low-dimensional input space because $k < n \ll m$, making the overall process significantly faster than the backward search.

In addition to its computational tractability—since the number of evaluations scales linearly with the total database size m, i.e. $\mathcal{O}(m)$, given $m \gg n$—forward search offers a practical advantage for baseline selection. By sequentially adding baselines while observing the progressive improvement in reconstruction performance, it is possible to monitor how the design-space coverage evolves, allowing for us to flexibly control over how the final subset size n be based on the observed performance trends during the search process, which is in line with the aim described in Section 3.1.

3.3. Selected baselines

To solve the optimization problems involving different baseline sets and target airfoils, as defined in Equation 10, a genetic algorithm (GA) was employed using the **Pymoo** framework (Blank & Deb, 2020) with **Dask**-based parallelization (Rocklin, 2015). Each optimization run used a population size of 100 for up to 500 generations, with crossover and mutation operators set to simulated binary crossover and polynomial mutation, respectively, as provided by default in the framework. Parallel evaluation across multiple Dask workers (32 in this study) significantly accelerated the optimization process. Termination was based on convergence criteria evaluated over a rolling window of 20 generations: variable-space change (i.e. $\|\Delta\vec{w}_{\text{opt}}\|_\infty$ between successive generations less than 10^{-6}) and objective-space change (i.e. $\Delta S'$ less than 10^{-8}). We warm-started each optimization by including the previously obtained optimal weight vector (augmented with a zero morphing weight factor for the newly added baseline) as one of the initial population members when solving for the expanded baseline set.

Throughout the forward search strategy that is powered by GA, the optimal baseline set of size $n = 12$ was identified, which are presented in Table 1. The index indicates the order in which each airfoil was added to the baseline set during the forward search. Accordingly, an optimal baseline set of any smaller size $\eta < 12$ can be constructed by considering only the first η airfoils from this table (that is, baselines #1 – #η). With these 12 baseline shapes, the DbM approach successfully reconstructed all 1644 airfoil shapes in the database with S' (hereafter used interchangeably with MAE) below 0.01. Figure 3 illustrates the comparison between the original and DbM-reconstructed airfoil geometries for 10 airfoils selected at equal rank intervals from the best to worst MAE. The best-performing reconstruction, observed for Eppler E197, exemplifies the inherent redundancy within the database. The geometry is nearly similar to the first baseline shape, Eppler E195, albeit with a slight variation in camber thickness, demonstrating that

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

Table 1: DbM baseline airfoil shape set of size 12.

Index	Geometry	Airfoil Name in the UIUC Database (Selig, 2024)
$\mathcal{B}_1$		Eppler E195
$\mathcal{B}_2$		Wortman FX 79-W-660A
$\mathcal{B}_3$		Gottingen 531
$\mathcal{B}_4$		Eppler 864 Strut
$\mathcal{B}_5$		Roncz R1145MSM VariEze Canard Main
$\mathcal{B}_6$		UIUC Chen
$\mathcal{B}_7$		Griffith 30% Suction
$\mathcal{B}_8$		Selig S9104
$\mathcal{B}_9$		Althaus AH 93-W-480B
$\mathcal{B}_{10}$		Althaus AH 81-K-144 W-F KLAPPE
$\mathcal{B}_{11}$		Eppler E664 (Extended)
$\mathcal{B}_{12}$		Saratov R/C Sailplane

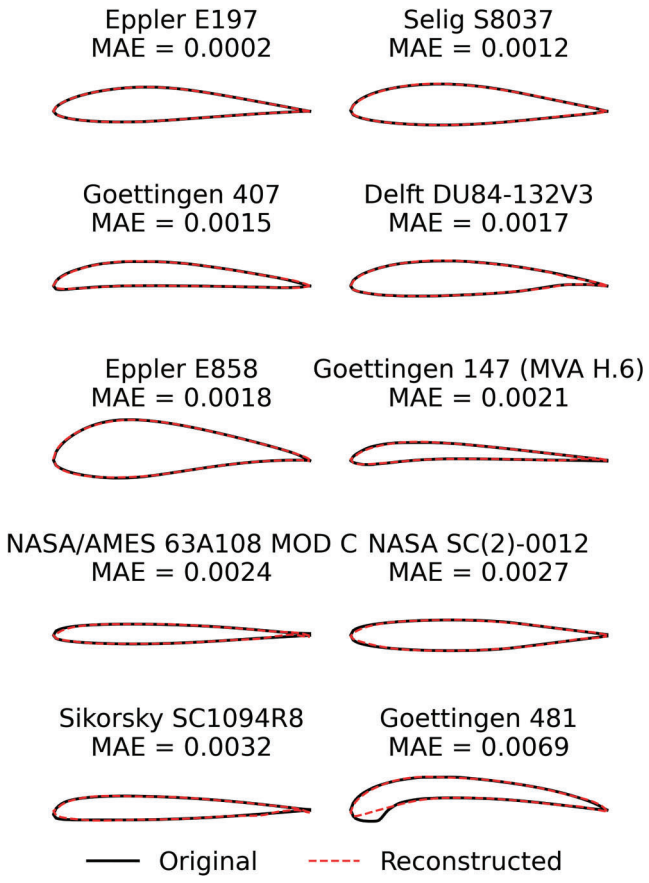


Figure 3: Comparison of original (black solid line) and reconstructed (red dashed line) airfoil geometries via DbM using the selected 12 baselines (see Table 1). The 10 airfoils displayed here are selected at equal rank intervals from best to worst reconstruction based on the MAE similarity metric (see Equation 9).

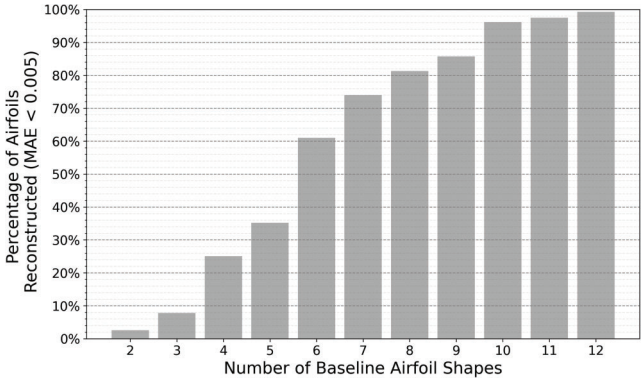


Figure 4: Percentage of airfoils in the database that can be reconstructed via DbM with an MAE below 0.005 for baseline set sizes from 2 to 12 (see Table 1).

not all airfoil shapes represents truly unique design features. This underscores the motivation for the current practice of dimensionality reduction.

On the other hand, in Figure 3, the worst reconstruction case of Gottingen 481 reveals the limited coverage imposed by the reduced baseline set. There exists non-negligible deviation in shape, particularly due to its pronounced curvature at the bottom surface around the leading edge, which suggests that such a geometric feature is not fully taken into account by the current 12 baselines. In compliance with the forward search strategy, we could additionally consider Gottingen 481 as a potential additional baseline shape to better encompass such *strongly curved* airfoil shapes within the design space, while increasing the design-space dimensionality by one. The decision to augment the baseline set with such shapes should require a careful consideration of the balance between design space diversity and the minimal design-space dimensionality. This trade-off can be informed by analyzing the trend of reconstruction convergence as the number of baseline shapes increases.

The effectiveness of the selected 12 baseline shapes is supported by the convergence trend of the reconstruction rate, as visualized in Figure 4. This plot illustrates the percentage of airfoil shapes in the database reconstructed within an MAE tolerance of 0.005 as the number of baseline shapes varies from 2 (i.e. baselines #1– #2) to 12 (i.e. baselines #1–#12). This tolerance threshold is chosen based on its previous use in Sheikh *et al.* (2023) and, as our previous comparison plots demonstrate, it approximately marks the level at which visually notable discrepancies between original and reconstructed shapes become apparent (compare the second-worst case (MAE = 0.0032) to the worst case (MAE = 0.0069) in Figure 3). As the number of baselines increases, the reconstruction rate increases with a flattening of the curve beyond 10 baselines, which indicates diminishing returns for further increases in dimensionality.

Moreover, with the selected set of 12 baselines, more than 98% of airfoil shapes were reconstructed using DbM within an MAE tolerance of 0.005. This level of reconstruction matches the performance reported in our previous DbM study on airfoil optimization (Sheikh *et al.*, 2023), which relied on 25 baseline shapes. Thus, the current selection successfully achieves comparable reconstruction quality while reducing the design-space dimensionality by more than half, from 25 to 12.

Given that the selected 12-baseline set almost entirely spans the UIUC database design space, one might expect that randomly

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

Table 2: Design capacity comparison across airfoil design methods.

Method	Overall MAE (Mean ± Std.)	Percentage of Airfoils Reconstructed (MAE < 0.005)
CST	0.0012 ± 0.0008	99.4%
AirDbM	**0.0021 ± 0.0009**	**99.3%**
NURBS	0.0024 ± 0.0019	91.6%
PARSEC	0.0032 ± 0.0025	84.5%
Hicks-Henne	0.0053 ± 0.0035	58.7%

selecting 12 baselines could maintain reconstruction capacity while bypassing the search cost, analogous to a change of basis in linear vector spaces. However, DbM's morphing process is inherently nonlinear, so replacing shapes would not guarantee preservation of space-spanning capability. To verify this, we tested three randomly selected 12-baseline sets. Their successful reconstruction rates (MAE < 0.005) were significantly lower and inconsistent, achieving only 75%, 66%, and 49%, respectively. These outcomes confirm that a systematic search is essential to achieve both high performance and consistency. In any case, users can bypass this search process entirely and directly utilize the validated baseline set provided in Table 1.

Lastly, it is worth recalling that the current baseline set selection process assumes the UIUC database to be globally representative. However, one can point out its limited coverage of supersonic or hypersonic airfoils. Similar to the Gottingen 481 case, the DbM framework can readily accommodate baseline augmentation when applications require non-inclusive or underrepresented designs, such as diamond-shaped supersonic airfoils (Jernell, 1974), which seamlessly expands the global design space. This inherent adaptability demonstrates the framework's strength in accommodating unforeseen or novel designs.

3.4. Design capacity comparison

While design capacity comparisons of DbM against conventional airfoil parameterization methods constituted the main theme of our previous work (Sheikh *et al.*, 2023), which confirmed DbM's competitiveness with methods specifically designed for airfoils, we provide a brief comparison here again to evaluate the performance of the present reduced 12-baseline DbM (henceforth denoted AirDbM to specify its application to airfoil design and optimization and its reduced baseline set of 12 as defined in Table 1).

To assess AirDbM's design capacity under reduced dimensionality constraints, we compared reconstruction performance using a consistent number of design variables across all methods: 12 design variables for AirDbM, Hicks–Henne bump functions, class-shape transformation (CST), and the parametric section (PARSEC) method, with 13 variables for non-uniform rational B-splines (NURBS) due to its formulation requirements. Implementation details are provided in Appendix A. The comparison was conducted across all 1644 airfoils in the UIUC database using the same MAE evaluation scheme as established in the earlier sections.

Table 2 presents the quantitative comparison results. AirDbM shows competitive performance, achieving reconstruction quality comparable to CST while significantly outperforming NURBS, PARSEC, and Hicks-Henne. This performance is particularly significant given that conventional methods like NURBS and Hicks-Henne experience substantial degradation in design coverage when constrained to lower dimensionalities, compared to the higher-dimensional cases (24–26 design variables) achieving 98% reconstruction success reported in Sheikh *et al.* (2023). The superior performance of CST appears to come from its aerodynamic-specific design principles, employing carefully crafted class and shape functions with mathematical rigor tailored for airfoil applications (see Kulfan & Bussoletti, 2006). AirDbM's comparable performance is therefore noteworthy, considering its universal morphing principle that remains applicable across diverse design domains while maintaining high reconstruction fidelity under significant dimensionality reduction.

To illustrate the practical implications of the design capacity differences, Figure 5 presents reconstruction results for the Wortman FX 79-W-660A airfoil, which represents the worst-reconstructed case (i.e. highest summed MAE across all five methods). This result is primarily due to its unusually thick profile which deviates from typical airfoil geometries. Although conventional methods struggle with this unconventional shape, AirDbM naturally overcomes this challenge by directly incorporating the specific thick profile as its baseline ($\mathcal{B}_2$). The key strength of the DbM framework lies in the ease of incorporating such novel designs contributing to design diversity: regardless of how unconventional a target shape may be, it can be effectively utilized in the design process with the cost of only a single design parameter. Readers are encouraged to refer to a similar discussion in our previous work regarding the reconstruction of a 'mirrored' airfoil with flipped sharp and blunt edges (Sheikh *et al.*, 2023, pp. 1447–1448).

4. Performance Evaluation

4.1. Multi-objective airfoil optimization

We first incorporate AirDbM into an airfoil shape optimization problem, optimizing the airfoil shape based on aerodynamic information obtained from a flow solver. Since the primary objective of this test study is to evaluate the computational efficiency of the proposed DbM approach with reduced design-space dimensionality, the focus is placed more on analyzing optimization performance than on the optimal outcomes themselves. Accordingly, we revisit the airfoil optimization setup from our previous work (Sheikh *et al.*, 2023) to investigate the optimization performance under reduced design-space dimensionality, compared with the previous 25-baseline case. It should be noted that multi-objective optimization for airfoil dynamics has garnered increasing attention in recent years (e.g. Jing *et al.*, 2023; Zhang *et al.*, 2024; Jung & Gu, 2024), further underscoring the practical utility of this work.

An airfoil with chord length c, subjected to a freestream flow of speed U, fluid density ρ, and kinematic viscosity ν (i.e. under the Reynolds number condition $\mathrm{Re} = Uc/\nu$), is characterized by two key dynamic performance parameters: the lift coefficient C_l and the drag coefficient C_d, defined as

$$C_l(\alpha) = \frac{2l(\alpha)}{\rho U^2 c}, \tag{13}$$

$$C_d(\alpha) = \frac{2d(\alpha)}{\rho U^2 c}, \tag{14}$$

where l and d represent the lift and drag forces per unit span, respectively, both being functions of the airfoil's angle of attack α. The lift coefficient $C_l(\alpha)$ typically increases with α at low angles until stall occurs, beyond which it decreases. The stall angle α_s thus can be defined as the first local maximum of C_l while increasing α from 0°. Following Sheikh *et al.* (2023), we then consider the optimization of the following two composite objectives based on C_l and C_d:

$$(l/d)_{\max} \equiv \max_{\alpha} \frac{C_l(\alpha)}{C_d(\alpha)}, \tag{15}$$

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

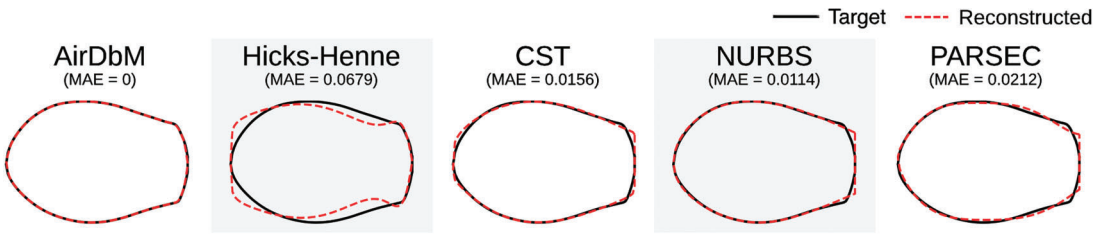


Figure 5: Reconstruction of Wortman FX 79-W-660A airfoil. Depicted are target (black solid line) versus reconstructed (red dashed line) airfoils for the present 12-baseline DbM (AirDbM), Hicks-Henne, class-shape transformation (CST), non-uniform rational B-spline (NURBS), and parametric section (PARSEC) methods.

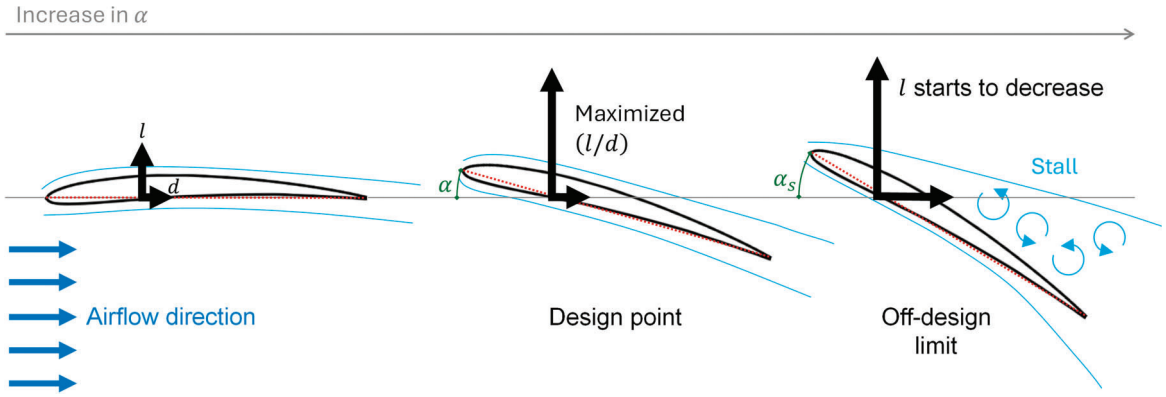


Figure 6: Airfoil performance with increasing angle of attack α, depicting changes in lift l and drag d. The figure illustrates the design point for maximum lift-to-drag ratio $(l/d)_{\max}$ and the stall tolerance, quantified by the angle of attack range from the design point to the off-design limit of stall at α_s.

$$\Delta\alpha \equiv \max\left(0,\ \alpha_s - \arg\max_{\alpha} \frac{C_l(\alpha)}{C_d(\alpha)}\right), \tag{16}$$

where first objective $(l/d)_{\max}$ is the maximum lift-to-drag ratio under the design operating condition (i.e. α associated with the maximum l/d), and the second $\Delta\alpha$ quantifies the stall tolerance, representing the angle of attack range for off-design operations. They are illustrated in Figure 6.

These aerodynamic objectives are evaluated using XFOIL 6.99, a widely-accepted inviscid/viscous zonal airfoil analysis program (Drela, 1989) for quick initial design studies, at $\mathrm{Re} = 10^6$ under incompressible flow conditions suitable for subsonic flows with negligible air density variation ($\mathrm{Ma} = U/c_s \ll 1$, where $c_s \approx 3 \times 10^2$ m/s is the speed of sound). The low computational cost of this solver enables direct exploration of the objective space (without the need for a surrogate model). Accordingly, we use the MATLAB-based non-dominated sorting genetic algorithm (NSGA-II) **gamultiobj** (Deb *et al.*, 2002), by taking each morphed airfoil shape $\mathcal{M}$'s 12 DbM weight factors $(w_1, \cdots, w_{12})$ as its genetic representation. The detailed setup of the optimization is provided in Appendix B. Readers are also recommended to refer to Sheikh *et al.* (2023, see Appendices A and B), which includes preliminary validation steps of the setup we replicate here for the sake of comparison. For instance, to ensure evaluation robustness, our XFOIL implementation employs convergence check strategies such as restarting with fresh initial guesses and correctness verification through viscous-inviscid drag coefficient comparison. These metrics and evaluation procedures were validated against the reference XFOIL evaluation database (Airfoil Tools, 2025).

Before diving into the analysis of the results, it is pertinent to distinguish expected behaviors from unusual improvements when using the reduced baseline set. Generally, decreasing design variables should accelerate convergence, requiring fewer total generations. Our previous optimization with 25 baselines ran for 3000 GA generations, and we anticipate the current one to take fewer. However, faster convergence does not guarantee a superior or even equivalent Pareto front. Dimensionality reduction inevitably compacts the design space, likely leading in our case to the minimal space encompassing the existing database. Consequently, the common expectation is that the resulting Pareto front will hardly outperform that from a larger baseline set (assuming sufficient convergence). If the optimization with the reduced design space yields enhanced solutions that dominate prior Pareto-optimal solutions, this would constitute a key improvement.

Figure 7 illustrates the progression of the hypervolume indicator—a widely adopted metric in multi-objective optimization for evaluating the quality of a set of non-dominated solutions (i.e. solutions for which no objective can be improved without degrading at least one other objective) (see Li & Yao, 2020; Guerreiro *et al.*, 2022)—throughout the GA generations for the current optimization. In the present bi-objective context, the hypervolume of a set of non-dominated solutions $((l/d)_{\max,1}, \Delta\alpha_1), \cdots, ((l/d)_{\max,k}, \Delta\alpha_k)$ is defined as the total area of the following 2D Pareto dominance region $\mathcal{R}$:

$$\mathcal{R} = \bigcup_{i=1,\cdots,k} \left\{(f_1, f_2) \in \mathbb{R}^2 \,\middle|\, 0 \le f_1 \le (l/d)_{\max,i} \text{ and } 0 \le f_2 \le \Delta\alpha_i\right\}, \tag{17}$$

where the origin $(0,\ 0)$ is taken as the reference (nadir) point. The observed evolutionary trend of the hypervolume in Figure 7, characterized by a steep initial increase in hypervolume followed by a more gradual convergence with sporadic leaps, is consistent with

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

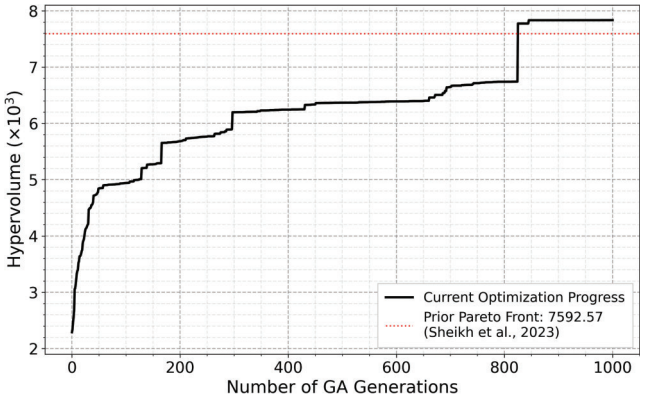


Figure 7: Hypervolume progression for the multi-objective airfoil optimization using AirDbM, tracking the hypervolume with respect to GA generations (solid black line), demonstrating improved Pareto front quality. The dotted red line indicates the hypervolume from the prior study with 25 baselines (see Sheikh *et al.*, 2023, p. 1450).

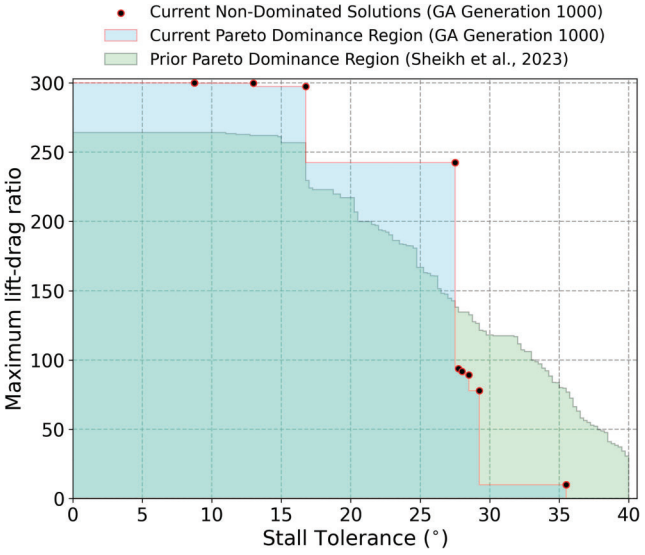


Figure 8: Pareto front comparison for airfoil optimization: AirDbM (current) versus prior work (see Sheikh *et al.*, 2023, p. 1450).

typical performance patterns reported in NSGA-II literature (e.g. Steuler *et al.*, 2020; Antoniou *et al.*, 2020).

Notably, the hypervolume achieved by the current optimization with 12 baselines surpasses the final hypervolume of 7592.57 in Sheikh *et al.* (2023), which utilized 25 baseline airfoils, at approximately the 850th generation. This milestone is achieved significantly earlier than the 3000 generations run in the previous study, underscoring the expedited convergence attributable to the reduced design-space dimensionality. While the outperformance in terms of the hypervolume indicator is a positive indication of the efficacy of AirDbM in navigating the design space, it is important to note that this is merely one measure of Pareto front quality. Accordingly, a detailed comparison of the Pareto fronts, crucial for understanding the specific trade-offs associated with the current, more compact design space, should follow.

A direct comparison of the Pareto front obtained using the current AirDbM approach after the 1000 GA generations against that from Sheikh *et al.* (2023) is presented in Figure 8. AirDbM successfully identifies new non-dominated solutions achieving significantly higher $(l/d)_{max}$, particularly at moderate stall tolerances, thereby dominating the prior Pareto front in this portion of the objective space. Nonetheless, Figure 8 also reveals that the current Pareto front does not extend to the same stall tolerance values achieved by the prior study, which found non-dominated solutions approaching $\Delta\alpha \approx 40°$. This suggests that while the 12 selected baselines enable efficient design exploration and yield improvements in certain regions, they may not possess the geometric diversity required to reproduce solutions at the extreme end of the $\Delta\alpha$ spectrum previously accessible with 25 baselines.

The inability to reach these high $\Delta\alpha$ solutions is further evidenced by attempts to reconstruct specific Pareto-optimal airfoils from the prior study. For instance, when reconstructing the prior optimal airfoil solution characterized by the highest $\Delta\alpha$ using AirDbM, the reconstruction resulted in MAE exceeding the 0.005 threshold, indicating significant discrepancies. Considering that much finer geometric tolerances, on the order of 10^{-4}, e.g. Kulfan's wind-tunnel tolerance (Kulfan & Bussoletti, 2006; Masters *et al.*, 2017), are regarded as necessary to ensure the replication of aerodynamic performance, it is likely that the observed truncation in the current Pareto front for high stall tolerance implies the bounds imposed by the reduced geometric variability of the AirDbM design space. Nevertheless, given the substantial decrease in lift-to-drag ratios typical in 3D wing applications (usually by an order of magnitude), the enhanced Pareto front in $(l/d)_{max}$ found by AirDbM can offer greater practical utility to offset 3D decrease.

Figure 9 showcases a selection of Pareto-optimal airfoil shapes obtained from the current AirDbM optimization. Excluding clustered solutions with minimal geometric differences from the presented one with $(l/d)_{max} = 77.86$ and $\Delta\alpha = 29.25°$, we present six representatives that capture the range of trade-offs between $(l/d)_{max}$ and $\Delta\alpha$.

The first three airfoils (top row) exhibit similar thin-profile geometries, achieving high lift-to-drag ratios ($(l/d)_{max} = 300.00$, 299.85, and 297.32) that represent improvements over our previous Pareto front. The $\Delta\alpha$ variance among these airfoils falls within the expected range for high $(l/d)_{max}$ optimal airfoil groups identified in our previous work. The fourth airfoil ($(l/d)_{max} = 242.56$ and $\Delta\alpha = 27.50°$) demonstrates a thicker profile that achieves greater stall tolerance at the expense of lift-to-drag ratio. This thickness-induced performance trade-off is consistent with observations from Sheikh *et al.* (2023).

The final two airfoils, while non-dominant compared to our previous Pareto front, offer instructive insights into optimization behavior. The fifth airfoil shares a geometric appearance similar to the fourth, albeit slightly thicker, but exhibits a substantial performance drop ($(l/d)_{max} = 77.86$), implying the highly nonlinear nature of aerodynamic performance. However, this can stem from potential limitations in XFOIL's solution accuracy—even when converged, performances may represent physically irrelevant solutions arising from the simplified modeling inherent to preliminary design tools. The sixth airfoil presents a distinctly different morphology: a spear-like sharp profile ($(l/d)_{max} = 10.08$, $\Delta\alpha = 35.50°$). This solution presumably represents a physically unrealistic configuration that emerges from XFOIL's inherent modeling simplifications in preliminary aerodynamic analysis.

For readers interested in the physical aspects of optimal airfoil designs resulting from this optimization, it should be carefully taken into account that, due to XFOIL's 2D nature that cannot completely capture real-world 3D wing effects (e.g. tip effects, wakes, and spanwise separations), predicted $(l/d)_{max}$ and $\Delta\alpha$ values can be excessively elevated compared to actual 3D applications. While the scope of the current test study is limited to validating design-space dimensionality effectiveness, fu-

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

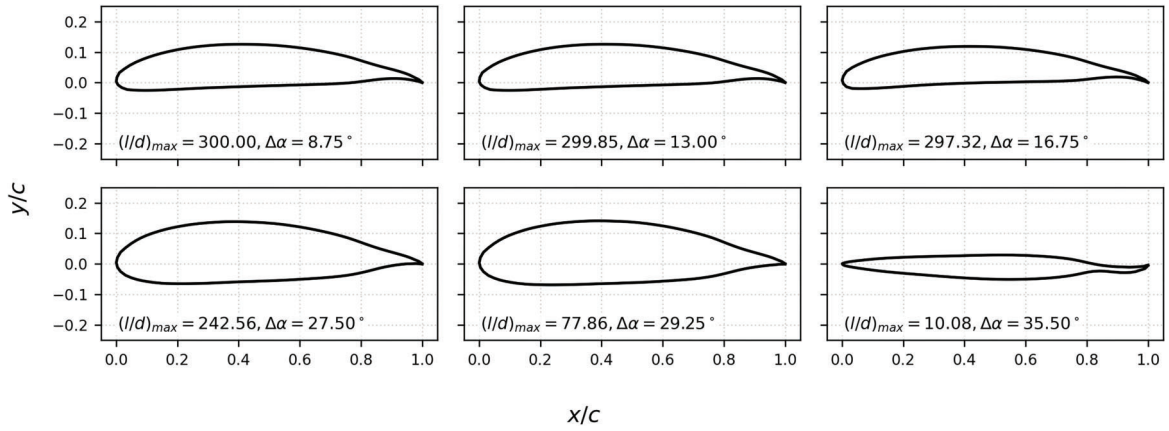


Figure 9: A selection of airfoil shapes from AirDbM's Pareto front displayed in Figure 8, arranged in descending order of $(l/d)_{max}$ from top-left to bottom right.

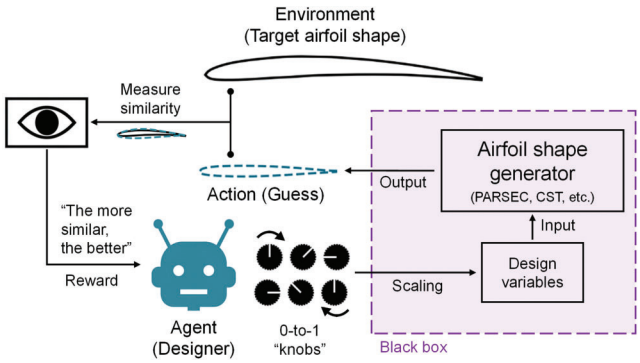


Figure 10: Reinforcement learning framework for airfoil geometry generation 'game,' where the agent (designer) manipulates normalized control inputs (tuning 'knobs') for a black-box airfoil shape generator. The agent acts to generate an airfoil 'guess' without *a priori* insight into the generator's internal process, and receives a reward based on similarity to the 'target' airfoil shape provided in the environment.

ture work will replace this preliminary-level solver with higher-fidelity solvers directly solving the Navier-Stokes equations, such as three-dimensional Reynolds-averaged Navier-Stokes (RANS) simulations. This will enable detailed physics-based analysis of solutions, building upon the framework robustness demonstrated in the current study.

4.2. Airfoil geometry learning

Through our practice of dimensionality reduction in Section 3, AirDbM has demonstrated a design capacity comparable to its predecessor using a larger baseline set and achieves a design span on par with or superior to several conventional airfoil parameterization methods with a consistent number of design variables. This similar database reconstruction rate with significantly fewer design variables underscores its effectiveness.

Additionally, from a designer's perspective, adaptability is considered as crucial as effectiveness. In this example, we assume that adaptability across different methods could be assessed by observing designers, initially unfamiliar with airfoil parameterization, while they iteratively generate shapes and improve their learning in an empirical manner. However, using *human* designers could present challenges in validating their level of unfamiliarity, leading to uncontrollable biases in the evaluation.

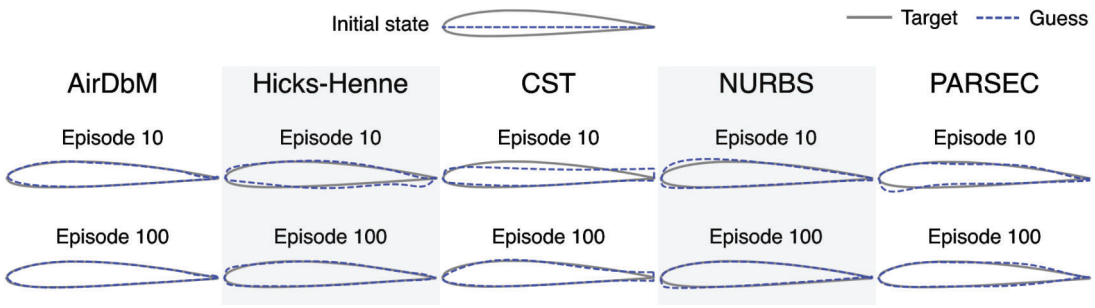
Instead, a *machine* agent driven by recent advancements in reinforcement learning (or neuro-dynamic programming) algorithms offers a compelling alternative to serve as an unbiased and initially 'ignorant' designer. Reinforcement learning (RL) enables an agent to learn optimal behavior through trial-and-error interactions within an environment by maximizing a reward signal (Bertsekas, 2019). It has seen increasing application in aerodynamic design optimization problems demanding intelligence and experience (e.g. Hui *et al.*, 2021; Patel *et al.*, 2024). In this study, we utilize this framework for airfoil geometry generation, where the agent plays a 'game' of guessing the control inputs for a *black-box* airfoil shape generator to match the output airfoil shape with the given target airfoil shape. Over multiple iterations, the agent gets empirical knowledge about input–output relations, resulting in getting a more similar guess to the target. The overall framework scheme is illustrated in Figure 10.

The most important setup in this RL framework is that the agent is completely unaware of the internal process of airfoil generation, thus the agent lacks *a priori* insight into it. To ensure complete isolation, the agent does not directly control the design variables (which might imply knowledge of the airfoil generation method). Instead, it only manipulates normalized control inputs, like tuning 'knobs' ranging from zero to one, that are linearly scaled to the design variables' bounds. By maintaining a consistent learning policy, we can then replace the airfoil parameterization method (airfoil shape generator) and assess the learning rate—how quickly the agent's guesses converge towards the target—over successive iterations (i.e. cumulative episodes).

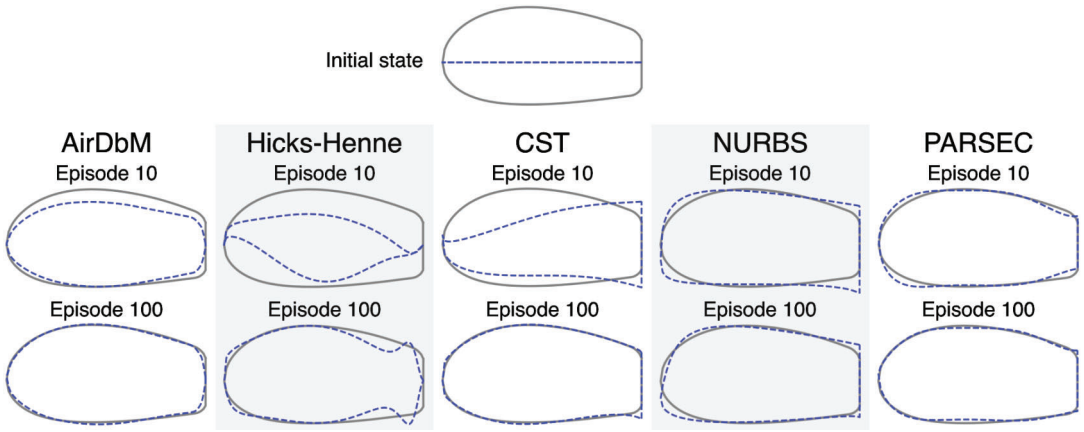
It is noteworthy that exploring different RL approaches is beyond the scope of this study. We use the **Gymnasium** framework (Towers *et al.*, 2024), training a proximal policy optimization (PPO) agent with a multi-layer perceptron (MLP) surrogate policy (Raffin *et al.*, 2021), optimizing a reward signal defined as the negatively signed MAE of the guessed airfoil shape against a target one provided in the environment. For details regarding the airfoil parameterization methods used for comparison, adhering to 12 or 13 design variables—AirDbM, Hicks–Henne, CST, NURBS, and PARSEC—and the RL setup, refer to Appendix A and Appendix C, respectively.

In Figure 11, two representative outcomes of the airfoil geometry generation 'game' are depicted: (A) NACA 2412 (thin airfoil) and (B) Althaus AH 93-W-480B (thick airfoil). Particularly, the latter case is one of the baselines of AirDbM, which is depicted to demonstrate the agent's lack of *a priori* insight into the genera-

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026



A NACA 2412



B Althaus AH 93-W-480B

Figure 11: Comparison of geometry generation for (A) NACA 2412 and (B) Althaus AH-W-480B target shapes using various airfoil shape generators. Starting from the same initial state, the guessed shapes (blue dashed line) after Episodes 10 and 100 are compared to the target airfoil (gray solid line) for AirDbM, Hicks-Henne, CST, NURBS, and PARSEC.

tion process as intended—had the agent been aware of this fact, it could have arrived at the target directly. During each episode, the agent makes 100 attempts, iteratively refining its guess based on the best outcome from previous cumulative episodes, which gradually improves its control input tuning and ultimately results in better predictions (comparing Episode 10 to Episode 100).

Compared against the other four conventional airfoil parameterization methods, AirDbM exhibits relatively fast convergence to the target in these two representative cases. For instance, looking into Episode 10 of the Althaus AH 93-W-480B environment (Figure 11 B), AirDbM's guess, albeit slightly thin yet, already becomes akin to the target airfoil shape, while the other guesses either are still far from airfoil shapes (Hicks-Henne and CST) or suffer from bloated leading edge curvature (NURBS and PARSEC). Despite the agent's unawareness, AirDbM inherently possesses the feature information of airfoils in the baselines. Therefore, for any weight inputs, the resulting shape is likely to be an airfoil shape as it is constructed by the mixture of the existing design features. In this regard, PARSEC, which more explicitly carries airfoil design features (since the design variables are directly geometric parameters of airfoils), is expected to show fast convergence but it is presumably the method's fundamental inferiority in reconstructing airfoil shapes that limits the performance (see Sheikh *et al.*, 2023, p. 1447).

For quantitative and non-prejudiced evaluation, all 1644 target airfoil shapes in the database were tested under the same learning setup. A comprehensive comparison result is shown in Figure 12. The trends of the best MAE measures achieved over cumulative episodes are plotted with respect to the five airfoil shape generation methods under consideration, where the solid line is the average of the entire 1644 environment runs at each episode while the shaded area represents ±0.25× standard deviation (the factor of 0.25 is merely for visual clarity to minimize overlapping between the shaded areas).

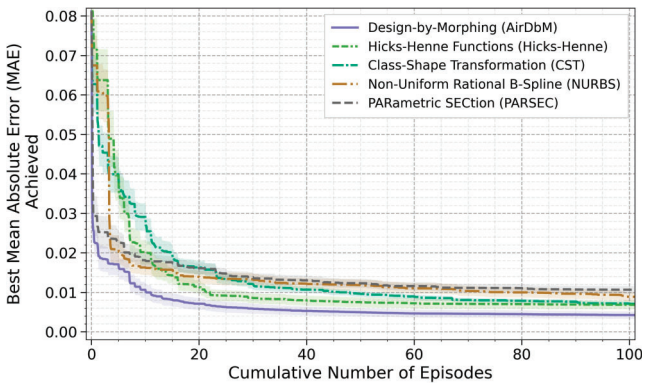


Figure 12: Quantitative evaluation of 5 airfoil shape generation methods—AirDbM (the present DbM), Hicks-Henne, CST, NURBS and PARSEC—illustrating average (solid line) and ±0.25×standard deviation (shaded area) of best MAE achieved over cumulative episodes across all 1644 target airfoil shapes tested.

In line with the findings from the representative cases, AirDbM overall exhibits the fastest decrease in the best MAE for ini-

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

tial episodes, keeping the lead up to the long run (Episode 100). PARSEC initially shows comparable decreasing rate in MAE with AirDbM, but as the episode accumulates, the performance gap widens, ultimately remaining the worst performance. The other three methods, Hicks-Henne, CST and NURBS, show relatively decent decrease, but their performances gradually improve and in the long run all arrive in between PARSEC and AirDbM.

Based on these results, it can be concluded that AirDbM, using just 12 systematically selected baseline airfoils, not only matches the reconstruction accuracy of conventional parameterizations but also excels in adaptability and learning efficiency when integrated with RL agents. In a full comparison across 1644 target airfoils in the database, AirDbM enabled the agent—as an unbiased designer of no prior knowledge—to achieve lower MAE and faster convergence than all compared airfoil parameterization methods, maintaining the same or a larger number of design variables.

5. Discussion

This study has successfully demonstrated an improved DbM approach, AirDbM, which significantly reduces design-space dimensionality for airfoil design and optimization. By focusing on maintaining design diversity through effective reconstruction of the rich airfoil database, our study achieved a substantial reduction in the number of baseline shapes required. The resulting AirDbM approach not only yielded benefits in multi-objective aerodynamic optimization, such as accelerated convergence and even partial enhancement of the Pareto-optimal solutions, as demonstrated in our former example (Section 4.1), but also showed excellence in airfoil shape generation compared to several conventional parameterization methods.

Nonetheless, it should be admitted that the dimensionality reduction in AirDbM primarily concentrated on the *geometric* feature preservation. Although this approach ensures broad geometric coverage, the aerodynamic optimization results suggest a trade-off, as exemplified by the inability to reach the extreme stall tolerance values achieved previously with a larger baseline set. Such a geometrically focused compact design space appears to limit the exploration of more aerodynamically diverse or specialized design candidates.

Thus, future developments of the DbM framework for airfoil design could possibly benefit from incorporating aerodynamic considerations more directly into the baseline selection process. Beyond geometric diversity, selecting or augmenting baseline sets with airfoils known for specific, superior aerodynamic characteristics (e.g. high C_l and low C_d) could allow the design space to better support the exploration of dynamically advanced airfoil designs. In particular, considering the reconstruction rate convergence observed (see Figure 4), one may only choose the first 10 or 11 airfoils from the current baseline set and supplement the remaining slots with designs proven for their aerodynamic merit, creating a *hybrid* baseline set. This can also be supplemented by additional dimensionality reduction efforts that preserve the essential design space scope, such as employing n-sphere coordinate variables for $(n+1)$ weight factor mapping, or pruning the design space by leveraging internal problem symmetries (e.g. Lee *et al.*, 2024).

It is clear that the significant design-space dimensionality reduction achieved by AirDbM mitigates the *curse of dimensionality*. The reduction in the number of design variables is crucial as it opens up possibilities for integrating more computationally intensive or higher-fidelity solvers into the efficacious optimization loop. The current reliance on inexpensive solvers like XFOIL 6.99 facilitates rapid design exploration, but could be replaced or augmented by, for example, RANS simulations or even experiments. In this way, we could pave the way for optimizing more realistic, 3D wing designs or tackling more complex aerodynamic phenomena necessitating higher fidelity. Such an approach requires efficient optimization algorithms with smaller data points of exploration rather than large-sample algorithms like GA, such as those based on Bayesian inference (e.g. Sheikh *et al.*, 2022) or the use of PPO agents explored in our latter example (Section 4.2), for sample efficiency.

The data-efficient, interpretable parameterization of AirDbM (and DbM more broadly) reveals significant implications for machine learning-driven design. While deep generative models like generative adversarial networks (GANs; e.g. Chen *et al.*, 2020; Wang *et al.*, 2023; Xie *et al.*, 2024) excel at synthesizing novel designs through data-driven pattern recognition, they typically require thousands of training samples and lack inherent physical constraints. DbM can address these issues by providing geometrically consistent priors through systematic morphing of a baseline set containing $O(10^1)$ or perhaps fewer elements, generating physically plausible candidate designs that can seed and constrain GAN training. This symbiotic relationship enables generative models to focus on refining physical meaningful variations in avoidance of suffering from hallucinations of non-feasible geometry generations, leading to reduced training time while maintaining design feasibility. To sum up, while recent machine learning-driven methods (e.g. GANs and variational autoencoders, VAEs) reduce latent-space dimensionality when large datasets are available, DbM begins with a few known baseline designs to span a wide and physically relevant design space. These approaches imply their synergistic relation: DbM is not a direct competitor to these methods, but could rather be complementary.

6. Conclusions

We addressed the challenge of reducing design-space dimensionality in DbM for airfoil optimization by introducing AirDbM, an DbM-based airfoil design approach with a systematically reduced baseline set. Utilizing an effective forward search strategy, we identified a compact yet highly representative set of 12 baseline airfoils selected from the UIUC database of 1644 airfoils. This reduced set retained broad airfoil design capability, as demonstrated in reconstruction tests where 98% of the database was reproduced within an MAE of 0.005. This performance rivals—and, in terms of dimensionality, surpasses—the previous DbM efforts that used 25 baselines, thereby achieving a substantial reduction in design parameters without compromising geometric diversity.

The efficacy of AirDbM was quantitatively demonstrated in both multi-objective aerodynamic optimization using a GA and airfoil geometry generation in the context of reinforcement learning. In aerodynamic shape optimization aimed at maximizing both lift-to-drag ratio and stall tolerance, AirDbM achieved accelerated convergence. Its hypervolume indicator value surpassed that of the earlier 25-baseline study in significantly fewer GA generations. The resulting Pareto front identified new Pareto-optimal solutions with enhanced lift-to-drag ratios, especially at low to moderate stall tolerances. In a comparative study that employs machine agents as unbiased designers from a reinforcement learning framework, AirDbM achieved faster convergence and lower errors than several conventional airfoil parameterization methods, while using a similar number of design variables.

These findings lay the groundwork for further advancements in the DbM methodology for airfoil design and optimization. Future

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

directions may include incorporating aerodynamic performance criteria into the baseline selection process to create hybrid sets that maintain geometric representativeness while targeting specific aerodynamic objectives. Additionally, the computational efficiency gained from operating in a lower-dimensional design space facilitates the integration of higher-fidelity solvers, paving the way for a transition from 2D airfoil analysis to more realistic 3D wing design applications. Such developments are anticipated to be synergistically combined with modern machine learning-driven generative design approaches for expedited optimization.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Sangjoon Lee: Formal analysis, Investigation, Methodology, Software, Writing—original draft. **Haris Moazam Sheikh:** Conceptualization, Methodology, Project administration, Supervision, Writing—review & editing.

Funding

This work used Anvil at Purdue RCAC through allocation PHY250071 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

Data Availability

The present DbM (AirDbM) codes and data, incorporated into the multi-objective airfoil optimization problem in this study, is available in the public repository on https://github.com/UCBCFD/DbMAirfoilOpt. Additional codes or data can be shared on a reasonable request to the corresponding author.

Acknowledgments

A significant portion of the computations for data analyses was performed on the Yellowstone at the Stanford HPC Center, supported through awards from Intel, National Science Foundation, DOD HPCMP, and Office of Naval Research.

References

Airfoil Tools (2025). Retrieved from http://airfoiltools.com/. Accessed 21 October 2025.

Anderson, J. D. (1997). *Aerodynamics in the Age of Strut-and-Wire Biplanes, Cambridge Aerospace Series*(pp. 267–318). Cambridge University Press.

Antoniou, M., Petelin, G., & Papa, G. (2020). On formulating the ground scheduling problem as a multi-objective bilevel problem. In Filipič B., Minisci E., & Vasile M. (Eds.), *Bioinspired Optimization Methods and Their Applications*(pp. 177–188). Springer International Publishing. https://doi.org/10.1007/978-3-030-63710-1_14

Bai, X., Zhang, X., & Yuan, C. (2016). Numerical analysis of drag reduction performance of different shaped riblet surfaces. *Marine Technology Society Journal*, **50**, 62–72. https://doi.org/10.4031/MTSJ.50.1.9.

Benjamin, M., & Iaccarino, G. (2025). A systematic dataset generation technique applied to data-driven automotive aerodynamics. *APL Machine Learning*, **3**, 016110. https://doi.org/10.1063/5.0233367.

Bertsekas, D. (2019). *Reinforcement Learning and Optimal Control*(1st edn.). Athena Scientific Optimization and Computation Series, Athena Scientific.

Bilstein, R. E. (1989). *Orders of Magnitude*(3rd edn.). National Aeronautics and Space Administration.

Blank, J., & Deb, K. (2020). pymoo: Multi-objective optimization in Python. *IEEE Access*, **8**, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567.

Chen, W., Chiu, K., & Fuge, M. D. (2020). Airfoil design parameterization and optimization using Bézier generative adversarial networks. *AIAA Journal*, **58**, 4723–4735. https://doi.org/10.2514/1.J059317.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**, 182–197. https://doi.org/10.1109/4235.996017.

Derksen, R., & Rogalsky, T. (2010). Bezier-PARSEC: An optimized aerofoil parameterization for design. *Advances in Engineering Software*, **41**, 923–930. https://doi.org/10.1016/j.advengsoft.2010.05.002.

Drela, M. (1989). *XFOIL: An Analysis and Design System for low Reynolds Number Airfoils*(pp. 1–12). Springer.

Drela, M. (1998). Pros & cons of airfoil optimization. In *Frontiers of Computational Fluid Dynamics 1998*(pp. 363–381). World Scientific.

Gordon, W. J., & Riesenfeld, R. F. (1974). Bernstein-Bézier methods for the computer-aided design of free-form curves and surfaces. *Journal of the ACM*, **21**, 293–310. https://doi.org/10.1145/321812.321824.

Guerreiro, A. P., Fonseca, C. M., & Paquete, L. (2022). The hypervolume indicator. *ACM Computing Surveys*, **54**, 1–42. https://doi.org/10.1145/3453474.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, **3**, 1157–1182.

Hicks, R. M., & Henne, P. A. (1978). Wing design by numerical optimization. *Journal of Aircraft*, **15**, 407–412. https://doi.org/10.2514/3.58379.

Hui, X., Wang, H., Li, W., Bai, J., Qin, F., & He, G. (2021). Multi-object aerodynamic design optimization using deep reinforcement learning. *AIP Advances*, **11**, 085311. https://doi.org/10.1063/5.0058088.

Jernell, L. S. (1974). Comparisons of two-dimensional shock-expansion theory with experimental aerodynamic data for delta-planform wings at high supersonic speeds. NASA Technical Note NASA-TN-D-7583, NASA Langley Research Center. https://ntrs.nasa.gov/api/citations/19740018331/downloads/19740018331.pdf.

Jing, S., Zhao, Q., Zhao, G., & Wang, Q. (2023). Multi-objective airfoil optimization under unsteady-freestream dynamic stall conditions. *Journal of Aircraft*, **60**, 293–309. https://doi.org/10.2514/1.C036708.

Jung, J., & Gu, G. X. (2024). Data-driven airfoil shape optimization framework for enhanced flutter performance. *Physics of Fluids*, **36**, 101706. https://doi.org/10.1063/5.0232055.

Kulfan, B., & Bussoletti, J. (2006). "Fundamental" parameteric geometry representations for aircraft component shapes. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2006-6948

Lee, S., Sheikh, H. M., Lim, D. D., Gu, G. X., & Marcus, P. S. (2024). Bayesian-optimized riblet surface design for turbulent drag reduction via design-by-morphing with large eddy simulation. *Jour-*

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

nal of Mechanical Design, **146**, 081701. https://doi.org/10.1115/1.4064413.

Lepine, J., Trepanier, J.-Y., & Pepin, F. (2000). Wing aerodynamic design using an optimized NURBS geometrical representation. In *38th Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2000-669

Li, J., Du, X., & Martins, J. R. (2022). Machine learning in aerodynamic shape optimization. *Progress in Aerospace Sciences*, **134**, 100849https://doi.org/10.1016/j.paerosci.2022.100849.

Li, M., & Yao, X. (2020). Quality evaluation of solution sets in multiobjective optimisation. *ACM Computing Surveys*, **52**, 1–38. https://doi.org/10.1145/3300148.

Lyu, Z., Kenway, G. K. W., & Martins, J. R. R. A. (2015). Aerodynamic shape optimization investigations of the common research model wing benchmark. *AIAA Journal*, **53**, 968–985. https://doi.org/10.2514/1.J053318.

Martins, J. R. (2022). Aerodynamic design optimization: Challenges and perspectives. *Computers & Fluids*, **239**, 105391. https://doi.org/10.1016/j.compfluid.2022.105391.

Masters, D. A., Taylor, N. J., Rendall, T. C., Allen, C. B., & Poole, D. J. (2017). Geometric comparison of aerofoil shape parameterization methods. *AIAA Journal*, **55**, 1575–1589. https://doi.org/10.2514/1.J054943.

MathWorks (2024). gamultiobj. https://uk.mathworks.com/help/gads/gamultiobj.html.

Rocklin, M. (2015). Dask: Parallel Computation with Blocked algorithms and Task Scheduling. In *Proceedings of the 14th Python in Science Conference*(pp. 126–132). scipy.org. https://doi.org/10.25080/Majora-7b98e3ed-013

Oh, S., Jiang, C.-H., Jiang, C., & Marcus, P. S. (2018). Finding the optimal shape of the leading-and-trailing car of a high-speed train using design-by-morphing. *Computational Mechanics*, **62**, 23–45. https://doi.org/10.1007/s00466-017-1482-4.

Parent, R. (2012). *Interpolation-Based Animation*(3rd edn., pp. 111–160). Elsevier.

Patel, J., Spyridis, Y., & Argyriou, V. (2024). Enhancing vehicle aerodynamics with deep reinforcement learning in voxelised models. In *2024 14th International Conference on Pattern Recognition Systems (ICPRS)*(pp. 1–7). IEEE.

Piegl, L., & Tiller, W. (1996). *The NURBS Book, Monographs in Visual Communication*(2nd edn). Springer Science & Business Media.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, **22**, 1–8.

Raffin, A., Gallouédec, Q., Dormann, N., Gleave, A.,Anssi, Pasquali, A., Rocamonde, J., Ernestus, M., Helm, P.,Corentin,Quinn, Simonini, T., Rohrer, T., Tio, S., Towers, M., Tangri, R., DÖrr, T.,Wilson, Wang, S. H.,... Passault, G. (2025). Stable-Baselines3: v2.6.0. Source Code.

Sederberg, T. W., & Greenwood, E. (1992). A physically based approach to 2–D shape blending. *ACM SIGGRAPH Computer Graphics*, **26**, 25–34. https://doi.org/10.1145/142920.134001.

Sederberg, T. W., & Parry, S. R. (1986). Free-form deformation of solid geometric models. *ACM SIGGRAPH Computer Graphics*, **20**, 151–160. https://doi.org/10.1145/15886.15903.

Selig, M. S. (2024). UIUC Airfoil Coordinates Database (Version 2.0). https://m-selig.ae.illinois.edu/ads/coord_database.html.

Serani, A., & Diez, M. (2024). A survey on design-space dimensionality reduction methods for shape optimization. In *Archives of Computational Methods in Engineering*, Springer Science and Business Media B.V. https://doi.org/10.1007/s11831-025-10349-x.

Sheikh, H. M., Callan, T. A., Hennessy, K. J., & Marcus, P. S. (2022). Optimization of the shape of a hydrokinetic turbine)s draft tube and hub assembly using design-by-morphing with Bayesian optimization. *Computer Methods in Applied Mechanics and Engineering*, **401**, 115654. https://doi.org/10.1016/j.cma.2022.115654.

Sheikh, H. M., Lee, S., Wang, J., & Marcus, P. S. (2023). Airfoil optimization using design-by-morphing. *Journal of Computational Design and Engineering*, **10**, 1443–1459. https://doi.org/10.1093/jcde/qwad059.

Skinner, S., & Zare-Behtash, H. (2018). State-of-the-art in aerodynamic shape optimisation methods. *Applied Soft Computing*, **62**, 933–962. https://doi.org/10.1016/j.asoc.2017.09.030.

Sobester, A., & Barrett, T. (2008). Quest for a truly parsimonious airfoil parameterization scheme. In *The 26th Congress of ICAS and 8th AIAA ATIO*, American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2008-8879

Sobieczky, H. (1999). Parametric airfoils and wings. in *Recent Development of Aerodynamic Design Methodologies*, pp. 71–87, Vieweg+Teubner Verlag.https://doi.org/10.1007/978-3-322-89952-1_4

Steuler, J. L., Beck, M., Passow, B. N., & Guckert, M. (2020). Optimizing the energy consumption of neural networks. in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*(pp. 2312–2319). IEEE. https://doi.org/10.1109/SSCI47803.2020.9308576

Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., Cola, G. D., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Tan, H., & Younis, O. G. (2024). Gymnasium: A standard interface for reinforcement learning environments. https://arxiv.org/abs/2407.17032.

van den Bergh, J., di Fiore, F., Claes, J., & van Reeth, F. (2002). Interactively morphing irregularly shaped images employing subdivision techniques. In *1st Ibero-American Symposium in Computer Graphics (SIACG2002)*(pp. 315–321). European Association for Computer Graphics.

Viswanath, A., Forrester, A. I. J., & Keane, A. J. (2011). Dimension reduction for aerodynamic design optimization. *AIAA Journal*, **49**, 1256–1266. https://doi.org/10.2514/1.J050717.

Wang, Y., Shimada, K., & Farimani, A. B. (2023). Airfoil gan: encoding and synthesizing airfoils for aerodynamic shape optimization. *Journal of Computational Design and Engineering*, **10**, 1350–1362. https://doi.org/10.1093/jcde/qwad046.

Xie, H., Wang, J., & Zhang, M. (2024). Parametric generative schemes with geometric constraints for encoding and synthesizing airfoils. *Engineering Applications of Artificial Intelligence*, **128**, 107505. https://doi.org/10.1016/j.engappai.2023.107505.

Zhang, R., Li, D., Chang, H., Wei, X., & Wang, H. (2024). Multi-objective optimization on blade airfoil of vertical axis wind turbine. *Physics of Fluids*, **36**, 085172. https://doi.org/10.1063/5.0220194.

Appendix A. Airfoil Design Methods

Throughout this work, we considered five distinct airfoil parameterization methods for comparison: AirDbM, Hicks-Henne, CST, NURBS, and PARSEC, as detailed in Table A1. A key aspect of this setup was the standardization of the dimensionality across these methods. The DbM, Hicks-Henne, CST, and PARSEC methods are configured to utilize 12 design variables. The NURBS parameterization was slight exception, making use of 13 design variables; this number was chosen to maintain the method's parametric integrity while aligning it as closely as possible with the 12-design variable target used by the other methods. This consistent dimensionality facilitated a fair comparison of the different methods' capabilities in airfoil geometry generation.

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

Table A1: Airfoil parameterization methods considered for comparison.

Method	Design Variables (DVs)	Remarks
AirDbM	w_i: Morphing weight factors $\in [-1.0, 1.0]$ $(i = 1, \cdots, 12)$	See Table 1 for the baselines
Hicks-Henne	$p_{u,i}$: Upper bump powers $\in [1.0, 4.0]$ $(i = 1, 2, 3)$ $a_{u,i}$: Upper bump amplitudes $\in [-0.2, 0.2]$ $(i = 1, 2, 3)$ $p_{l,i}$: Lower bump powers $\in [1.0, 4.0]$ $(i = 1, 2, 3)$ $a_{l,i}$: Lower bump amplitudes $\in [-0.2, 0.2]$ $(i = 1, 2, 3)$	Base: flat plate; cosine-distributed bump points; see Hicks & Henne (1978)
CST	N_1: 1st class function exponent $\in (0.0, 2.0]$ N_2: 2nd class function exponent $\in (0.0, 2.0]$ $A_{u,i}$: Upper Bernstein coefficients $\in [-0.5, 0.5]$ $(i = 1, \cdots, 4)$ $\Delta\xi_u$: Upper trailing edge height $\in [-0.5, 0.5]$ $A_{l,i}$: Lower Bernstein coefficients $\in [-0.5, 0.5]$ $(i = 1, \cdots, 4)$ $\Delta\xi_l$: Lower trailing edge height $\in [-0.5, 0.5]$	See Kulfan & Bussoletti (2006)
NURBS	x_1: 1st control point's x-coordinate $\in [0.0, 1.0]$ y_1: 1st control point's y-coordinate $\in [-0.5, 0.5]$ x_2: 2nd control point's x-coordinate $\in [-0.5, 0.5]$ y_2: 2nd control point's y-coordinate $\in [-0.5, 0.5]$ x_3: 3rd control point's x-coordinate $\in [0.0, 1.0]$ y_3: 3rd control point's y-coordinate $\in [-0.5, 0.5]$ $y_{te,u}$: Upper trailing edge height $\in [-0.5, 0.5]$ $y_{te,l}$: Lower trailing edge height $\in [-0.5, 0.5]$ ω_i: Control point weights $\in [0.1, 5.0]$ $(i = 1, \cdots, 5)$	3rd-order B-spline with evenly distributed knots; see Piegl & Tiller (1996)
PARSEC	$r_{le,u}$: Upper leading edge radius $\in [0.0, 1.0]$ x_u: Upper crest's x-coordinate $\in (0.0, 1.0)$ y_u: Upper crest's y-coordinate $\in [-0.5, 0.5]$ $y_{xx,u}$: Upper crest curvature $\in [-0.5, 0.5]$ $r_{le,l}$: Lower leading edge radius $\in [0.0, 1.0]$ x_l: Lower crest's x-coordinate $\in (0.0, 1.0)$ y_l: Lower crest's y-coordinate $\in [-0.5, 0.5]$ $y_{xx,l}$: Lower crest curvature $\in [-0.5, 0.5]$ y_{te}: Trailing edge mid-position $\in [-0.5, 0.5]$ t_{te}: Trailing edge thickness $\in [0.0, 1.0]$ α_{te}: Trailing edge direction $\in [-\pi/4, \pi/4]$ β_{te}: Trailing edge wedge angle $\in [0, \pi/2]$	See Sobieczky (1999)

Appendix B. Multi-Objective Airfoil Optimization Setup

As detailed in Section 4.1, the multi-objective airfoil optimization was conducted using the **gamultiobj** optimizer in MATLAB (MathWorks, 2024), based on NSGA-II. The optimization aimed to identify superior airfoil designs, each parameterized by 12 morphing weight variables founded upon AirDbM that range from −-1 to 1, with respect to lift-to-drag ratio as a primary design point and stall tolerance as a robustness for off-design operation.

The optimizer was configured with a population size of 372 individuals and a maximum of 1000 generations. The evolution of this population was driven by the following genetic operators: selection based on a tournament approach considering non-domination rank and crowding distance (calculated in the objective or fitness function space), an intermediate crossover strategy with a crossover fraction of 0.8, and an adaptive feasible mutation scheme that introduces variations using randomly generated directions adapting to the previous generation. Following the initialization of Sheikh *et al.* (2023), the initial population was composed by incorporating outcomes from two preliminary single-objective GA runs (each with a population of 128 and run for 100 generations) for each of the design targets—lift-to-drag ratio and stall tolerance. The remaining individuals of the initial population were randomly distributed. A summary of these algorithmic parameters is provided in Table B1.

Table B1: Details of the multi-objective GA used in this study.

Option	Selection
Population size	372
Total generations	1000
Selection scheme	Binary tournament (Pareto fraction = 0.35)
Crossover scheme	Intermediate crossover (Crossover fraction = 0.8)
Mutation scheme	Adaptive feasible
Distance measure of individuals	Crowding distance in fitness function space

At each generation, the XFOIL performance evaluations were parallelized; this study utilized up to 128 cores to perform XFOIL analyses concurrently for 128 airfoil samples, significantly reducing the overall duration of the optimization process.

Appendix C. Airfoil Geometry Learning Setup

In Section 4.2, the airfoil geometry generation task was formulated as a reinforcement learning problem and addressed using a PPO agent. The PPO agent utilized an MLP for both the actor and critic networks. Key hyperparameters for the PPO algorithm included the learning rate of 0.0003, the number of steps per update of 2048, the batch size of 64, the epochs per update of 10,

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

the discount factor of 0.99, the generalized advantage estimator (GAE) lambda of 0.95, the PPO clipping range of 0.2, the entropy coefficient of 0, and the value function coefficient of 0.5. Other PPO parameters largely followed the default values as in the Stable-Baselines3 library (v2.6.0) (Raffin *et al.*, 2025).

The learning environment was configured for episodic tasks. Each episode consisted of 100 steps. The observation space provided to the agent was the set of 12 or 13 normalized inputs ranging from 0.0 to 1.0. The reward at each step was calculated as the negative of the MAE, incentivizing the agent to produce airfoils more closely matching the target. Each PPO model was trained for a total of 10 000 steps (i.e. 100 episodes).

Downloaded from https://academic.oup.com/jcde/article/13/1/108/8321690 by guest on 26 January 2026

Received: July 3, 2025. **Revised:** October 21, 2025. **Accepted:** October 21, 2025
© The Author(s) 2025. Published by Oxford University Press on behalf of the Society for Computational Design and Engineering. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.