Diego I. Lopez[1]
Department of Mechanical, Chemical and
Materials Engineering,
University of Cagliari,
Cagliari 09123, Italy
e-mail: diegoi.lopez@unica.it

Tiziano Ghisu
Department of Mechanical, Chemical and
Materials Engineering,
University of Cagliari,
Cagliari 09123, Italy
e-mail: t.ghisu@unica.it

Shahrokh Shahpar
Innovation Hub, Central Technology,
Rolls-Royce plc,
Derby DE24 8BJ, UK
e-mail: shahrokh.shahpar@rolls-royce.com

Global Optimization of a Transonic Fan Blade Through AI-Enabled Active Subspaces

The increased need to design higher performing aerodynamic shapes has led to design optimization cycles requiring high-fidelity CFD models and high-dimensional parametrization schemes. The computational cost of employing global search algorithms on such scenarios has typically been prohibitive for most academic and industrial environments. In this paper, a novel strategy is presented that leverages the capabilities of artificial neural networks for regressing complex unstructured data, while coupling them with dimensionality reduction algorithms. This approach enables employing global-based optimization methods on high-dimensional applications through a reduced computational cost. This methodology is demonstrated on the efficiency optimization of a modern jet engine fan blade with constrained pressure ratio. The outcome is compared against a state-of-the-art adjoint-based approach. Results indicate that the strategy proposed achieves comparable improvements to its adjoint counterpart with a reduced computational cost and can scale better to multi-objective optimization applications. [DOI: 10.1115/1.4052136]

Keywords: computational fluid dynamics (CFD), fan, compressor, and turbine aerodynamic design

1 Introduction

Modern day aircraft engines are required to meet increasingly tighter regulations in terms of their performance and by-product emissions, which increases the need for every component in the system to be carefully designed so as to achieve its operational requirements while employing the least amount of resources.

Design optimization based on computational fluid dynamics (CFD) is one of the primary tools engineers have used to improve their aerodynamic designs [1]. In recent years, the need for higher performing shapes has led to an increased need for higher fidelity CFD models, which are able to represent more faithfully the complex physical characteristics of the flow. In addition, capitalizing on the increased fidelity requires a finer control over the geometry, enabling more degrees-of-freedom to respond to even the smallest flow features, thus making the optimization process more likely to lead to novel, high-performing designs.

Gradient-based optimization methods have lately been the subject of many studies, due to the increased availability of adjoint flow solvers. The adjoint method [2,3] allows computation of the gradient of an objective function with respect to design parameters while maintaining the cost of the operation almost independent of the number of parameters employed. This approach has been successfully applied to optimize turbomachinery components with high-dimensional parametrization schemes [4–9]. However, as described in Ref. [3], the residual of the Navier–Stokes equations is assumed to be zero when computing the adjoint, which requires the primal CFD to achieve exceptional convergence and a near-zero residual. This implies longer solve times than would be employed for traditional CFD analysis and a corresponding increase in the computational expense of the flow solver. Moreover, as documented in Ref. [5], typical running times for discrete adjoint solvers are about three times higher than their primal counterpart, making the adjoint calculation a costly endeavor. In optimization scenarios involving several cost function (CF) or constraints that depend on the flow behavior, the adjoint calculation must be repeated for each objective and constraint, reducing the scalability of this approach. In addition, much of the outcome of gradient-based approaches is dictated by the initial design and, if starting from already high-performing shapes close to a local optimum, the optimization process might not yield significantly improving designs, regardless of the parametrization scheme.

Global search algorithms, on the other hand, only require zeroth order information, which can usually be obtained with one CFD computation for any number of objectives. Moreover, these algorithms are stochastic in nature and this, in principle, makes them capable of identifying the global optimum of the CF. The application of such algorithms to turbomachinery design optimization has been an active area of research, with approaches falling into one of two categories: *direct* application to the CFD solution [10–12] and *surrogate-based* optimization, where a number of computations are first employed to regress the behavior of the CF in the design space and the search algorithm is subsequently applied on this analytical model [13,14]. Global search algorithms typically require function evaluations on the order of hundreds to thousands, depending on the dimensionality of the problem, which causes direct approaches on high-fidelity CFD to be prohibitive for most academic and industrial environments.

Indeed, with the advent of machine learning and data-driven methods, novel surrogate models are being researched which enable highly accurate predictions over complex high-dimensional datasets, making them ideal for global optimizations. In particular, artificial intelligence (AI), or *artificial neural networks* (ANNs) have become a primary area of study due to their capability to interpolate unstructured data. Unlike other machine learning methods, ANNs do not assume a predefined shape for the CF, thus providing a higher level of flexibility and predictive accuracy in cases where the system cannot be well expressed through explicit functions. In the field of turbomachinery, they have been applied to develop physics-based models for unsteady compressor behavior [15] and on global, surrogate-based optimization environments [16–18]. However, the fact that they are unstructured also implies that the number of data points required to adequately fit them to high-dimensional, multi-modal systems—such as the parametrized flow

[1] Corresponding author.

Contributed by the International Gas Turbine Institute (IGTI) of ASME for publication in the Journal of Turbomachinery. Manuscript received July 6, 2021; final manuscript received August 5, 2021; published online September 21, 2021. Tech. Editor: David G. Bogard.

behavior of transonic blades—is typically higher than other methods. Thus, the cost-effectiveness of ANNs can be enhanced by coupling them with dimensionality reduction (DR) algorithms.

Several of such DR algorithms exist and have been extensively documented, such as principal component analysis (PCA) [19] or Sobol' sensitivity indices [20]. In recent years, a novel set of ideas that facilitate subspace-based dimension reduction has emerged, called *active design subspaces* (ADS) [21]. Instead of determining a subset of input parameters which are most important, ADS identifies dominant linear combinations of all the parameters that best describe the variability in the output. While PCA is also able to identify such subspaces from the parameter space, with that approach the data must be correlated or conditioned by some process, like the pareto-optimal design points from a multidisciplinary optimization cycle [11]. ADS, on the other hand, can learn the dominant subspaces from a non-conditioned dataset, which enables its application to uniformly spread sampled points, such as those coming from a design of experiment approach, which are typically used in regression processes. In the field of turbomachinery, active subspaces have been linked with fundamental aerodynamic principles to infer pedigree design rules to improve a compressor's performance [22], to visualize the robustness of a design point with respect to uncertain parameters [8], and to identify dominant subspaces for stagnation temperature probes [23]. The ADS process requires estimating the gradient of the CF at various random points in the design space, for which Giugno et al. [8] employed adjoint computations and Seshadri et al. [22] regressed a second-order polynomial to CFD data and evaluated the gradients from that model. Both these approaches suffer from previously described issues: the former does not scale well for multi-objective scenarios and the latter infers that the function behavior can be well described by a second-order polynomial.

In this paper, a novel global optimization strategy is proposed exploiting the unstructured nature of ANNs and coupling them with ADS to reduce the dimensionality of the input space, thus providing an ameliorated performance with minimal samples. The remainder of this paper is structured as follows: after a description of the ANN fitting procedure in Sec. 2.1, Sec. 2.2 details the coupling between the neural networks and ADS. This link is exploited in Sec. 2.3, where the optimization problem is reformulated to efficiently explore the active subspaces. Next, Sec. 4 undertakes the optimization of a modern jet engine fan blade with constrained pressure ratio (PR). After a brief introduction to the computational tools employed in Sec. 4.1, Sec. 4.3 applies the novel strategy proposed to find a globally optimum design. These results are compared against the optimum design coming from a state-of-the-art adjoint-based approach, which is described in Sec. 4.4.

2 Global Optimization Strategy

A general optimization problem can be expressed as in Eq. (1) for one objective function and n_c constraints. This problem is typically solved employing numerical search algorithms that require repeated evaluations of the objective and constraint functions and, in some cases, higher order information like the function's gradient or Hessian matrix. The number of iterations of the optimization process is usually dictated by the dimensionality and range of the design vector, $\mathbf{x}$, and whether a local or global optimum is desired.

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) \\ &\text{subject to} \quad g_i(\mathbf{x}) \leq h_i,\ i = 1, \ldots, n_c \\ &\mathbf{x} \in \chi \subset \mathbb{R}^m \end{aligned} \tag{1}$$

The strategy developed in this study is tailored for cases where a global optimum is sought, function evaluations are performed through costly high-fidelity CFD simulations, and the inputs' dimension, m, is arbitrarily large. Moreover, no inferences are made on the complexity of the objective or constraint functions.

2.1 Fitting Artificial Neural Networks.

Artificial neural networks are mathematical constructions where information from a set of inputs, $\mathbf{v}$, is linked, in a non-linear manner, to a set of outputs $\hat{f}(\mathbf{v})$ by having the information flow across a structure of neurons, whose behavior can be trained to achieve a desired outcome. The parameters influencing the neuron's behavior are the weights, T, the biases, $\mathbf{b}$, and the activation function a, defined through an optimization process where the CF is typically designed to maximize the network's prediction accuracy over a training dataset, for which the true values of the function are known. An in-depth description of neural networks is beyond the scope of this paper; the reader is referred to Ref. [24] for an introduction to the topic.

For this application, ANNs are selected to regress the behavior of the CF and constraints because, as detailed in Refs. [25,24], provided there are sufficient neurons in the network, ANNs are able to fit any function of arbitrary shape, which makes them ideal for interpolating complex multi-modal systems, typically encountered in turbomachinery design optimization problems. A traditional perceptron feed-forward network structure is proposed with a backpropagation algorithm for gradient estimation, as shown schematically in Fig. 1. This work employs the tensorflow framework to construct and train the networks, through its python wrapper Keras [26]. The hyper-parameters defining the network architecture, such as the number of hidden layers (HLs) or number of neurons in each HL, is defined through a hyper-parameter tuning process. Since this work deals mainly with small datasets, in the order of hundreds of CFD samples, the training of the networks is not a costly procedure. Therefore, the hyper-parameter tuning can be achieved through a grid search, evaluating a large number of different parameter combinations. Due to the potentially very large number of neurons being employed, overfitting the neural networks is a possible but unwelcome outcome of the training process. To prevent this scenario, a Tikhonov regularization term [27] is included, as well as neuron drop-out. For the activation function, only two options were allowed, detailed in Table 1, along with the other hyper-parameters and their corresponding ranges. The objective function for the tuning process is to maximize the R^2 between the predictions of the network and the values from a testing set composed of CFD simulations. Some network parameters are general for this application and thus are fixed for all network configurations evaluated in the grid search:

- Number of neurons in the input layer: set to match the dimensionality of the design vector.
- Output layer definition: a single neuron without any activation function.

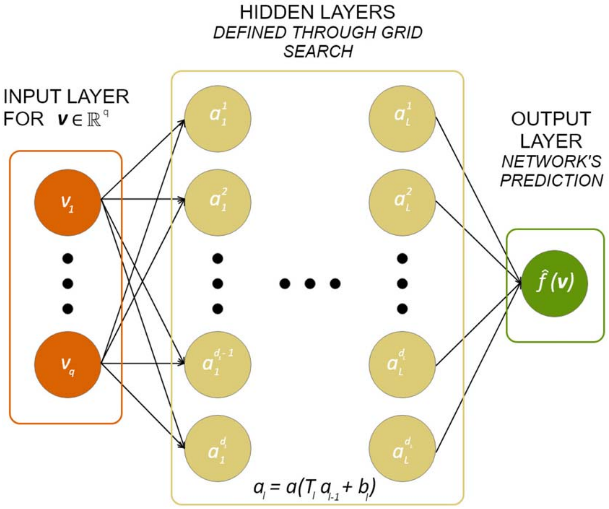

Fig. 1 Schematic of the feed-forward neural network employed

Table 1 Hyper-parameters tuned in grid search

Parameter	Lower limit	Upper limit
No. of HL	1	4
No. of neurons in each HL	5	500
Regularization coefficient	0.0	0.02
Drop-out factor	0.0	0.5
Activation function	Sigmoid	Relu

The CF for optimizing the weights and biases (training the network) is the mean squared error (MSE) between the network's predictions and real data for the whole of a training set composed of CFD samples. The optimization algorithm selected is RMSProp with a learning rate of 0.001. The number of training epochs is defined via an early stopping callback, monitoring the convergence of the MSE evaluated on a validation set comprising 20% of the samples, which are spared and not used for training. A maximum of 500 epochs without change in the MSE of the validation set is specified to prevent overfitting.

Thus, the network fitting process proposed requires two datasets: one for training the weights and biases and a second for tuning the network architecture. The former is additionally split to allow for a validation set and prevent overfitting. The proportions of the total number of samples being placed in each dataset is as follows: 64% for the weights and biases training, 20% for hyper-parameter tuning, and 16% for early stopping. The splitting of the dataset is performed using seeded random number generators, such that each network evaluated in the grid search is trained and tested using the same samples. It is worthy to note that the samples employed in this study correspond to a same model. While current AI research indicates NN capabilities that could enable the generalization of these networks to other models, these features have not been considered in this work.

The performance of the network, as determined by the MSE and R^2, was discovered to increase significantly when trained and tested on inputs and outputs that were normalized to have zero mean and a standard deviation of one, obtained through the transformation defined in Eq. (2) for a given vector $\mathbf{v}$.

$$\mathbf{v}_n = \frac{\mathbf{v} - \mathbb{E}[\mathbf{v}]}{\sigma(\mathbf{v})} \tag{2}$$

2.2 Coupling Active Design Subspaces.

Adequately fitting a neural network to high-dimensional input data might require a large number of samples, which can make the method infeasible in cases where function evaluations are obtained through a costly procedure. To enable the application of neural networks to high-fidelity CFD data, this work proposes coupling them with ADS.

The ADS approach identifies the linear combination of the input parameters that best describes the variability in an objective function, through the eigenvalue decomposition of the function's gradient covariance matrix, C, defined in Eq. (3), from Ref. [28].

$$C = \mathbb{E}[\nabla_x f \ \nabla_x f^T] \tag{3}$$

The eigenvalue decomposition of C, computed as $C = W \Lambda W^T$, yields the dominant directions in the columns of W. Based on the decay of the eigenvalues, W and Λ can be partitioned as per Eq. (4), such that the active subspace is captured in the matrix W_1, which maps the m-dimensional inputs, $\mathbf{x}$, to a k-dimensional *active* vector, $\mathbf{y}$, through the transformation expressed in Eq. (5).

$$\Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \quad W = [W_1 \ W_2], \quad W_1 \in \mathbb{R}^{m \times k} \tag{4}$$

In practice, the matrix C is approximated through a Monte Carlo method, by drawing M independent samples $\{\mathbf{x}_i\}$ according to the sampling density $\rho = \rho(\mathbf{x})$ in χ and computing the gradient for each sample, as defined in Eq. (6). Constantine [21] suggests adopting $M = \alpha k \log(m)$, where α is an oversampling factor and k is the maximum number of dimensions in the ADS that can be accurately resolved.

$$\mathbf{y} = W_1^T \mathbf{x} \tag{5}$$

$$C \approx \frac{1}{M} \sum_{i=1}^{M} \nabla_x f_i \ \nabla_x f_i^T \tag{6}$$

The application of ADS to the optimization problem stated in Eq. (1) requires building such a covariance matrix for the objective function and the n_c constraints. If n_c is sufficiently low and an adjoint code is available, then computing the gradients with adjoint can be a viable alternative, requiring M primal CFD computations and $M \times (n_c + 1)$ adjoint computations. However, for a general scenario, n_c might be overly large or adjoint codes might not be available. In such scenarios, an analytical model can be trained on sampled data and the gradients estimated from said model. The computational expense of building C in this case is dependent on the model employed. This work employs ANNs for this purpose.

Therefore, an ADS is sought to reduce the number of samples needed to adequately fit a neural network to high-dimensional input data, and a neural network is required to learn this ADS. This impasse is solved through the iterative process, shown schematically in Fig. 2. This process is defined based on the knowledge that, regardless of the number of input parameters, as an ANN is trained with increasing number of samples, provided there is no overfitting (for which the fitting procedure in Sec. 2.1 accounts), the prediction of said network will converge to the true function from which the samples were taken. Thereby, the estimation of C via gradients evaluated with the ANN also approximates the true matrix and its eigenvector decomposition converges to the same

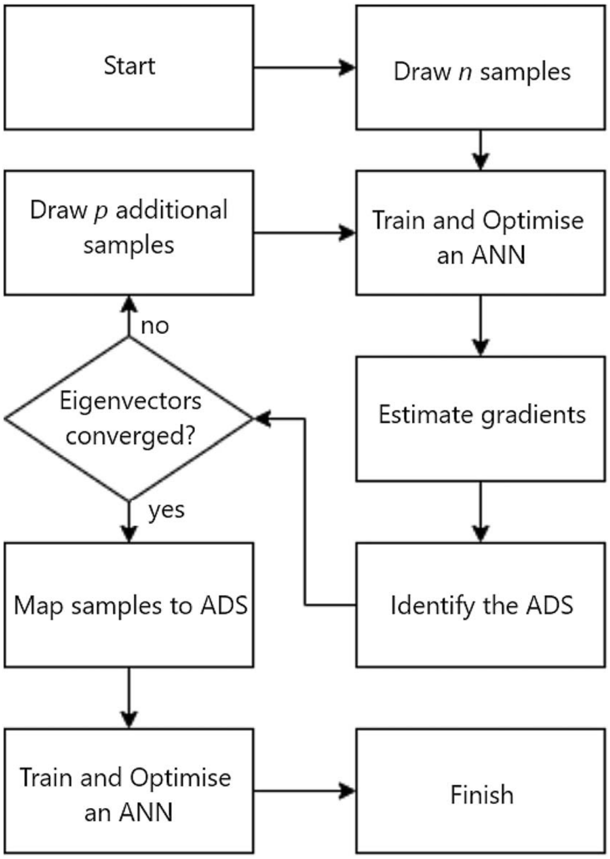

Fig. 2 Iterative process proposed to learn the ADS and fit an ANN to CFD samples

set of dominant directions. This work proposes iterating on the number of samples employed to fit the ANN while monitoring the convergence of the first k eigenvectors of C, through Eq. (7), which measures the angular variation, θ, between the estimated dominant directions from successive iterations.

$$\theta_i^{(j)} = \cos^{-1}\left(e_i^{(j)} \cdot e_i^{(j-1)}\right); \quad i = 1, \ldots, k \tag{7}$$

where $e_i^{(j)}$ is the ith eigenvector of C, estimated at the jth iteration.

Once θ is small enough, the ANN prediction approximates the function behavior in the dominant directions sufficiently well to reveal the ADS. This can therefore be exploited to reduce the dimensionality of the inputs. Mapping the samples to the active directions and fitting a final ANN to this low-dimensional dataset generally increases the accuracy of the network predictions since the data are now structured and there are more data points per dimension. The initial number of samples, n, and the increment, p, in the iterative process can be selected to exploit parallel computing capabilities, if available, to reduce the time requirements of the loop.

2.3 Rewriting the Optimization Problem. The transformation defined in Eq. (5) is called a *forward map* and, as described, transforms the input vector $\mathbf{x} \in \mathbb{R}^m$ to an active vector, $\mathbf{y} \in \mathbb{R}^k$. There is no reason why the active vectors of different functions should be the same; hence, after the fitting procedure, $n_c + 1$ ANNs are obtained, all responding to different active vectors in different subspaces. This section details a reformulation to the optimization problem described in Eq. (1) that allows to *navigate* through the active subspaces to find an optimum in the m-dimensional space, while exploiting the low-dimensional structure discovered through the ADS.

The forward map provides a unique vector $\mathbf{y}$ for each vector $\mathbf{x}$. However, the converse is not true. There are infinitely many $\mathbf{x}$ that satisfy the inverse map for a given $\mathbf{y}$. Let $\hat{f}$ be the neural network prediction of the CF for its active vector:

$$f(\mathbf{x}) \approx \hat{f}(W_{1_0}^T \mathbf{x}) \tag{8}$$

where W_{1_0} contains the first k eigenvectors of the CF's C-matrix. Additionally, let the constraints be approximated by a neural network built on their active vector:

$$g_i(\mathbf{x}) \approx \hat{g}_i(W_{1_i}^T \mathbf{x}); \quad i = 1, \ldots, n_c \tag{9}$$

A new function F is defined as

$$F(\mathbf{y_0}) = \hat{f}(W_{1_0}^T \mathbf{x}^*) \tag{10}$$

where

$$\begin{aligned} &\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \quad \frac{1}{2}\left\| W_{1_0}^T \mathbf{x} - \mathbf{y_0} \right\|_2^2 \\ &\text{subject to} \quad \hat{g}_i(W_{1_i}^T \mathbf{x}) \le h_i, i = 1, \ldots, n_c \\ &\mathbf{x} \in \chi \subset \mathbb{R}^m \end{aligned} \tag{11}$$

Each evaluation of $F(\mathbf{y_0})$ requires solving a constrained least-squares minimization problem designed to select from the infinitely many $\mathbf{x}$ that solve the inverse map, one that is feasible in terms of the n_c constraints. In case there are no feasible points that satisfy this, then the feasible $\mathbf{x}$ that maps to the *closest* active vector to $\mathbf{y_0}$ is selected. This is a convex minimization problem that can be easily solved by a gradient-based search algorithm, employing finite differences for the gradients of the constraints, $\nabla \hat{g}_i$, and Eq. (12) for the CF gradient. It is worthy to note that there might still be infinitely many $\mathbf{x}$ that solve Eq. (11). In such cases, all design points are considered equal since they map to the same active vector and only differ in their *inactive* directions. Hence, the CF should experience little variation between these points. The introduction of a regularization term (e.g., Tikhonov) to the CF in Eq. (11) is possible without loss of generality, in the case where there is preference for particular types of $\mathbf{x}$.

$$\nabla_{CF} = W_{1_0}(W_{1_0}^T \mathbf{x} - \mathbf{y_0}) \tag{12}$$

The constrained optimization problem defined in Eq. (1) can thus be rewritten as an unconstrained optimization through function F, as per Eq. (13).

$$\min_{y_0} \; F(\mathbf{y_0}) \tag{13}$$

This formulation enables employing the ANN built on the CF's ADS for global optimization. Exploiting the dominant directions usually accelerates the convergence of the optimization procedure since the function is very respondent to changes in its active variable. Moreover, selecting the feasible point through Eq. (11) allows employing the ANNs built for the constraints, which can be obtained cheaply, through the algorithm described in Fig. 2.

3 Comparison to an Adjoint Gradient-Based Approach

To compare the performance of the global strategy proposed, the optimization problem in Eq. (1) is also solved employing an adjoint-based technique. The search algorithm chosen is the sequential least-squares programing (SLSQP) from scipy [29]. The method wraps the SLSQP routine developed by Kaft [30]. It employs a Han–Powell quasi-Newton method with a Broyden–Fletcher–Goldfarb–Shanno (BFGS) update of the B-matrix for defining the search direction. This update requires gradient information of the CF and the constraints, hereby referred to as the *objectives*, which in this section is obtained through the adjoint approach. Next, SLSQP selects the optimum step size via a line search optimization, which only requires solving a primal to obtain the objectives' value at each iteration.

As discussed in Sec. 1, the adjoint computation requires a near-zero residual from the primal solver which increases the computational expense of that stage. Additionally, typical adjoint running times are about three times longer than the flow solver, making the B-matrix update a costly endeavor. This work aims to lessen the overall time expense by employing parallel computing capability and solving the adjoint computation for the objectives simultaneously.

For the step-length algorithm, there is no need for the primal to achieve such exemplary convergence and adequate results can be obtained much earlier. Figure 3 shows a typical CFD primal convergence history for an axial fan. By iteration 400, the objective convergence curves have stabilized and only minor variations in their value are noticed thereafter. From this point, in traditional CFD-based SLSQP implementations, the primal solver would be continued for an additional 1200 iterations to comply with the adjoint convergence requirements. This implies a fourfold increase

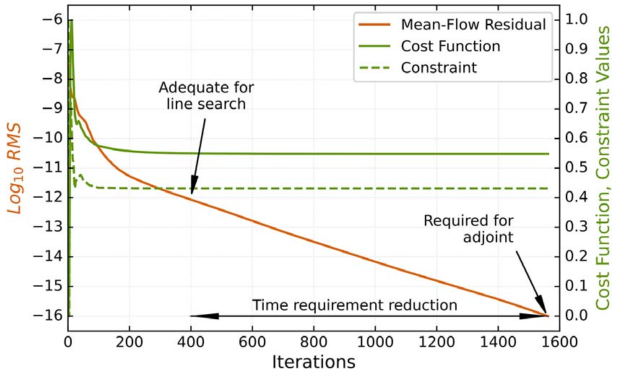

Fig. 3 Primal convergence history for mean-flow residual and non-dimensionalized CF and constraint values

in the computational expense for each line search iteration while there is no such requirement. Hence, this work proposes employing partially converged values to accelerate the step-size optimization of the algorithm.

4 Application to a Jet Engine Fan Blade

In this section, the optimization of a modern, low speed, high bypass ratio research fan blade is approached with the global strategy described in Sec. 2 and the adjoint-based strategy from Sec. 3. The test case under study is called Vital, hereby referred to as the *research blade*, shown in Fig. 4. As documented in Ref. [31], for high bypass ratio fans, a 1.4% increase in efficiency yields a 1.0% reduction in the engine's specific fuel consumption, making this component an ideal test case to study the benefits of global optimization on high-dimensional parametrization schemes. The span of the research blade is about two-thirds smaller than that of a conventional fan blade making it suitable for rig tests. The rotational speed has been adjusted to emulate the flow physics at the cruise condition and the thickness has been increased to achieve the mechanical integrity required of a large aero-engine composite fan blade.

4.1 Computational Tools. The Rolls-Royce proprietary CFD code Hydra [32] was used throughout this study to simulate the flow about the research blade. Hydra is an unstructured solver employing an edge-based data structure and convergence acceleration through an element collapsing multi-grid algorithm. A five-stage Runge–Kutta scheme with a block Jacobi preconditioner is employed for pseudo time-stepping when solving the steady-state Reynolds-averaged Navier–Stokes equations. The turbulence closure model employed in this work is Spalart–Almaras. Hydra's discrete adjoint capability was employed in this work to estimate the gradients of CF and constraints with respect to the design parameters.

The computational domain used in this study is shown in Fig. 4. It is a single-passage, single-blade row model with the downstream splitter. The whole domain is modeled on a rotating frame, with the casing, splitter, inlet and exit surfaces set as stationary. The rotor, hub, and splitter surfaces are set as viscous walls. At the inlet, a one-dimensional boundary condition (BC) is enforced, specifying a radial distribution of total pressure, total temperature, whirl and pitch angles, and turbulence intensity, where the values for these quantities are obtained from experimental analyses. For the bypass and core exit surfaces, a non-reflecting capacity exit BC is enforced.

The domain is discretized using the Rolls-Royce proprietary geometry and meshing software, PADRAM [33]. The blocking strategy employed by PADRAM consists of an H-O-H topology, with H blocks for the upstream and downstream regions, as well as the upper and lower periodic boundaries. The blade is enveloped in an O-mesh while a C-mesh is employed for the splitter. A mesh convergence study was previously undertaken to identify the optimal distribution of nodes [8], leading to a total of 5.4×10^6 cells, placing 30 mesh nodes in the tip gap. The $y+$ of the mesh is below 1 on all viscous surfaces. CFD-experimental validation of this setup have previously been reported achieving a good match for a number of different operating conditions [8,9].

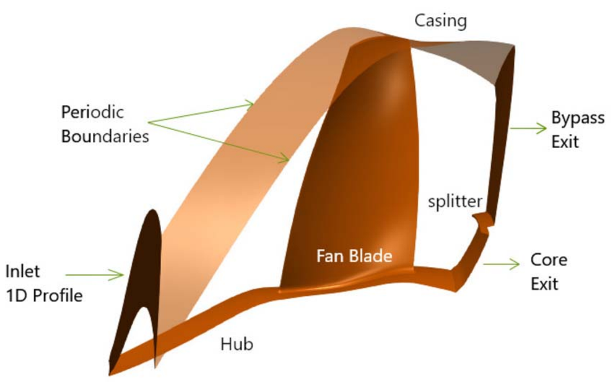

Fig. 4 Research blade CFD domain

4.2 Optimization Problem. The definition of the geometry parametrization is a critical factor dictating much of the success that can be obtained through optimization, since the search will only comprise geometries that are attainable with it. Many parametrization schemes have been introduced in the literature, such as free-form deformation or B-splining. However, most of these are abstract in the sense that the user has little knowledge on how a particular parameter affects the geometry, and indeed much less on how the interaction between various parameters affects the final shape. In this work, the geometry parametrization is defined through PADRAM's *engineering design parameters* (EDP), which comprise a set of intuitive geometry manipulation handles based on first principles. Each EDP, illustrated in Fig. 5 for an aerofoil section, controls a particular degree-of-freedom (DOF) for the geometry. The DOFs applied are *sweep* (axial movement of the section), *lean* (circumferential movement of the section), *skew* (rotation about the section's centroid), and leading edge (LE) and trailing edge (TE) *recambering*. Two additional DOFs controlling the locality of the LE and TE recambering are also introduced, such that low values of these parameters cause very localized camber line alterations, and vice-versa. Sufficiently large values can propagate the perturbations through the aerofoil, thus providing complete control over the camberline. The EDP are applied on five aerofoil *control sections* uniformly distributed through the blade span—at 0%, 25%, 50%, 75%, 100%—providing a total of 35 DOFs, arranged in the design vector, **x**. The value of the deformation applied as a function of the blade span is achieved through smooth cubic B-spline interpolation, with multiple control points via the control sections.

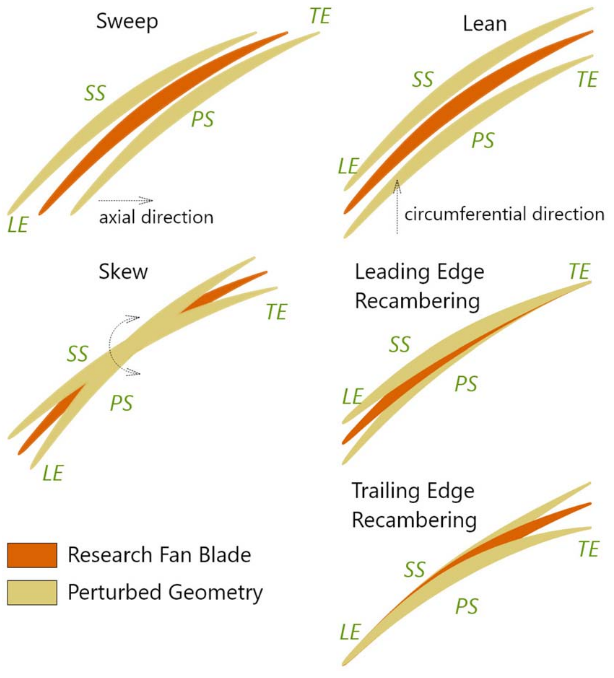

Fig. 5 Geometry parametrization through EDP. The magnitude of the perturbations has been enlarged for clarity.

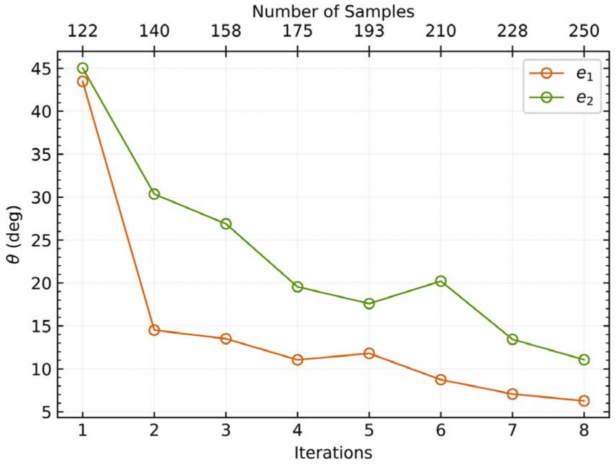

Fig. 6 Eigenvector convergence

The upper and lower optimization ranges for each DOF employed in this study, denoted by the vectors $\mathbf{x}^U$ and $\mathbf{x}^L$ respectively, were defined based on previous experience with the parametrization scheme. Let χ be the space of possible designs attainable with the 35 DOFs described; the design space is thus defined as follows:

$$\{\mathbf{x} \in \chi \mid \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U\} \tag{14}$$

The CF for the optimizations carried out in this work is the fan bypass isentropic efficiency, defined in Eq. (15), where the total pressure and total temperature quantities employed are extracted from the CFD solution by performing double averaging over the surface of interest: mix-out circumferentially and mass-mean radially.

$$CF = \eta(\mathbf{x}) = \frac{\left(\frac{p_{0_{\text{exit}}}}{p_{0_{\text{inlet}}}}\right)^{\frac{(\gamma-1)}{\gamma}} - 1}{\frac{T_{0_{\text{exit}}}}{T_{0_{\text{inlet}}}} - 1} \tag{15}$$

$$0.99\ PR_{\text{datum}} \leq PR(\mathbf{x}) \leq 1.05\ PR_{\text{datum}} \tag{16}$$

Through the optimization, upper and lower constraints are enforced for the fan PR according to Eq. (16). The lower bound is specified to prevent new geometries from maximizing efficiency by greatly reducing the PR, which would cause the low pressure (LP) shaft to operate at a higher speed to meet the engine's thrust requirement. Similarly, the upper bound is enforced to avoid designs with overly large PR that operate at significantly lower shaft speeds and can lead to overloading of the LP turbine blade. It is worthy to note that the EDP employed do not alter the thickness of the blade, thus preventing the search for an aerodynamically optimum design to lead to overly thin blades which would significantly affect the mechanical integrity of the blade.

4.3 Global Optimization Approach. In this section, the research blade is optimized employing the global strategy detailed in Sec. 2. The iterative process from Fig. 2 is solved for efficiency and PR, employing $n = 105$ initial function evaluations with an increment of $p = 17$. The sampling was done following a design of experiment approach employing Sobol' sequence. The convergence of the algorithm for the first two dominant directions is shown in Fig. 6. The trend demonstrates a decreasing magnitude of the angular variation between successive iterations, suggesting the C-matrix estimation via the neural network predictions is

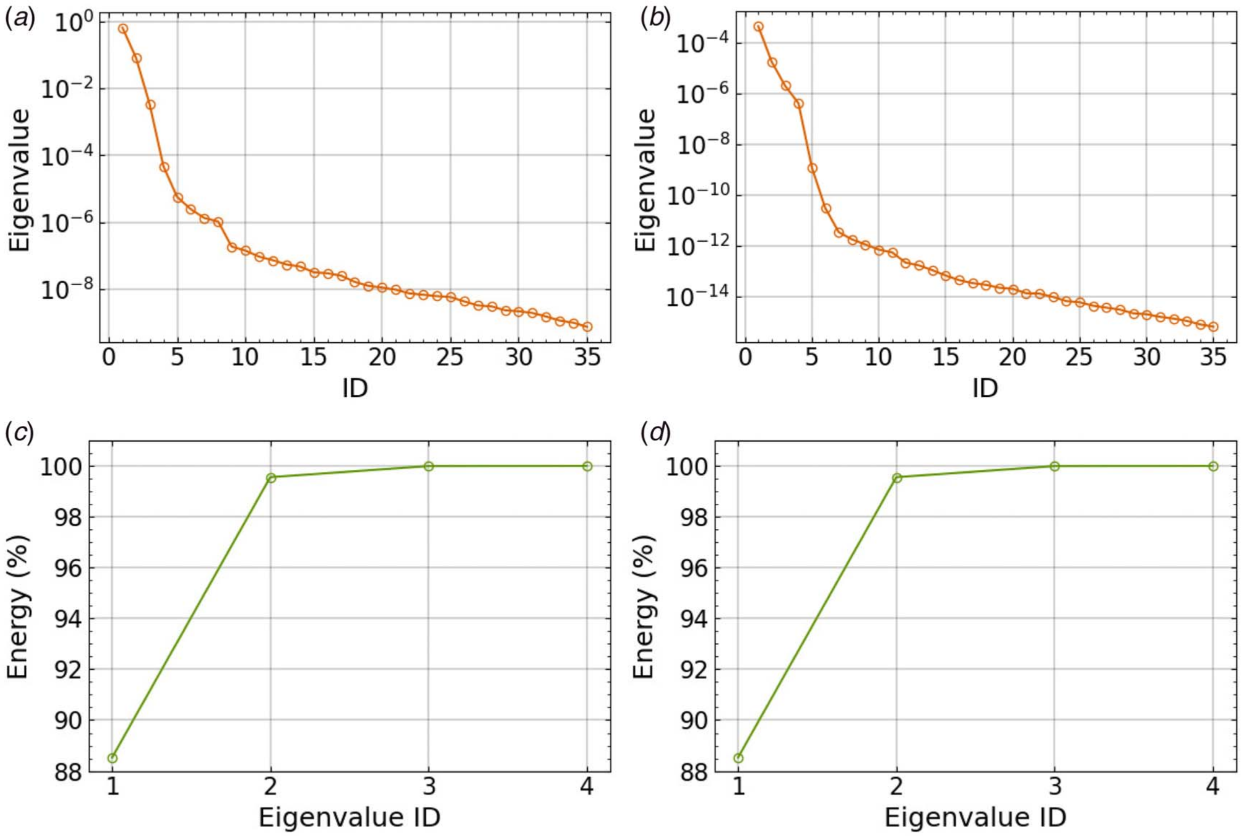

Fig. 7 Eigenvalue decay and cumulative energy plots for efficiency and PR gradient covariance matrices: (*a*) efficiency eigenvalues, (*b*) PR eigenvalues, (*c*) efficiency cumulative energy, and (*d*) PR cumulative energy

Table 2 Network prediction accuracy trained with high-dimensional inputs and active vectors

Function	$R^2[\hat{f}(\mathbf{x})]$	$R^2[\hat{f}(\mathbf{y})]$
Efficiency	0.892	0.937
PR	0.987	0.990

converging to its true form. By the eighth cycle, corresponding to 250 total samples, a sufficiently low angular variation has been achieved. Certainly, the process could be continued with increasing number of samples leading to an ameliorated resolution in the active directions; however, the trend shown in the figure suggests that only minor variations in the directions can be expected thereafter. Hence, the process is deemed converged at the eighth iteration.

Figures 7(*a*) and 7(*b*) plot the eigenvalue decay of the final efficiency and PR gradient covariance matrices. The rapid decrease noticeable in the eigenvalues is positive because it denotes that most of the system's variance is captured by just a few dominant directions. In fact, through the cumulative energy plots shown in Figs. 7(*c*) and 7(*d*), it is noticeable that almost 100% of the total system's energy can be represented by the first four eigenvalues. Thereby, for the optimization task at hand, the ADS was constructed employing the first four dominant directions.

Through Eq. (5), 250 samples were mapped to the PR and efficiency active subspaces, and a final neural network was trained and tuned for both functions. Table 2 gathers the prediction accuracy obtained with these networks and compares it against the ones trained on the high-dimensional space. A noticeable increase in prediction accuracy is achieved for both functions when training the networks with the low-dimensional space, due to the fact that the data are structured and the ratio of samples to dimensions is greatly increased. The ameliorated network performance provides an additional level of confidence in the subsequent surrogate-based optimization process, since the performance benefit of any optimized designs estimated from the networks should be close to what CFD would predict.

For visualization purposes, the data can be condensed even further. The cumulative energy plots show that over 98% of the energy is captured by just the first two eigenvalues and this allows building two-dimensional active subspace performance maps, following Ref. [22], without much loss of information. These maps, shown in Fig. 8, enable visualizing the behavior of the objective functions in their active subspaces. The active vector coordinates (0, 0) in these plots correspond to the datum design. The efficiency map shows that this design lies in a corridor of high-performance, but could still be improved quite significantly by moving in the second active variable's positive direction. However, such design changes could affect the PR beyond the specified tolerances. The active subspace on which the PR map is constructed is different than that for efficiency, preventing a direct combination of both plots. Through the reformulation of the optimization problem, discussed in Sec. 2.2, the search algorithm will be

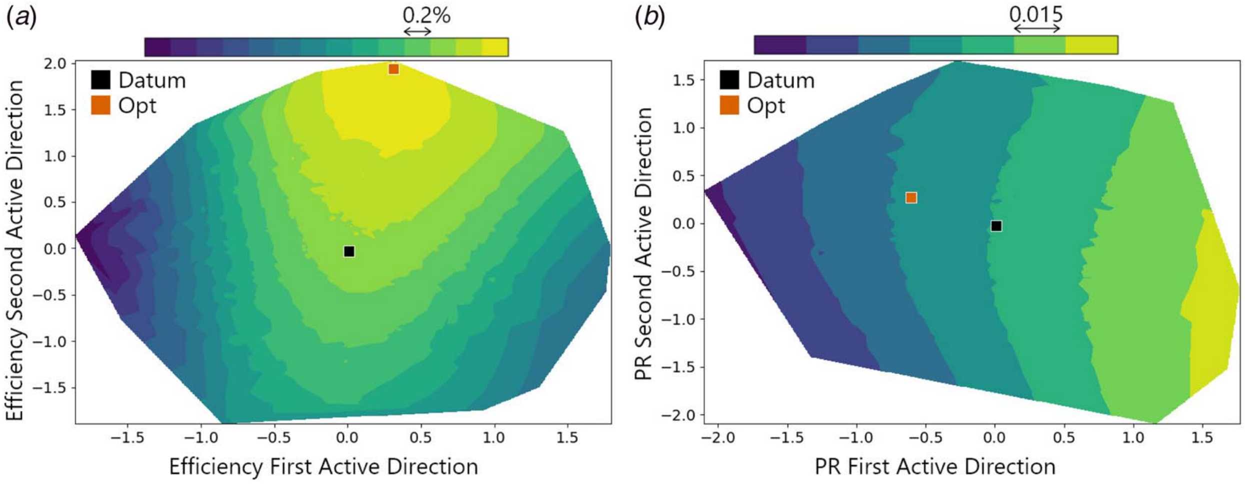

Fig. 8 2D active subspace maps for the optimization functions: (*a*) efficiency subspace and (*b*) PR subspace

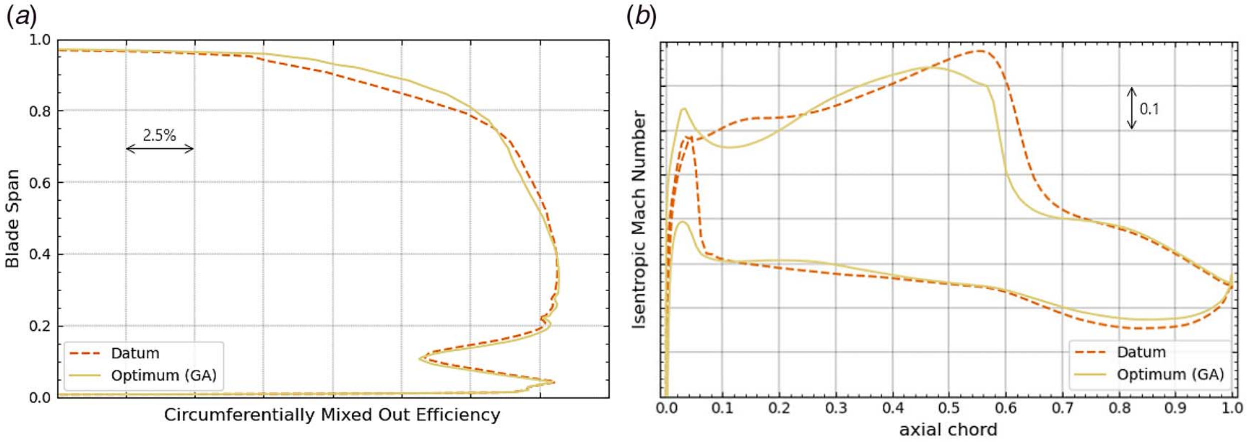

Fig. 9 Radial efficiency profile for datum and optimized designs in (*a*) and lift plot at 90% span in (*b*)

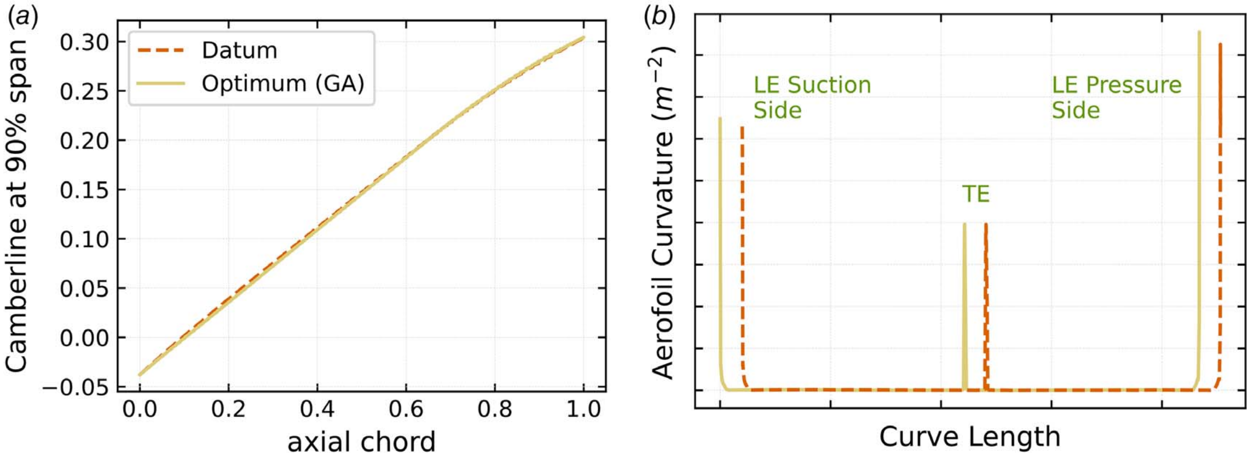

Fig. 10 Aerofoil geometry at 90% span. (*a*) Camberline distribution and (*b*) curvature versus arc length. The datum curve has been shifted for clarity.

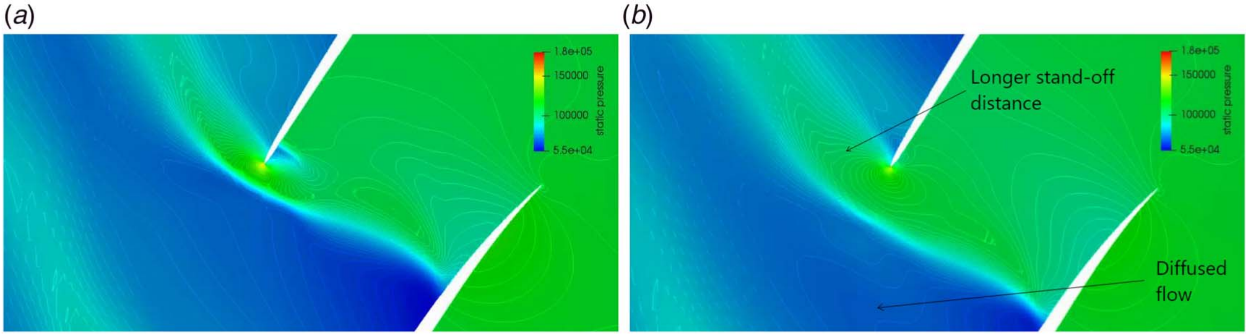

Fig. 11 Static pressure contours at 90% span. (*a*) Datum blade and (*b*) optimized design.

able to navigate through these subspaces searching for the best trade-off design.

In this work, a genetic algorithm was employed to perform the search for a globally optimum design using Eq. (13). Upon convergence, the optimal design point was simulated with CFD leading to a significant 0.47% improvement in efficiency, while the PR constraint was achieved.

Figure 9(*a*) plots the radial profile of circumferentially mixed-out values of efficiency (as per Eq. (15)) for the datum and optimized designs. This graph reveals that the performance increase for the optimum design arises largely from the upper 20% of the blade's span, while a slight reduction in efficiency is noticeable for the mid span region, due to the radial adjustment of the flow. The isentropic Mach number distribution for the 90% span section, plotted in Fig. 9(*b*), suggests that the primary mechanism for this performance benefit is an improved shock behavior via the reduction of the pre-shock Mach number. Additionally, the pressure side spike has been mitigated, leading to a more uniform loading at the LE. The suction side LE loading, however, has slightly been worsened for the optimum, noticeable by a minor peak followed by a sudden decay.

Geometrically, the ameliorated flow behavior is achieved through a slightly more convex camber line in the 20% to mid-chord region, as shown in Fig. 10(*a*). This causes a stronger pressure gradient that increases the supersonic diffusion of the flow, as shown in the contour plots of Fig. 11, reducing the pressure difference across the shock and thereby the entropy creation. In addition, the shock is spilled out of the passage, increasing the bow shock stand-off distance. This enables a pressure recovery in the region after the shock downstream to the LE. The effect is achieved in spite of the LE being sharper for the optimized design, as can be noted from the increased curvature shown in Fig. 10(*b*), which produces a smoother expansion, mitigating the Mach number spike at the pressure side.

4.4 Adjoint-Based Approach.

The efficiency improvement at each iteration of the optimization is shown in Fig. 12. It can be appreciated that within the first three iterations, the optimizer identified an optimal step size, which was employed throughout the optimization. This suggests that the function's Hessian matrix—estimated with each gradient evaluation—does not significantly change for the design points assessed during the process.

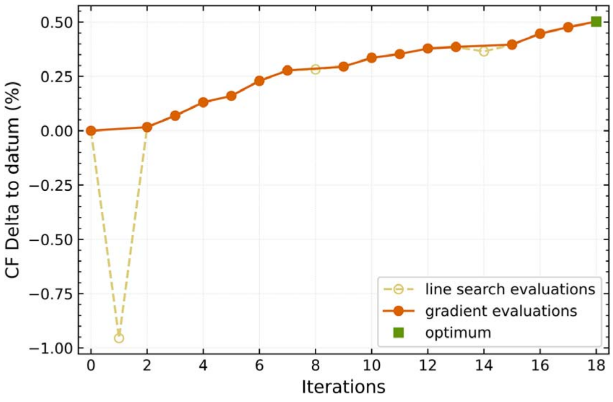

Fig. 12 Adjoint-based optimization history. The optimum found is highlighted by the square marker.

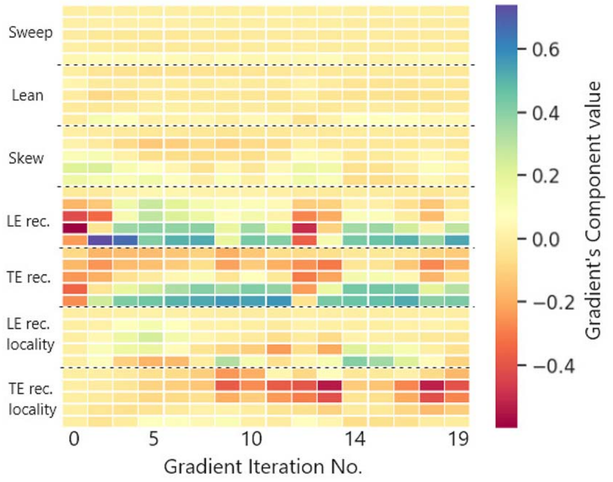

Fig. 13 Evolution of efficiency gradient through optimization. The gradient has been normalized by l_2 norm.

Thus, the SLSQP routine employed in this section is able to identify the dominant direction for each iteration.

Additional understanding of the process can be gained by analyzing the evolution of the gradients through the optimization. Each gradient evaluation provides information regarding the sensitivity of the CF with respect to each of the parameters at a particular iteration. A gradient component with a large magnitude reveals a parameter that is more influential than the others and dictates the primary geometrical modification of that iteration. This information is shown in Fig. 13, where each column encodes the efficiency gradient at a particular iteration. Throughout the process, LE and TE recambering were the most influential parameters, with negative gradient values near the root of the blade, and positive towards the tip.

Geometrically, this implies that the camberline curvature was reduced for the tip section, resulting in an s-shape which is more convex towards the pressure side, as shown in Fig. 14(*a*), with a similar outcome to the global optimum design from Sec. 4.3. This mitigates the expansion along the suction surface that delays and weakens the shock, as shown in the lift plot in Fig. 14(*b*). For the midspan regions, the aerodynamic outcome was similar, but the geometrical mechanism was different. As shown in Fig. 15(*a*), the blade inlet angle was increased, which causes a reduction in the effective flow incidence, thus reducing the loading at the LE. Additionally, the camberline concave curvature was also increased, as shown in Fig. 14(*c*), causing a smoother expansion which delayed the shock further downstream, reducing the shock-induced separated region. This ameliorated shock behavior enabled increasing the flow turning through a reduction in the exit angle (towards more negative angles), shown in Fig. 15(*b*), thus increasing the performance of the fan blade at these sections.

The gradient evolution in Fig. 13 reveals another interesting behavior of the optimization process. Primarily, that sweep, lean, and skew had little effect on the objectives. Since the research blade is already high-performing, it is likely that the geometrical features controlled by these parameters were already at an optimum setting. Additionally, it can be noted that, after the initial design cycles where predominantly the recambering parameters were modified, the search direction became affected by the parameters controlling the locality of the recambering. This suggests that the optimizer was trying to fine-tune the camberline

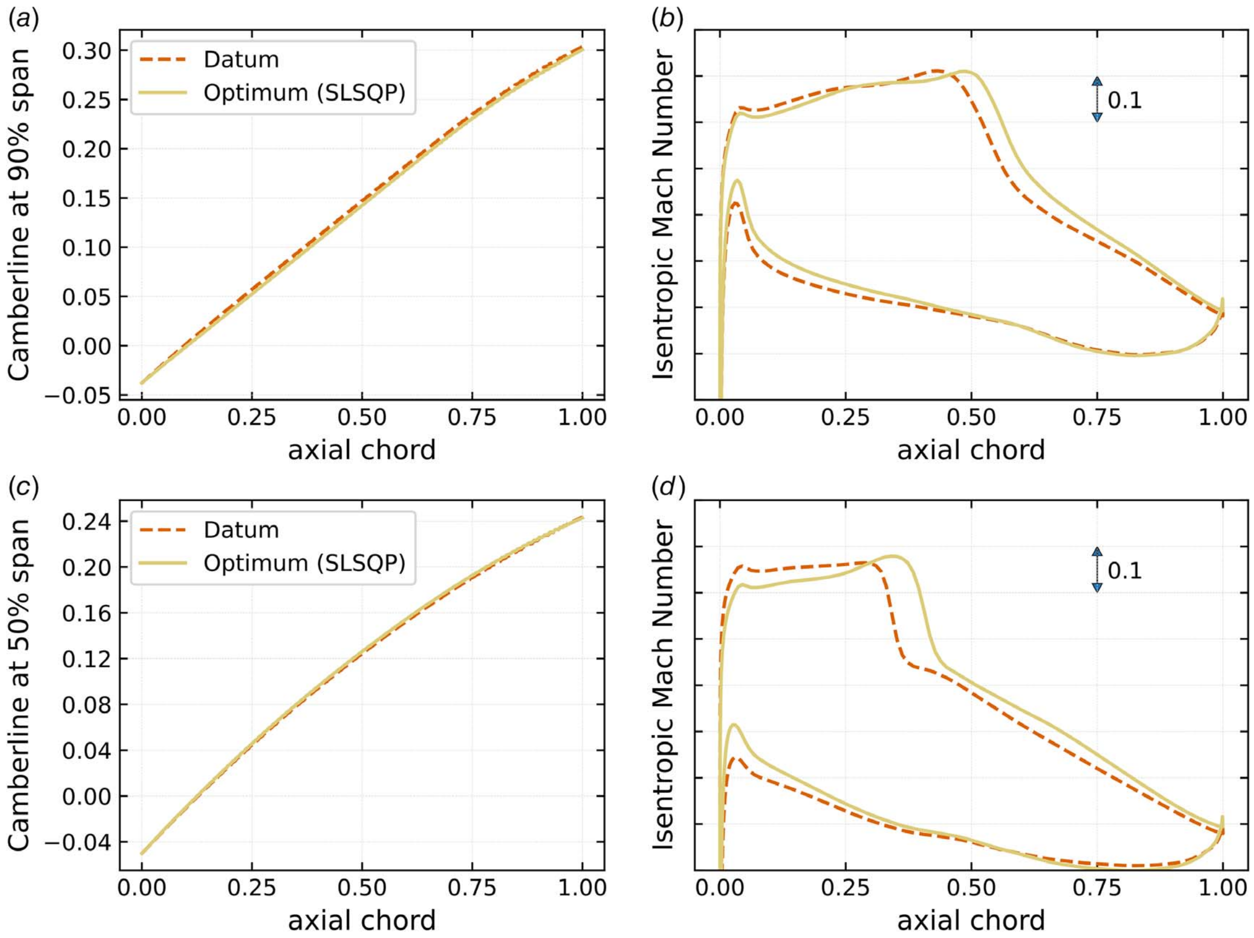

Fig. 14 Camberline distribution and lift plots of datum and optimum designs at (*a*) and (*b*) 90% span and (*c*) and (*d*) 50% span

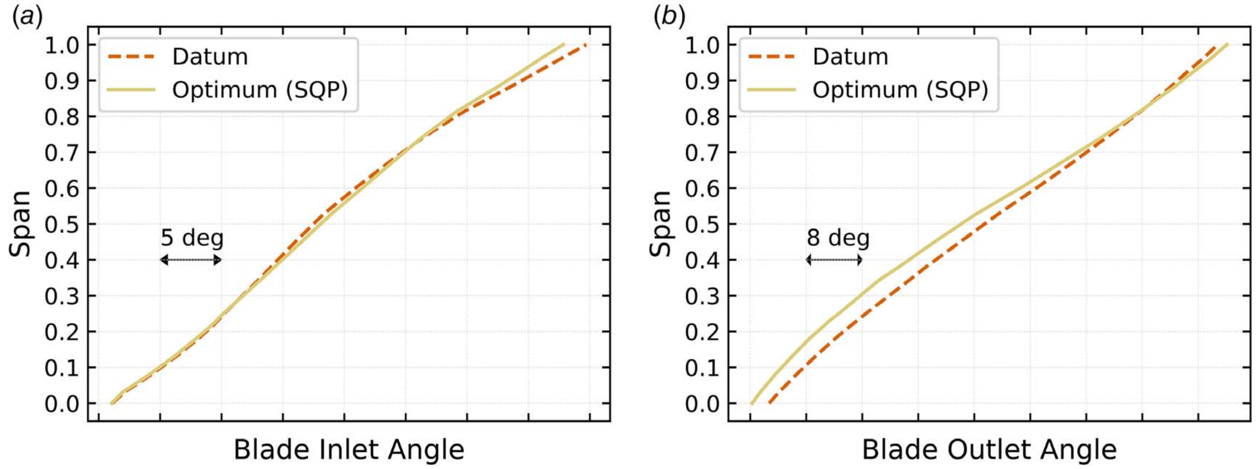

Fig. 15 Blade metal angle distribution: (*a*) inlet angle and (*b*) exit angle

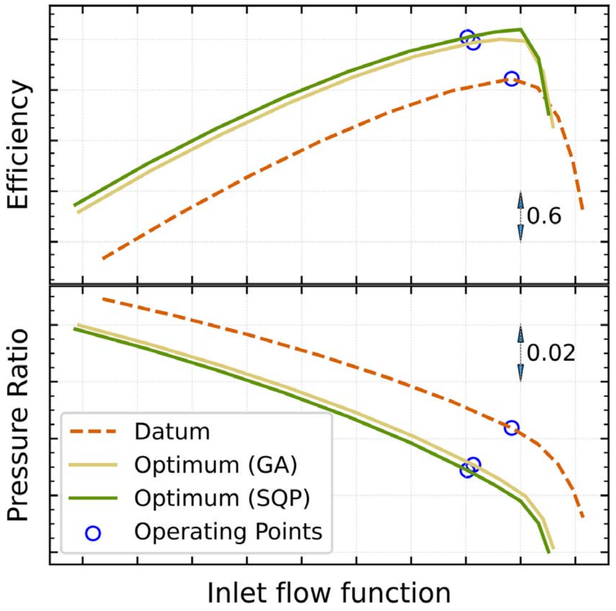

Fig. 16 Fan characteristic curves for optimized designs

distribution, resulting in the final shape previously described, which was found to be tailored at mitigating the shock.

This work was centered on the blade design point. However, off-design performance is an important consideration when assessing fan blade designs. This information is presented in Fig. 16 for the optimized and datum blades. The characteristic curves show that stall margin is maintained for both GA and SQP optimums, while the choke margin has been slightly reduced. Additional objective or constraint functions could be introduced in the optimization problem to maintain the datum off-design performance. Additionally, bulk-skewing the optimized blades could be considered to recover the choke margin [8].

4.5 Remarks on Computational Efficiency. Table 3 summarizes the computational expense and improvement achieved by each optimization method assessed in this study. The global optimization of the research fan blade, through the novel strategy proposed, required 250 CFD samples. The incremental number of samples employed for the iterative loop from Fig. 2 was adopted to exploit parallel computing capabilities such that each iteration ran all samples in batch and had the run time requirement of a single CFD computation. For generating the initial dataset, six of such batches were required. Thus, the time requirement for generating the complete dataset was approximately 14 times the running time of a single CFD. The time requirement of fitting and optimizing the neural networks is not considered since it is significantly lower than that of CFD runs.

Table 3 Summary of computational efficiency for the optimization approaches

	Global strategy	Adjoint-based
Total cost (CFD)	250	387
Total run time (CFD)	14	256
CF improvement	0.47%	0.5%
Constraint achieved	Yes	Yes

In contrast, the adjoint-based optimization required 19 iterations, of which 16 were B-matrix updates and the remaining 3 were for line search. As discussed, the update of the B-matrix requires a primal CFD with 4 times the cost of a normal run, and two adjoint calculations with 3 times the cost of the primal, hence, 12 times the cost of a normal run. This optimization amounts to a total computational expense of approximately 387 CFD runs. In terms of the time requirement, only the adjoint calculation for the CF and constraint can be parallelized, since the optimizer is sequential and decisions are made based on the previous solution. This leads to a total time requirement of approximately 256 CFD runs.

The behavior of the cost and constraint functions in the design space was such that both algorithms were able to converge to similar regions and produce optimized designs with comparable performance. The slightly lower benefit achieved with the global approach arises from the fact that, although highly accurate, the meta model built on the active subspace is a simplification of the true function shape. To overcome this, additional samples can be taken in the vicinity of the optimized design to refine the neural network in said region and improve the optimization accuracy.

The global optimization strategy achieved, with a reduced computational cost, an efficiency improvement comparable to a state-of-the-art adjoint-based approach. Additionally, the inherent implementation enables exploitation of parallel computing capabilities, which can significantly reduce the time requirement of the process. Moreover, the CF and constraint values can all be obtained with a single CFD run, thus providing better scalability of the global approach to multi-objective or multi-constrained optimization frameworks. This could enable the introduction of additional constraints designed to maintain the off-design performance, or the specification of an exit total pressure radial profile via multiple constraints.

5 Conclusion

A novel global optimization strategy has been developed that leverages the capabilities of ANNs for regressing complex functions while coupling them with ADS to reduce the number of samples required. This strategy was applied on the efficiency optimization of a modern jet engine fan blade with constrained PR and compared, both in terms of overall improvement and computational expense, to an adjoint-based approach employing the same parametrization. The global strategy achieved an efficiency increase comparable to the adjoint approach, with a reduced computational cost. In addition, it was demonstrated that adequate scalability to multi-objective or multi-constrained optimization applications is achieved.

Acknowledgment

The authors would like to thank Rolls-Royce for their support and permission to publish this work. The first author is funded by a Scholarship from the Italian Ministry of Education, through the University of Cagliari. This project has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 769025.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The authors attest that all data for this study are included in the paper.

Nomenclature

k = dimensionality of active vector
m = dimensionality of design vector
$\mathbf{x}$ = vector of design parameters
$\mathbf{y}$ = vector of design parameters in the active directions
C = gradient covariance matrix
W = eigenvectors of the covariance matrix
$\hat{f}$ = neural network approximation of function f
n_c = number of constraints in optimization problem
$\mathbb{E}[\cdot]$ = expected value of the argument in $[\cdot]$
Λ = eigenvalues of the covariance matrix
$\sigma(\cdot)$ = standard deviation of the argument in $(\cdot)$

References

[1] Horlock, J., and Denton, J., 2005, "A Review of Some Early Design Practice Using Computational Fluid Dynamics and a Current Perspective," ASME J. Turbomach., **127**(1), pp. 5–13.

[2] Jameson, A., 1988, "Aerodynamic Design Via Control Theory," J. Sci. Comput., **3**(3), pp. 233–260.

[3] Giles, M., and Pierce, N., 2000, "An Introduction to the Adjoint Approach to Design," Flow Turbul. Combust., **65**(3–4), pp. 393–415.

[4] Papadimitriou, D., and Giannakoglou, K., 2006, "Compressor Blade Optimization Using a Continuous Adjoint Formulation," Proceedings of ASME Turbo Expo 2006, Barcelona, Spain, May 8–11, pp. 1309–1317.

[5] Duta, M., Shahpar, S., and Giles, M., 2007, "Turbomachinery Design Optimization Using Automatic Differentiated Adjoint Code," Proceedings of ASME Turbo Expo 2007, Montreal, Canada, May 14–17, pp. 1435–1444.

[6] Wang, D., He, L., Li, Y., and Wells, R., 2010, "Adjoint Aerodynamic Design Optimization for Blades in Multi-Stage Turbomachines: Part II—Validation and Application," ASME J. Turbomach., **132**(2), p. 021012.

[7] Xu, S., Radford, D., Meyer, M., and Mueller, J.-D., 2015, "Cad-Based Adjoint Shape Optimisation of a One-Stage Turbine With Geometric Constraints," Proceedings of ASME Turbo Expo 2015, Montreal, Canada, June 15–19, p. V02CT45A006.

[8] Giugno, A., Shahpar, S., and Traverso, A., 2020, "Adjoint-Based Optimization of a Modern Jet Engine Fan Blade," Proceedings of ASME Turbo Expo 2020, Virtual, Online, Sept. 21–25.

[9] John, A., Ning, Q., and Shahpar, S., 2020, "The Influence of Parametrisation Setup on the Constrained Adjoint Optimisation of Transonic Fan Blade," Proceedings of ASME Turbo Expo 2020, Virtual, Online, Sept. 21–25.

[10] Connor, A., Clarkson, P., Shahpar, S., and Leonard, P., 2016, "Engineering Design Optimisation Using Tabu Search," Proceedings of Engineering Design Conference 2000, Uxbridge, UK, June, pp. 371–378.

[11] Ghisu, T., Parks, G. T., Jarret, J. P., and Clarkson, J., 2008, "Accelerating Design Optimisation Via Principal Components' Analysis," Proceedings of 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, Canada, Sept. 10–12, pp. 323–345.

[12] Oyama, A., Liou, M.-S., and Obayashi, S., 2004, "Transonic Axial-Flow Blade Optimization: Evolutionary Algorithms/Three-Dimensional Navier–Stokes Solver," J. Propul. Power, **20**(4), pp. 612–619.

[13] Chen, B., and Yuan, X., 2008, "Advanced Aerodynamic Optimization System for Turbomachinery," ASME J. Turbomach., **130**(2), p. 021005.

[14] Caloni, S., Shahpar, S., and Coull, J., 2016, "Numerical Investigations of Different Tip Designs for Shroudless Turbine Blades," J. Power Energy, **230**(7), pp. 709–720.

[15] Venturini, M., 2005, "Simulation of Compressor Transient Behavior Through Recurrent Neural Network Models," ASME J. Turbomach., **128**(3), pp. 444–454.

[16] Pierret, S., and Van den Braembussche, R., 1999, "Turbomachinery Blade Design Using a Navier–Stokes Solver and Artificial Neural Network," ASME J. Turbomach., **121**(2), pp. 326–332.

[17] Rai, M., and Madavan, N., 2000, "Aerodynamic Design Using Neural Networks," AIAA J., **38**(1), pp. 173–182.

[18] Kosowski, K., Tucki, K., and Kosowski, A., 2010, "Application of Artificial Neural Networks in Investigations of Steam Turbine Cascades," ASME J. Turbomach., **132**(1), p. 014501.

[19] Berguin, S., and Mavris, D., 2014, "Dimensionality Reduction in Aerodynamic Design Using Principal Component Analysis With Gradient Information," Proceedings of 10th AIAA Multidisciplinary Design Optimization Specialist Conference, National Harbor, MD, Jan. 13–17.

[20] Sobol', I., 2001, "Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates," Math. Comput. Simul., **55**(1–3), pp. 271–280.

[21] Constantine, P. G., 2015, *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, SIAM Spotlights, Philadelphia, PA.

[22] Sheshadri, P., Shahpar, S., Constantine, P., Parks, G., and Adams, M., 2017, "Turbomachinery Active Subspace Performance Maps," ASME J. Turbomach., **140**(4), p. 041003.

[23] Scillitoe, A., Ubald, B., Shahpar, S., and Seshadri, P., 2020, "Design Space Exploration of Stagnation Temperature Probes Via Dimension Reduction," Proceedings of ASME Turbo Expo 2020, Virtual, Online, Sept. 21–25.

[24] Cybenko, G., 1989, "Approximation by Superpositions of a Sigmoidal Function," Math. Control Signals Syst., **2**(4), pp. 303–314.

[25] Haykin, S., 1999, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Hoboken, NJ.

[26] Gulli, A., and Pal, S., 2017, *Deep Learning With Keras*, Packt Publishing Ltd., Birmingham, UK.

[27] Tikhonov, A. N., and Arsenin, V. Y., 1977, *Solution of Ill-Posed Problems*, Winston & Sons, Washington, DC.

[28] Lukaczyk, T., Palacios, F., Alonso, J., and Constantine, P., 2014, "Active Subspaces for Shape Optimization," Proceedings of 10th AIAA Multidisciplinary Design Optimization Specialist Conference, National Harbor, MD, Jan. 13–17.

[29] Oliphant, T. E., 2007, "Python for Scientific Computing," Comput. Sci. Eng., **9**(3), pp. 10–20.

[30] Kaft, D., 1998, "A Software Package for Sequential Quadratic Programming," Tech. Rep., DFVLR-FB 88-28, DLR German Aerospace Center—Institute for Flight Mechanics.

[31] Zamboni, G., and Xu, L., 2009, "Fan Root Aerodynamics for Large Bypass Gas Turbine Engines: Influence on the Engine Performance and 3D Design," ASME J. Turbomach., **134**(6), p. 061017.

[32] Lapworth, L., 2004, "Hydra-CFD: A Framework for Collaborative CFD Development," Proceedings of International Conference on Scientific & Engineering Computation, Singapore, June 30–July 2.

[33] Milli, A., and Shahpar, S., 2012, "PADRAM: Parametric Design and Rapid Meshing System for Complex Turbomachinery Configurations," Proceedings of ASME Turbo Expo 2012, Copenhagen, Denmark, June 11–15, Vol. 8, pp. 2135–2148.