

1 Polynomial Ridge Flowfield Estimation

2 A. Scillitoe,^{1, a)} P. Seshadri,^{1, b)} C. Y. Wong,² and A. Duncan³

3 ¹⁾*Data-Centric Engineering, The Alan Turing Institute.*

4 ²⁾*Department of Engineering, University of Cambridge.*

5 ³⁾*Department of Mathematics, Imperial College London.*

6 (Dated: 9 June 2021)

Computational fluid dynamics plays a key role in the design process across many industries. Recently, there has been increasing interest in data-driven methods, in order to exploit the large volume of data generated by such computations. This paper introduces the idea of using spatially correlated polynomial ridge functions for rapid flowfield estimation. Dimension reducing ridge functions are obtained for numerous points within training flowfields. The functions can then be used to predict flow variables for new, previously unseen, flowfields. Their dimension reducing nature alleviates the problems associated with visualising high dimensional datasets, enabling improved understanding of design spaces and potentially providing valuable physical insights.

The proposed framework is computationally efficient; consisting of either heavily parallelisable tasks, or linear algebra operations. To further reduce the computational cost, ridge functions can be computed at only a small number of subsampled locations. The flow physics encoded within covariance matrices obtained from the training flowfields can then be used to predict flow quantities, conditional upon those predicted by the ridge functions at the sampled points.

To demonstrate the utility of the framework, the incompressible flow around an ensemble of aerofoils is used as a test case. On unseen aerofoils the ridge functions' predictive accuracy is found to be reasonably competitive with a state-of-the-art convolutional neural network (CNN). The local ridge functions can also be reused to obtain surrogate models for integral quantities such a loss coefficient, which is advantageous in situations where long-term storage of the CFD data is problematic. Finally, use of the ridge framework with varying boundary conditions is demonstrated on a three dimensional transonic wing flow.

7 I. INTRODUCTION

8 The industrial penetration of Reynolds Averaged Navier-Stokes (RANS) has gone far beyond its
9 mainstay of aerospace engineering. Today it is being heavily used in automotive design¹, building
10 design², ventilation system design³, marine propeller design⁴ and more recently in the understand-
11 ing of oral transmissions^{5,6}. In using a turbulence model, RANS avoids the need to resolve perti-
12 nent length- and time-scales of turbulence; significantly slashing the cost of generating a flowfield.
13 That said, owing to the size and evermore realistic meshing topologies, these RANS computations
14 can themselves take several hours on modern clusters. Additionally, in both design and analysis
15 contexts, there is a need to assess multiple geometry and boundary conditions, adding to the com-
16 putational cost.

17 It is therefore unsurprising that many have sought to develop ways of (i) efficiently representing
18 a flowfield and (ii) approximating it for a new set of boundary conditions or geometry definition.
19 Falling under the remit of sparse reconstruction, there exists a variety of approaches for reconstruct-
20 ing full unsteady flowfield information from a limited subset of time-dependent measurements. A
21 suitable basis must first be chosen, onto which the sparse measurements are projected in order to
22 obtain a full reconstruction. Often the chosen basis is linear, and involves a direct computation of
23 the singular value decomposition. For example; Manohar *et al.*⁷ use proper orthogonal decompo-
24 sition to reconstruct full vorticity fields of laminar flow over a cylinder, while Discetti, Raiola, and

a)^{a)}Electronic mail: ascillitoe@turing.ac.uk; www.ascillitoe.com.

b)^{b)}Also at Department of Mathematics (Statistics Section), Imperial College London; www.psesh.com.

²⁵ Ianiro⁸ reconstruct turbulent velocity fields from sparse sensors. More recently, supervised machine
²⁶ learning methods have been used to provide a non-linear framework, with Erichson *et al.*⁹ using a
²⁷ shallow neural network to reconstruct the laminar flow over a cylinder.

²⁸ Meanwhile, in a range of fields, including uncertainty quantification, design optimisation, and
²⁹ sensitivity analysis, lower fidelity surrogate models (or emulators) are constructed from existing
³⁰ full-field simulation data. Deep neural network architectures, routinely used in data mining, have
³¹ been used with considerable success as a function approximation technique for high-dimensional
³² physics derived datasets^{10,11}. To increase accuracy, a network can be extended to arbitrary depth and
³³ width, however this leads to many weights and a very large amount of training data being required
³⁴ to alleviate overfitting. Recently, *convolutional neural networks* (CNN's) have been shown to be
³⁵ a promising approach to ameliorate this problem for CFD applications. CNN's use convolutional
³⁶ layers to take advantage of local spatial coherence in the input. These layers, combined with suc-
³⁷ cessive spatial resizing of the input data, can significantly reduce the number of weights compared
³⁸ to fully connected neural networks. Guo, Li, and Iorio¹² introduced the idea of using a CNN to
³⁹ learn a mapping between an object's geometric representation and the flowfield around it. Bhatna-
⁴⁰ gar *et al.*¹³ and Thuerey *et al.*¹⁴ recently built upon this work by introducing boundary conditions
⁴¹ as an additional input. Such approaches offer accurate and fast data-driven flowfield predictions,
⁴² allowing for near-immediate feedback for real-time design iterations. Following a different strat-
⁴³ egy, Tompson *et al.*¹⁵ replace the pressure projection step in a CFD solver with a CNN in order to
⁴⁴ accelerate simulations, whilst Sekar *et al.*¹⁶ use a CNN for the inverse design of an aerofoil, and
⁴⁵ Jin *et al.*¹⁷ use a CNN to estimate velocity fields from surface pressure distributions.

⁴⁶ Although applications of deep learning networks to real-world problems have become ubiquitous,
⁴⁷ our understanding of why they are so effective is lacking¹⁸. Additionally, techniques for interpreta-
⁴⁸ tion are available¹⁹, but they may not be particularly accessible for those without some specialised
⁴⁹ deep learning knowledge. Consequently, in many fields of engineering, there is still a demand
⁵⁰ for more readily interpretable surrogate models, even if these bring a slight trade-off in predictive
⁵¹ accuracy. One approach to easing the challenge of interpretation, is to reduce the dimensionality²⁰
⁵² of the problem. By reducing the dimensionality of the input space, data-driven ridge functions²⁰
⁵³ facilitate straightforward visualisation and understanding, whilst at the same time reducing data
⁵⁴ requirements. del Rosario, Constantine, and Iaccarino²¹ show how ridge functions can offer im-
⁵⁵ portant insights during the conceptual design phase, and in²², ridge functions aid exploration of a
⁵⁶ stagnation temperature probe's design space. Wong *et al.*²³ introduce the idea of using dimension
⁵⁷ reduction to estimate flow quantities, embedding polynomial ridge functions on an aerofoil surface
⁵⁸ to estimate the pressure distribution. In this paper, we develop spatially correlated ridge functions,
⁵⁹ with the target output being flowfield variables such as pressure, velocity or even turbulent variables.

⁶⁰ To demonstrate the utility of the proposed framework, we begin by exploring the prediction of
⁶¹ flowfields around a set of aerofoils. The design of efficient aerofoils is important since their use
⁶² is ubiquitous across many industries; for example aircraft wings and controls surfaces, car spoil-
⁶³ ers, and the gas turbine blades found in aircraft engines and power generation turbines. To provide
⁶⁴ a point of reference, we compare and contrast the proposed approach to a state-of-the-art flow-
⁶⁵ field prediction framework based on a convolutional neural network. In addition to examining the
⁶⁶ predictive accuracy, we explore how the dimension reducing nature of the learned model aids un-
⁶⁷ derstanding of the existing training dataset. To assess the limitations of the proposed approach, we
⁶⁸ then tackle the transonic flow over a three dimensional wing, with the inflow boundary conditions
⁶⁹ varied. We examine how the underlying polynomials perform when extrapolating away from the
⁷⁰ training data, and discuss approaches to handling this situation.

⁷¹ II. MATHEMATICAL FOUNDATIONS

⁷² In this section, we detail some of the mathematical ideas that underpin this paper.

⁷³ **A. Flowfield representation**

⁷⁴ It will be useful to represent a steady flowfield as a set of related scalar-field quantities. These
⁷⁵ include, but are not limited to, static pressure p , density ρ , a velocity component i.e. v_x , and even
⁷⁶ the turbulent viscosity v_t . These fluid properties are assumed known for any particular discretised
⁷⁷ location $\mathbf{s} \in \mathbb{R}^3$ within the three-dimensional flow domain \mathcal{S} . We consider \mathbf{s} to be synonymous with
⁷⁸ the number of nodes in the flowfield, where \mathbf{s}_1 corresponds to the first coordinate and \mathbf{s}_N corresponds
⁷⁹ to the last.

⁸⁰ Ignoring model parameters and limiters embedded within any modern flow-solver, one can group
⁸¹ the parameters influencing a flowfield into (i) those that govern the mesh and its associated geom-
⁸² etry, and (ii) those that set the boundary conditions. For simplicity, we condense the former into
⁸³ a vector of shape parameters $\mathbf{x} \in \mathbb{R}^d$ arising from an appropriately chosen design space \mathcal{X} , e.g.,
⁸⁴ Hicks-Henne bump functions, free-form deformation amplitude, or the inputs to an autoencoder
⁸⁵ that characterises the geometry. We define the latter as $\mathbf{v} \in \mathbb{R}^b$ where b represents the number of
⁸⁶ input boundary conditions, e.g., inlet stagnation pressure, exit static pressure, isothermal walls, etc.
⁸⁷ Then, following the statement at the start of this section, we can represent the steady flowfield as
⁸⁸ a set of related vectors

$$\mathcal{F} = \left\{ \begin{bmatrix} p(\mathbf{x}, \mathbf{v}; \mathbf{s}_1) \\ \vdots \\ p(\mathbf{x}, \mathbf{v}; \mathbf{s}_N) \end{bmatrix}, \begin{bmatrix} \rho(\mathbf{x}, \mathbf{v}; \mathbf{s}_1) \\ \vdots \\ \rho(\mathbf{x}, \mathbf{v}; \mathbf{s}_N) \end{bmatrix}, \begin{bmatrix} v_x(\mathbf{x}, \mathbf{v}; \mathbf{s}_1) \\ \vdots \\ v_x(\mathbf{x}, \mathbf{v}; \mathbf{s}_N) \end{bmatrix}, \dots, \begin{bmatrix} v_t(\mathbf{x}, \zeta; \mathbf{s}_1) \\ \vdots \\ v_t(\mathbf{x}, \zeta; \mathbf{s}_N) \end{bmatrix} \right\}. \quad (1)$$

⁸⁹ For generality, we refer to any scalar field quantity by f and thus with a slight abuse in notation,
⁹⁰ each of the vectors in \mathcal{F} is generalized via

$$\mathbf{y} = \begin{bmatrix} f(\mathbf{x}, \mathbf{v}; \mathbf{s}_1) \\ \vdots \\ f(\mathbf{x}, \mathbf{v}; \mathbf{s}_N) \end{bmatrix}. \quad (2)$$

⁹¹ Identifying a suitable surrogate model for \mathbf{f} is challenging because it effectively has $\mathbb{R}^d \times \mathbb{R}^b \times$
⁹² \mathbb{R}^N degrees of freedom, leading to an insuperable number of model evaluations. In this paper,
⁹³ we demonstrate how subspace-based dimension reduction approaches can be used to thwart the
⁹⁴ prohibitive cost associated with constructing such a surrogate.

⁹⁵ Without loss in generality, we consider the case where either the boundary conditions or the
⁹⁶ design parameters are precisely known, and thus any scalar field may be expressed purely as a
⁹⁷ function of \mathbf{x} and \mathbf{s}

$$\mathbf{f} = \begin{bmatrix} f(\mathbf{x}; \mathbf{s}_1) \\ \vdots \\ f(\mathbf{x}; \mathbf{s}_N) \end{bmatrix} \quad (3)$$

⁹⁸ or \mathbf{v} and \mathbf{s} . Note that each element in (3) is a scalar-valued function depending only on \mathbf{x} . In what
⁹⁹ follows, we present a class of techniques for approximating each element.

¹⁰⁰ **B. Ridge approximations**

Central to this paper is the notion that a scalar-valued high-dimensional function, such as the static pressure at a given computational node \mathbf{s}_i , can be approximated by a low-dimensional function defined over a sub-manifold

$$f(\mathbf{x}; \mathbf{s}_i) \approx p_i(\mathbf{W}_i^T \mathbf{x}; \mathbf{s}_i), \quad (4)$$

¹⁰¹ where $i = (1, \dots, N)$. Here $\mathbf{W}_i \in \mathbb{R}^{d \times n}$ is an orthonormal matrix where $n \ll d$, and p_i is a function
¹⁰² in \mathbb{R}^n . We define the approximation in (4) as a ridge approximation, an approximate form of what

¹⁰³ Pinkus²⁰ terms a generalised ridge function. The intuitive computational advantage behind ridge approximations is that instead of estimating a function in \mathbb{R}^d , we approximate it in \mathbb{R}^n . Numerous approaches are available in literature for identifying such structure using only data-driven techniques. ¹⁰⁶ These require input-output data pairs of the form $\{\mathbf{x}_m, f(\mathbf{x}_m; \mathbf{s}_i)\}_{m=1}^M$, where M represents a suitable number of design of experiments required to approximate the function in \mathbb{R}^n . Approaches rooted in classical regression include sliced inverse regression²⁴, principal Hessian directions²⁵, contour regression²⁶, sliced average variance estimation²⁷, and minimum average variance estimation²⁸. ¹¹⁰ More recent approaches include the works of Constantine *et al.*²⁹ and Hokanson and Constantine³⁰ that assume p_k is a polynomial, and explore optimisation techniques over the Grassmann manifold to identify both the coefficients of the polynomial and the dimension reducing subspace \mathbf{W} . Algorithms for identifying such ridge structure exist even for Gaussian process models (see³¹ and³²).

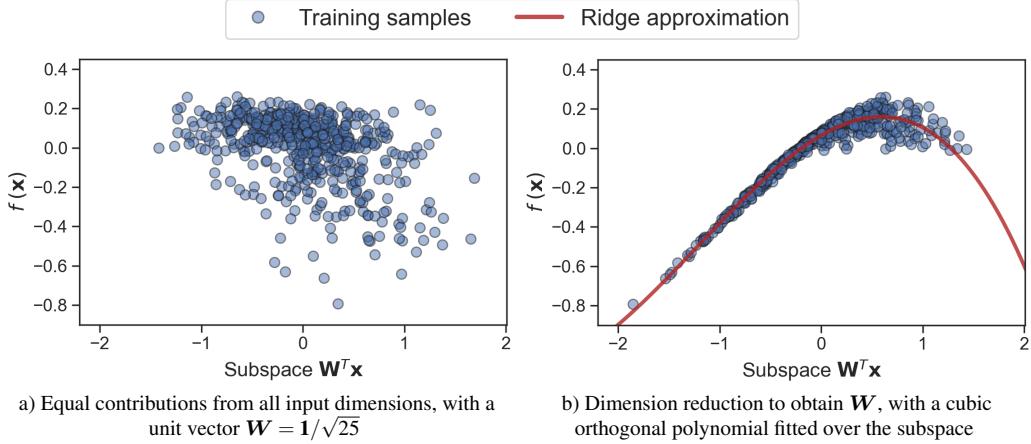


FIG. 1. Subspace-based dimension reduction applied to a turbo-machinery problem³³. The output scalar-valued quantity of interest $f(\mathbf{x})$, the efficiency of a turbo-machinery blade, is a function of the $d = 25$ design parameters in $\mathbf{x} \in \mathbb{R}^{25}$. The quantity f is projected over two different subspaces $\mathbf{W}^T \mathbf{x}$. In a), \mathbf{W} is chosen so that the subspace is formed from an equal contribution of all 25 parameters, whilst in b), a single linear combination of the 25 parameters which best describes the variation in f is obtained.

¹¹⁵ To provide an intuitive understanding of what these methods can offer, in Figure 1 we apply subspace-based dimension reduction to a dataset obtained from³³ and publicly available in ¹¹⁷ quadratures.org/datasets. When a suitable subspace \mathbf{W} is found, the M number of input- ¹¹⁸ output training data pairs $\{\mathbf{x}_m, f_m\}_{m=1}^M$ collapse onto a ridge over the subspace, where they can be ¹¹⁹ used to obtain a functional approximation (a polynomial in this case). Plots of function values over ¹²⁰ this subspace are called *sufficient summary plots*²⁹, and are useful in visualising the behaviour of ¹²¹ functions in high dimensions.

¹²² C. Polynomial ridge approximations

¹²³ Let us assume access to a training dataset $\{\mathbf{X}, \mathbf{F}\}$, where

$$\mathbf{X} = \begin{bmatrix} | & | \\ \mathbf{x}_1 & \dots & \mathbf{x}_M \\ | & | \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} - & \mathbf{f}_1^T & - \\ \vdots & & \\ - & \mathbf{f}_N^T & - \end{bmatrix} = \begin{bmatrix} | & | \\ \tilde{\mathbf{f}}_1 & \dots & \tilde{\mathbf{f}}_M \\ | & | \end{bmatrix} \quad (5)$$

¹²⁴ with $\mathbf{X} \in \mathbb{R}^{d \times M}$ and $\mathbf{F} \in \mathbb{R}^{N \times M}$. As before N represents the number of spatial nodes; d the dimension associated with the design space $\mathbf{x} \in \mathcal{X}$, and M the number of distinct flow-fields, each corresponding to a new design. At an isolated node s_i across all the flow-fields, we wish to identify

¹²⁷ a polynomial ridge approximation of the form

$$\underset{\mathbf{W}_i, \boldsymbol{\alpha}_i}{\text{minimise}} \|\mathbf{f}_i - \mathbf{P}(\mathbf{W}_i^T \mathbf{X}) \boldsymbol{\alpha}_i\|_2^2, \quad (6)$$

¹²⁸ where

$$\mathbf{P}(\mathbf{W}_i^T \mathbf{X}) = \begin{bmatrix} \phi_1(\mathbf{W}_i^T \mathbf{x}_1) & \dots & \phi_L(\mathbf{W}_i^T \mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ \phi_1(\mathbf{W}_i^T \mathbf{x}_M) & \dots & \phi_L(\mathbf{W}_i^T \mathbf{x}_M) \end{bmatrix} \quad (7)$$

¹²⁹ represents a Vandermonde-type matrix with n -variate polynomial basis terms $\{\phi_1, \dots, \phi_L\}$ with a
¹³⁰ cardinality of L , with unknown polynomial coefficients $\boldsymbol{\alpha}_i \in \mathbb{R}^L$. The cardinality here is defined as
¹³¹ the total number of basis terms, and will vary depending on the highest degree in along each dimen-
¹³² sion and the nature of the interactive polynomial terms between different dimensions. For further
¹³³ details on the construction of multivariate polynomials, we refer the interest reader to 1.1 in³⁴. The
¹³⁴ intention behind phrasing our objective as (6), is that for a new design \mathbf{x}^* , we can estimate the scalar
¹³⁵ field value at any node s_i using the computed polynomial ridge approximation. To ensure solutions
¹³⁶ to this problem are feasible, it will be useful to ensure that the polynomial matrix \mathbf{P} is well condi-
¹³⁷ tioned. This can be aided by adopting orthogonal polynomial basis terms—i.e., Legendre, Hermite
¹³⁸ or Jacobi—as these have demonstrably lower condition numbers than their monomial counterparts.
¹³⁹ In this paper, we utilise Legendre polynomials. This has the additional advantage of being able to
¹⁴⁰ rapidly yield moments and sensitivities (see³⁵).

The non-linear least squares problem in (6) can be simplified to an optimisation problem over the Grassmann manifold via the variable projection method³⁶

$$\underset{\mathbf{W}_i}{\text{minimise}} \|\mathbf{f}_i - \mathbf{P}(\mathbf{W}_i^T \mathbf{X}) \mathbf{P}^\dagger(\mathbf{W}_i^T \mathbf{X}) \mathbf{f}_i\|_2^2 \quad (8)$$

$$\Rightarrow \underset{\mathbf{W}_i}{\text{minimise}} \|(\mathbf{I} - \mathbf{P}(\mathbf{W}_i^T \mathbf{X}) \mathbf{P}^\dagger(\mathbf{W}_i^T \mathbf{X})) \mathbf{f}_i\|_2^2 \quad (9)$$

$$\Rightarrow \underset{\mathbf{W}_i}{\text{minimise}} \left\| \mathbf{O}_{\mathbf{P}(\mathbf{W}_i^T \mathbf{X})}^\perp \mathbf{f}_i \right\|_2^2 \quad (10)$$

¹⁴¹ where the superscript \dagger denotes the matrix pseudoinverse, and \mathbf{O}^\perp is the orthogonal projector onto
¹⁴² the complement of the column space of \mathbf{P} . Gradients for this objective can be readily computed
¹⁴³ and used in a Gauss-Newton algorithm over the Grassman manifold, as demonstrated in³⁰. The
¹⁴⁴ computation of the coefficients $\boldsymbol{\alpha}_i$ trivially follows. This enables us to compute the scalar field
¹⁴⁵ quantity at a given node i for a new design \mathbf{x}^* with $\mathbf{P}(\mathbf{W}_i^T \mathbf{x}^*) \boldsymbol{\alpha}_i$ and by extension the scalar field
¹⁴⁶ across all N nodes via

$$\begin{bmatrix} f(\mathbf{x}^*; \mathbf{s}_1) \\ \vdots \\ f(\mathbf{x}^*; \mathbf{s}_N) \end{bmatrix} \approx \begin{bmatrix} \mathbf{P}(\mathbf{W}_1^T \mathbf{x}^*; \mathbf{s}_1) \boldsymbol{\alpha}_1 \\ \vdots \\ \mathbf{P}(\mathbf{W}_N^T \mathbf{x}^*; \mathbf{s}_N) \boldsymbol{\alpha}_N \end{bmatrix}, \quad (11)$$

¹⁴⁷ which by construction constraints all subspaces $\mathbf{W}_1, \dots, \mathbf{W}_N$ to have the same dimension. While
¹⁴⁸ the calculation of polynomial ridge approximations across all N nodes is an embarrassingly parallel
¹⁴⁹ operation—as each the input-output data pairs at each node are treated independently—it will be
¹⁵⁰ useful to leverage the spatial correlations in a given scalar field to reduce the number of times (10)
¹⁵¹ has to be solved.

¹⁵² D. Spatial correlations

¹⁵³ Subspace-based dimension reduction approaches and the related active subspaces³⁷ have enjoyed
¹⁵⁴ tremendous computational success over the past few years—ushering in a shift from prior sparse
¹⁵⁵ grids and full-space design of experiment approaches. As a general observation, within computa-
¹⁵⁶ tional fluid dynamics applications, these methods have been primarily applied on integral quantities

of scalar field outputs, i.e., efficiency, lift, drag, etc. Wong *et al.*²³ recognised this, and posited that approximating the composite scalar fields of such integral quantities using dimension reducing subspaces may not only offer greater insight into significant flow features, but may even further reduce the number of simulations required to do so. One important idea born from their work is that the ridge approximations of nodes that are adjacent to each other in a flowfield are likely to be similar in the way they depend on input parameters. In locally smooth regions of the flowfield, nearby nodes are predominantly affected by a similar subset of inputs, resulting in similarity in their ridge subspaces. This notion is quantified via the subspace distance, and in Wong *et al.*²³ the authors demonstrate that this property enables efficient compression and recovery of flow fields. In what follows, we build upon this idea using a different formalism, leveraging some of our preliminary work in³⁸.

Using the data available in \mathbf{F} , we construct a sample covariance matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ of the form

$$\mathbf{K} = \frac{1}{M} \sum_{m=1}^M (\tilde{\mathbf{f}}_m - \mathbb{E}[\tilde{\mathbf{f}}_m]) (\tilde{\mathbf{f}}_m - \mathbb{E}[\tilde{\mathbf{f}}_m])^T, \quad (12)$$

where $\mathbb{E}[\tilde{\mathbf{f}}_m]$ denotes the mean of the m -th scalar field across all the N nodes. It is important to note that this covariance matrix captures the spatial correlations across all the nodes for a given scalar field quantity. Its diagonal represents the sample variances in the scalar field at each of the N nodes.

We partition the scalar field across all N nodes into two subsets: a smaller subset with J nodes ($\mathbf{s}_1, \dots, \mathbf{s}_J$) and what remains with $N - J$ nodes ($\mathbf{s}'_1, \dots, \mathbf{s}'_{N-J}$) with $J \ll N$. The spatial location of the nodes within each subset can be random. We assume that the scalar field for any design \mathbf{x}^* across all N nodes can be expressed as a sample from the multivariate normal distribution $\mathcal{N}(\mathbb{E}[\mathbf{f}], \mathbf{K})$, where the first argument represents the mean and the second the covariance. The covariance matrix in (12) can be partitioned into $(N - J) \times (N - J)$ and $J \times J$ blocks

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{bmatrix}, \quad \text{where } \mathbf{K}_{11} \in \mathbb{R}^{(N-J) \times (N-J)} \text{ and } \mathbf{K}_{22} \in \mathbb{R}^{J \times J}. \quad (13)$$

As the scalar fields arising from this partition will have Gaussian marginals, we can use the Schur complement to approximate the scalar field at $(N - J)$ nodes by computing the ridge approximations at ($\mathbf{s}_1, \dots, \mathbf{s}_J$) nodes only

$$\begin{bmatrix} f(\mathbf{x}^*; \mathbf{s}'_1) \\ \vdots \\ f(\mathbf{x}^*; \mathbf{s}'_{N-J}) \end{bmatrix} \approx \mathbf{K}_{12} \mathbf{K}_{22}^{-1} \begin{pmatrix} \mathbf{P}(\mathbf{W}_1^T \mathbf{x}^*; \tilde{\mathbf{s}}_1) \boldsymbol{\alpha}_1 \\ \vdots \\ \mathbf{P}(\mathbf{W}_J^T \mathbf{x}^*; \tilde{\mathbf{s}}_J) \boldsymbol{\alpha}_J \end{pmatrix}. \quad (14)$$

In a nutshell, this implies that we only need to solve (10) J times rather than that N .

III. COMPUTATIONAL SETUP

In this section we outline the two test cases examined in this paper, in addition to summarising the implementation of the ridge approximation framework and the CNN it will be compared to.

A. Test Cases

Within an engineering context, flowfield predictions around a given object are often required at a range of operating conditions. Whereas in other cases, flowfield predictions may be required at a given operating condition, whilst the object itself is altered. To examine both scenarios we use two test cases; 1) a two dimensional subsonic aerofoil flow, with a large number of different aerofoil designs; and 2) a three dimensional transonic wing flow, with the freestream conditions varied.

191 1. Exploring the design of a subsonic aerofoil

192 We start with the well known NACA0012 aerofoil, discretised with a 449x129 curvilinear C-
193 grid*, and deform it to obtain an ensemble of aerofoil designs. The aerofoil surface is deformed
194 using $d = 50$ Hicks-Henne bump functions⁴⁰

$$s_2(s_1) = s_{2,\text{base}}(s_1) + \sum_{j=1}^d \beta_j b_j(s_1), \quad (15)$$

195 where $s_{2,\text{base}}$ are the s_2 coordinates of the baseline aerofoil, b_j is the j^{th} bump function, and s_1 and
196 s_2 are normalised by the aerofoil's axial chord length C_1 . The bump amplitudes $[\beta_1, \dots, \beta_d]$ are then
197 stored within the input vector $\mathbf{x}_m \in \mathbb{R}^d$ for each m^{th} design.

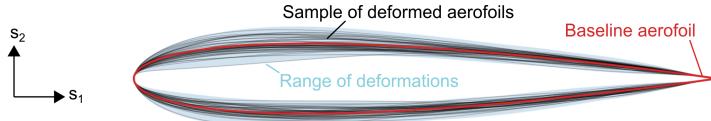


FIG. 2. Deformations made to the NACA0012 aerofoil. Fifty randomly selected deformed designs are shown.

198 To generate a dataset for training and testing, we create a ($M = 2000$)-point design of experiments
199 (DoE) with uniformly distributed Monte Carlo samples for \mathbf{x} . A random sample of the resulting
200 aerofoil designs, as well as the full range of deformations, is shown in Figure 2. Flowfields are sim-
201 ulated for each design using the incompressible solver of the SU2 CFD code⁴¹. The commonly used
202 Spalart–Allmaras RANS model⁴² is used to represent the effects of turbulence, with the freestream
203 turbulence viscosity ratio set to $(\nu_t / \nu)_\infty = 5$. The freestream velocity magnitude is set to $U_\infty = 1$
204 m/s, and the reference static pressure is set to zero. The laminar viscosity ν_∞ is then set to give
205 a Reynolds number of $Re = U_\infty C_1 / \nu_\infty = 6 \times 10^6$. Each design is run at three angles of incidence
206 $\alpha_\infty = 0^\circ, 10^\circ$ and 15° , leading to a dataset consisting of 6000 flowfields in total.

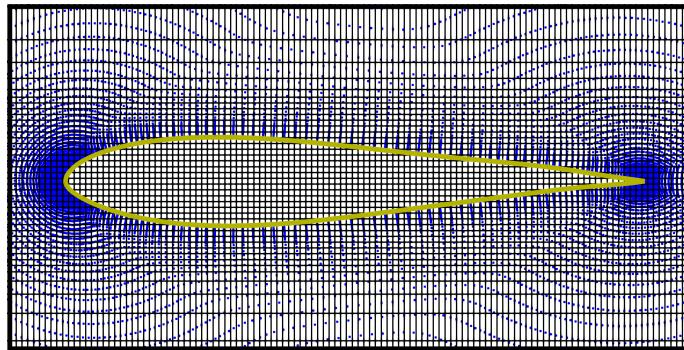


FIG. 3. Zoomed-in view of the discrete curvilinear C-grid representation of a deformed aerofoil (blue points), and the Cartesian grid (black) the flow variables are resampled onto. The aerofoil boundary is shown in yellow.

207 The grid deformations performed when perturbing the baseline aerofoil to reach each new design
208 mean that, even away from the aerofoil surface, grid points are at a slightly different location for
209 each design. Since we wish to learn a functional mapping for the flowfield variables at fixed points in
210 space, we re-sample[†] each flowfield onto the 90x318 Cartesian grid shown in Figure 3, and remove
211 points lying inside the solid region of the aerofoil. We choose a Cartesian grid here since we are

*This grid is used as a verification case by the AIAA Fluid Dynamics Technical Committee Turbulence Model Benchmarking Working Group (TMBWG) and is available from turbmodels.larc.nasa.gov/naca0012_grids.html. Diskin *et al.*³⁹ show lift and drag coefficients to be sufficiently grid independent at the 449x129 grid resolution.

[†]Resampling is performed with the `pyvista` python library⁴³, which uses linear interpolation for resampling.

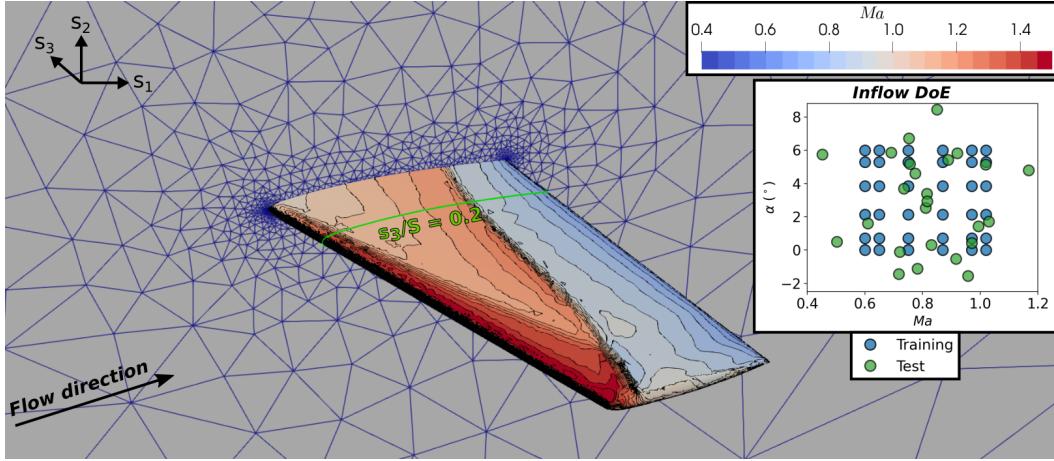


FIG. 4. Mach number contours on the suction surface of the ONERA M6 transonic wing, under the baseline freestream conditions of $Ma_\infty = 0.8395$ and $\alpha_\infty = 3.06^\circ$. The unstructured grid is shown on the $s_3 = 0$ symmetry plane, and the $s_3/S = 0.2$ slice is highlighted. The design of experiments (DoE) used to generate the different freestream conditions for training and test data is also shown in the inset figure.

interested in full flowfield visualisations, but the approach is equally applicable to individual points, surfaces (see Ref.²³) or planes.

After the above procedure we are left with $N = 90 \times 318 = 28.6 \times 10^3$ output values $f_{i,m}$ for each input \mathbf{x}_m , forming M input/output pairs $(\mathbf{x}_m, \mathbf{f}_m)_{m=1}^M$, for each of the three angles of incidence investigated. The scalar $f_{i,m}$ is the field variable at the i^{th} grid point for the m^{th} design, where we take the static pressure p , velocities u, v , and turbulent viscosity v_t as the field variables to predict. We normalise the aforementioned variables by taking the static pressure coefficient $C_p = (p - p_\infty)/(p_0 - p_\infty)$, axial velocity ratios $u/U_\infty, v/U_\infty$, and turbulent viscosity ratio v_t/v_∞ . The $M = 2000$ designs are further split up into $M_{train} = 500$ training designs and $M_{test} = 1500$ testing designs.

2. Varying freestream conditions for a transonic wing

To explore predictions with varying operating conditions, we take the compressible inviscid flow around the well known ONERA M6 transonic wing test case⁴⁴. The salient details of this case are presented in Figure 4. It is discretised by a $N = 108.4 \times 10^3$ point unstructured grid, with no boundary layer grid necessary due to the inviscid nature of the flow. Freestream boundaries are placed 10.5 root chord length's away from the wing, with the freestream Mach number and angle of incidence set at $Ma_\infty = 0.8395$ and $\alpha_\infty = 3.06^\circ$ for the baseline case. A span-wise symmetry boundary is then placed at $s_3 = 0$. The flows are simulated using the compressible inviscid solver of the SU2 CFD code⁴¹, and in Figure 4 the resulting Mach contours are shown on the surface of the wing for the baseline case. The flowfield here bears a close resemblance to the real viscous flow, with a characteristic “lambda shock” pattern visible on the upper surface of the wing.

Shown in the inset figure in Figure 4 is the design of experiments used to explore the freestream conditions. For the training data, operating points for Ma_∞ and α_∞ are obtained by applying six-point Gauss-Lobatto quadrature rules over the intervals $Ma_\infty \in [0.6, 1.02]$ and $\alpha_\infty \in [0, 6]$, leading to a total of $M_{train} = 36$ training flows. For testing, $M_{test} = 25$ points are sampled from the multivariate normal distribution $\mathcal{N}(\mu, \sigma)$, with $\mu = [0.8, 3]^T$ and $\sigma = [0.14, 3]^T$. Unlike in test case 1 the geometry is fixed here, therefore there is no need to resample the flowfields onto a base grid. For the scalar field variable $f_{i,m}$ we consider the local Mach number Ma .

²⁴⁰ **B. Polynomial ridge function implementation**

²⁴¹ To obtain dimension reducing ridges, we use the *equareatures*⁴⁵ python library which contains
²⁴² an implementation of the polynomial variable projection method described in Section II C. Since
²⁴³ each ridge function is independent of one another, we parallelise the task in an embarrassingly par-
²⁴⁴ allel fashion, with computation of individual ridge approximations performed by individual Python
²⁴⁵ worker processes. The code is parallelised using the *joblib* library with the *loky* backend. Each
²⁴⁶ parallel Python process is run concurrently on a separate computational core on a F72s_v2 virtual
²⁴⁷ machine (36 physical CPU's, 144 GiB memory) on the Microsoft Azure cloud computing service.
²⁴⁸ For the $N = 28.6 \times 10^3$ points in test case 1, with 500 training designs, training takes approximately
²⁴⁹ 20 minutes per field variable (≈ 24 ridge approximations per second). Note that training times
²⁵⁰ can be substantially reduced by taking advantage of spatial correlation in the manner described in
²⁵¹ Section IV E.

²⁵² The flowfield reconstruction procedure described in Section IV E is entirely made up of linear
²⁵³ algebra operations, thus it is straightforward to implement efficiently with existing linear algebra
²⁵⁴ libraries. For this we use the *numpy*⁴⁶ python library with Intel® MKL. To limit computation and
²⁵⁵ storage requirements, we compute only the K_{12} and K_{22} blocks of the covariance matrix in (13),
²⁵⁶ since only these blocks are required by the Schur complement step in (14). If further cost savings
²⁵⁷ are required, low-rank approximations for K can be obtained, as is done in³⁸.

²⁵⁸ **C. Convolutional neural network implementation**

Thuerey *et al.*¹⁴ explored a number of CNN architectures for flowfield predictions, and found the U-Net architecture⁴⁷ to be the most successful. In this paper we implement a modified version of the framework proposed by Thuerey *et al.*¹⁴, shown in Figure 5. The U-Net architecture consists of an encoder, which progressively down-samples the $128 \times 128 \times 4$ input data with strided convolutions. The four input channels consist of a boolean mask to define the aerofoil geometry, and three uniform input channels defining the freestream conditions Re , α_∞ , and $(v_t/v)_\infty$. This allows the network to extract increasingly large-scale and abstract information as the number of feature channels grows, until we are left with a $1 \times 1 \times 256$ derived feature vector. The decoder then does the opposite, with depooling layers reducing the number of features while increasing the spatial resolution. Eventually we are left with four 128×128 output channels, consisting of flowfield estimates for C_p , u/U_∞ , v/U_∞ and v_t/v_∞ . Skip connections help the network to consider low-level input information during the reconstruction of the solution in the decoding layers. The weights $w_{ij}^{(l)}$ (and biases $b_i^{(l)}$) of the neural network are obtained by minimising the loss function

$$\begin{aligned} \mathcal{L} = & \frac{1}{MN} \sum_{m=1}^M \sum_{i=1}^N \phi_{hub} \left(\hat{C}_{p,i,m} - C_{p,i,m} \right) + \phi_{hub} \left(\frac{\hat{u}_{i,m}}{U_\infty} - \frac{u_{i,m}}{U_\infty} \right) \\ & + \phi_{hub} \left(\frac{\hat{v}_{i,m}}{U_\infty} - \frac{v_{i,m}}{U_\infty} \right) + \phi_{hub} \left(\frac{\hat{v}_{t,i,m}}{v_\infty} - \frac{v_{t,i,m}}{v_\infty} \right), \end{aligned} \quad (16)$$

²⁵⁹ with $\hat{\cdot}$ denoting the CNN prediction for the field variable, and ϕ_{hub} denoting the Huber loss
²⁶⁰ function⁴⁸. As seen in (16), the error is averaged over all M number of training designs and N
²⁶¹ number of grid points. This loss function is differentiable, enabling it to be back-propagated into
²⁶² the network in order to compute the weight gradient $\nabla_w \mathcal{L}$, which is important for the optimisation
²⁶³ (*learning*) process.

²⁶⁴ The complexity of the above CNN is altered in this paper by varying the number of channels
²⁶⁵ (and therefore the number of parameters), with the 488k parameter CNN in Figure 5 found to offer
²⁶⁶ the best compromise between training times and accuracy. All CNN's detailed in this paper are
²⁶⁷ implemented in the PyTorch python library, and trained on a GTX970 1664 core NVIDIA GPU.
²⁶⁸ Training times range from 50 seconds to 80 minutes, depending on the complexity of the network
²⁶⁹ and the number of training designs. Further details are included in Appendix A.

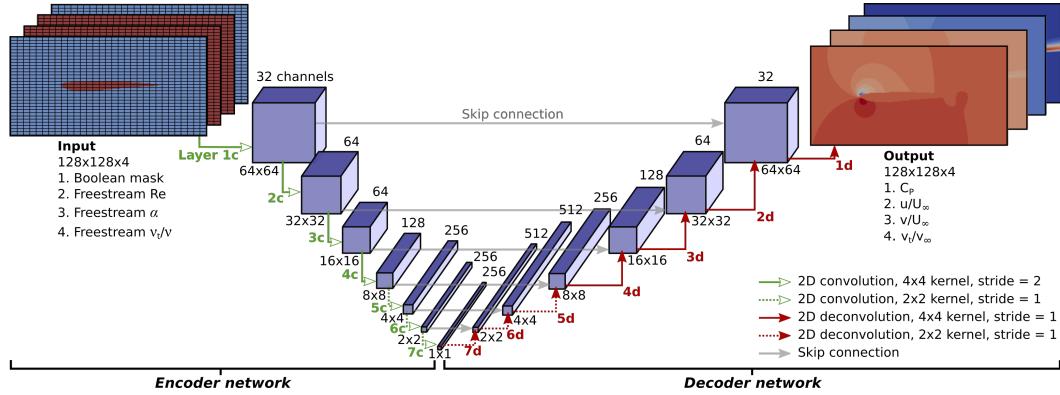


FIG. 5. The 488k parameter U-net convolutional neural network architecture used for flowfield predictions. Arrows indicate the direction of forward operations. The network is fully convolutional with 14 layers. Further details of the architecture are given in Appendix A.

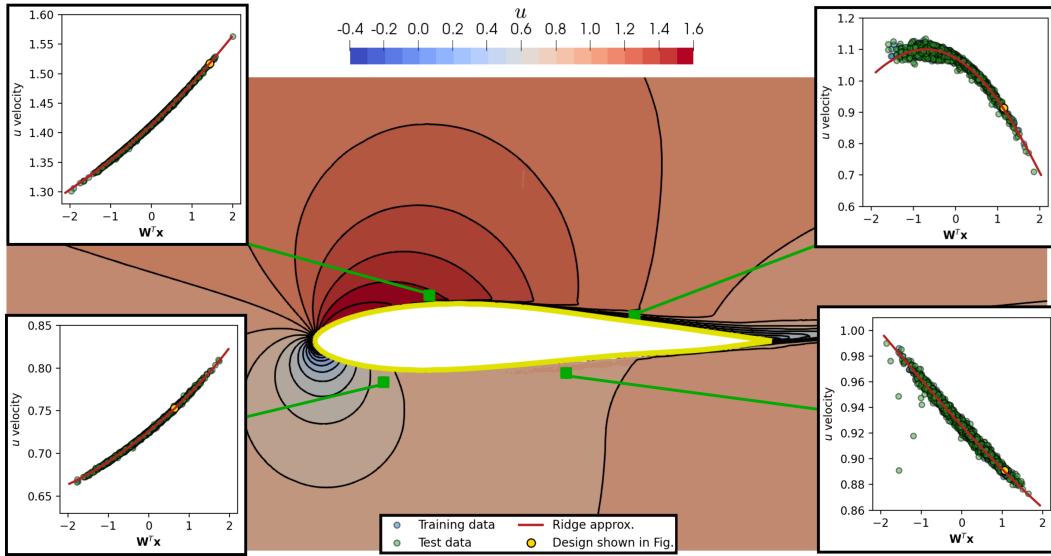


FIG. 6. Normalised axial velocity, u/U_∞ , for a deformed aerofoil (from the test set) at an angle of incidence of $\alpha_\infty = 10^\circ$. Colour contours show predictions from the second degree polynomial ridge approximations, isolines show the true CFD solution. Sufficient summary plots at four locations are shown.

270 IV. TEST CASE 1: DESIGN OF A SUBSONIC AEROFOIL

271 As an initial demonstration, ridge approximations are obtained for the static pressure coefficient
 272 $C_p = (p_0 - p)/(p_0 - p_\infty)$, normalised axial velocity u/U_∞ , and turbulent viscosity ratio v_t/v ,
 273 with 500 aerofoil designs used for training. Flowfield predictions for a new aerofoil design can then
 274 be obtained by transforming the new design vector \mathbf{x} to the reduced dimensional space, and evalua-
 275 ting the polynomial/ridge approximation $\mathbf{P}(\mathbf{W}_i^T \mathbf{x}^*) \alpha_i$ at each point $i = 1, \dots, N$ in the flowfield.
 276 In Figure 6, the predicted normalised u velocity is shown for a randomly selected design from the
 277 test set. Qualitatively, the predictions are in close agreement to the *true* velocity field from the CFD
 278 solution, shown by the black iso-lines. The design shown here was not used during training, hence
 279 the ridge approximations appear to be able to *learn* enough of the flow physics to be able to make
 280 predictions for the new design.

281 In addition to making flowfield predictions, a second key aspect of the ridge approximations are
 282 *sufficient summary plots*, seen in the insets within Figure 6. Applied to aerofoil design, they provide

283 visualisation of how the quantity of interest, in this case the u velocity at selected locations, is
 284 altered as the shape is changed. This is made possible since the sufficient summary plot shows the
 285 u velocity over the reduced dimensional space $\mathbf{W}^T \mathbf{x} \in \mathbb{R}$, instead of the full space $\mathbf{x} \in \mathbb{R}^{50}$. The low
 286 scatter of train and test points in the summary plots highlights that low dimensional structure has
 287 been successfully found.

288 A. Accuracy of Ridge Approximations

289 To explore how the accuracy of the proposed ridge approximation framework compares to a state-
 290 of-the-art convolutional neural network (CNN), ridge approximations and the 488k parameter U-Net
 291 in Figure 5 are trained on C_p and u/U_∞ around the aerofoil at two angles of incidence, $\alpha_\infty = 0^\circ$ and
 292 $\alpha_\infty = 10^\circ$. To enable fair accuracy comparisons the ridge approximations are obtained on the same
 293 128² grid used for the CNN instead of the finer 90x318 grid used elsewhere. The normalised mean
 294 absolute error of predictions

$$MAE_\sigma = \frac{1}{N} \sum_{i=1}^N \left[\frac{\frac{1}{M} \sum_{m=1}^M |\hat{f}_i(\mathbf{x}_m) - f_i(\mathbf{x}_m)|}{\sigma[f_i(\mathbf{x}_m)]} \right], \quad (17)$$

295 is then measured for the training and test designs. At each point, the absolute error is averaged
 296 for M number of designs, then normalised by the standard deviation of the true data at that point.
 297 The result is averaged over all N number of points. This is done whilst varying the number of
 298 training designs, in order to obtain the error curves in Figure 7. Both the ridge approximation and
 299 CNN prediction errors exhibit the same general trends; for low numbers of training designs the test
 300 errors are considerably higher than the training errors, suggesting a lack of generalisation (i.e. the
 301 models are overfitting to the training data). However the test errors are significantly decreased as
 302 the number of training designs is increased. Perhaps surprisingly, despite the data hungry reputation
 303 of deep neural networks, the CNN framework is able to achieve low accuracies with a relatively
 304 small number of training designs ($MAE_\sigma < 10\%$ with $M_{train} < 100$). As noted by Ronneberger,
 305 Fischer, and Brox⁴⁷, this is a key benefit of the U-Net and other CNN architectures; the strong use
 306 of data augmentation via convolutions allows the available training data to be used more efficiently
 307 (compared to a fully connected deep neural network).

308 Encouragingly, for the $\alpha_\infty = 0^\circ$ case (Figs 7a and 7b) the ridge approximations are able to achieve
 309 mean test errors which are competitive with the CNN. At the higher angle of incidence of $\alpha_\infty = 10^\circ$
 310 (Figs 7c and 7d) the flowfield is more complex, with stronger streamline curvature leading to larger
 311 pressure gradients and greater non-linear behaviour. This case is more challenging for the ridge
 312 approximations, with test errors of $MAE_\sigma \approx 4\%$ compared to the CNN test errors of $MAE_\sigma \approx 1\%$.
 313 However, accuracies of under 4% would be acceptable for many preliminary design applications.
 314 Additionally, as the MAE fields in Figure 8 show, the ridge approximations' prediction errors are
 315 actually lower than the CNN errors in many regions of the flow. Compared to the rather randomly
 316 distributed CNN errors, the ridge approximations' errors exhibit a smoother spatial distribution,
 317 with higher errors near to the aerofoil surface and in the wake.

318 In the above results, ridge approximations are computed using first degree polynomials. In Ta-
 319 ble I, results with higher degree polynomials are presented. Clearly, increasing the polynomial
 320 degree, p , allows for lower test errors to be obtained (although excessively high-degree polynomials
 321 will suffer from over-fitting). However, increasing p also increases the cardinality of the poly-
 322 nomial; if all interaction terms are included, and the polynomial has degree p in all n directions, its
 323 cardinality is given by $L = (p+1)^n$. More polynomial coefficients (recall $\alpha_i \in \mathbb{R}^L$) means longer
 324 training times, a larger model storage size, and a need for more training samples. The n^{th} power
 325 scaling here implies that higher degree polynomials are significantly more costly when in higher
 326 dimensions. This highlights a key benefit of obtaining polynomials in a dimension reducing sub-
 327 space of dimension n , where $n \ll d$. Various approaches exist to mitigate this n^{th} power scaling³⁴.
 328 However, even then, it is still desirable to reduce the number of dimensions. As a compromise
 329 between accuracy and computational cost, second degree polynomials are used in the remainder of
 330 the results presented for the present test case.

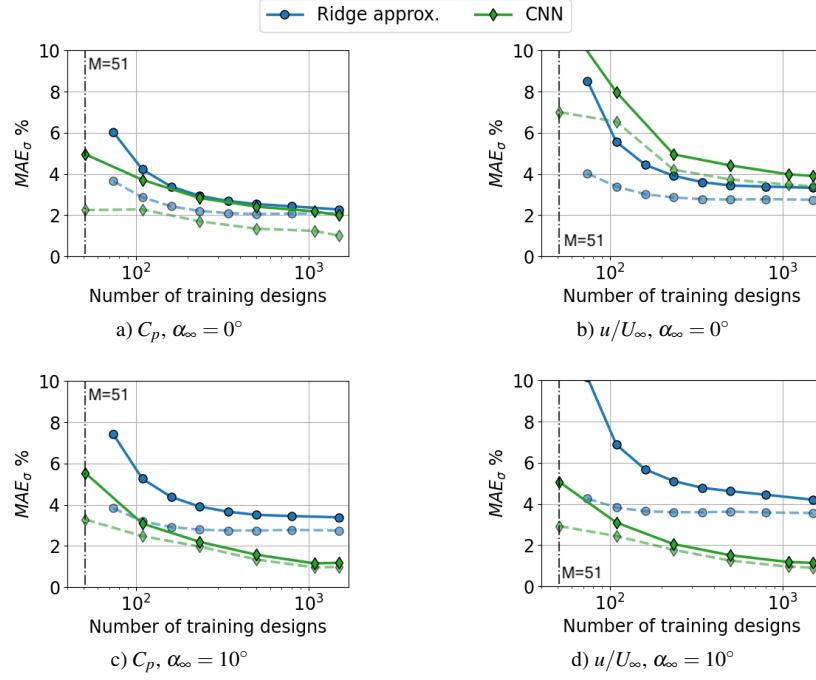


FIG. 7. Normalised MAE of C_p and u/U_∞ predictions for first degree polynomial ridge approximations and convolutional neural network at two angles of incidence. Solid lines = test error, dashed lines = training error. Results are averaged over three randomly selected train/test splits.

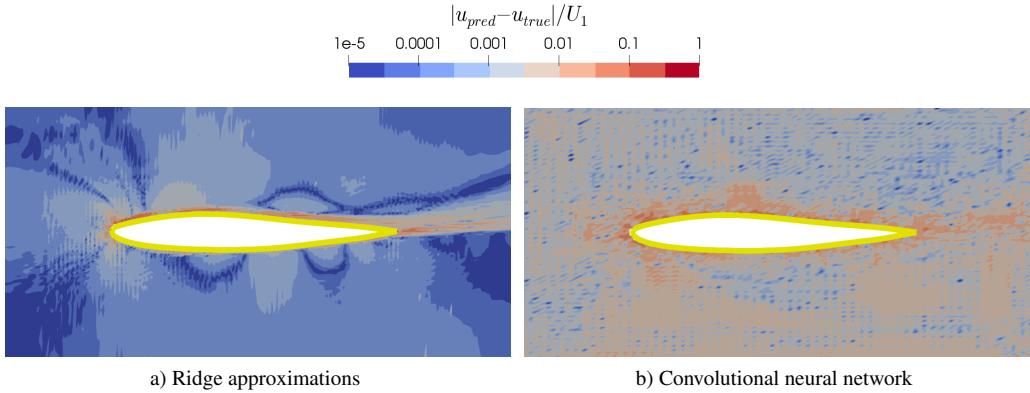


FIG. 8. Comparison of absolute error in predictions of normalised axial velocity from the first degree ridge approximations and convolutional neural network, for an aerofoil from the test set at $\alpha_\infty = 10^\circ$.

331 B. Design Space Exploration

332 The dimension reducing nature of the ridge approximations is particularly useful for design space
 333 exploration. To demonstrate, ridge approximations for the turbulent viscosity ratio v_t/v_∞ are ob-
 334 tained using all 500 training designs. In Figure 9 a sufficient summary plot for a point in the
 335 aerofoil's wake is shown. The turbulent viscosity ratio in the wake is of interest because high levels
 336 of turbulence in wakes can be a leading source of loss in many flows⁴⁹. The sufficient summary plot
 337 is examined, and a design at either end of the $\mathbf{W}^T \mathbf{x}$ subspace is selected. Contours of the turbulent
 338 viscosity ratio for the two designs are shown in Figure 9. Through this approach, the end user can
 339 explore how flowfield variables at select locations in the flowfield change as they traverse the design
 340 space. Such an activity would be far more challenging when negotiating the original design space

TABLE I. Normalised MAE of ridge approximations with various polynomial degrees, for the C_p and u/U_∞ fields at an angle of incidence of $\alpha_\infty = 10^\circ$.

Polynomial degree, p	Training MAE_σ (%)		Test MAE_σ (%)	
	C_p	u/U_∞	C_p	u/U_∞
1	2.8	3.5	3.6	4.5
2	2.1	2.8	2.9	3.7
3	1.6	2.0	2.7	3.6

³⁴¹ $\mathbf{X} \in \mathbb{R}^d$. Effectively, dimension reduction distils the design space into critical directions along which
³⁴² quantities of interest exhibit maximum variance, which enables efficient design space exploration.

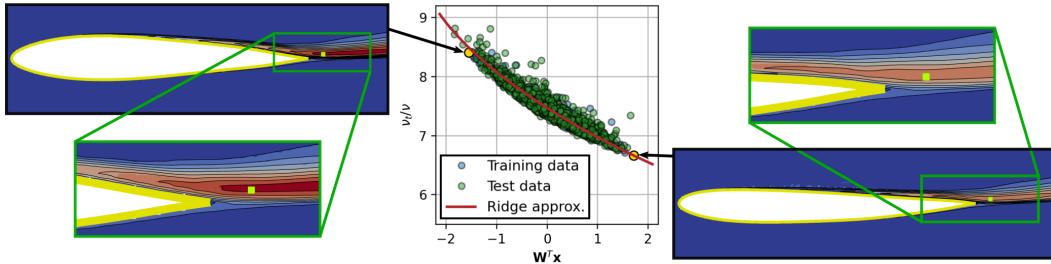


FIG. 9. Turbulent viscosity ratio, v_t/v_∞ , for two deformed aerofoils (from the test set) at an angle of incidence of $\alpha_\infty = 10^\circ$. Colour contours show predictions from the ridge approximations, isolines show the true CFD solutions. The two designs are at either end of the sufficient summary plot for a point in the wake.

³⁴³ In addition to exploring directions of maximum variance, one can also explore directions along
³⁴⁴ which quantities of interest exhibit minimum variance. Following Section II B, the input design
³⁴⁵ vector can be decomposed as

$$\begin{aligned} \mathbf{x} &= \mathbf{W}\mathbf{W}^T\mathbf{x} + \mathbf{V}\mathbf{V}^T\mathbf{x} \\ &= \mathbf{W}\mathbf{w} + \mathbf{V}\mathbf{v} \end{aligned} \quad (18)$$

³⁴⁶ where $\mathbf{w} = \mathbf{W}^T\mathbf{x}$ is the active coordinate and $\mathbf{v} = \mathbf{V}^T\mathbf{x}$ the inactive coordinate. The inactive sub-
³⁴⁷ space \mathbf{V} permits identification of designs which are invariant with regards to the quantity of inter-
³⁴⁸ est. The task is to find input vectors \mathbf{x} with a fixed \mathbf{w} but different \mathbf{v} , whilst obeying the constraint
³⁴⁹ $-1 \leq \mathbf{x} \leq 1$, for which we use a hit-and-run algorithm similar to that implemented in⁵⁰. In Fig-
³⁵⁰ ure 10, this is done for the ridge approximations at the locations labelled a) and b) in Figure 6. The
³⁵¹ resulting *design envelopes* inform us how the aerofoil can be deformed without the quantity of inter-
³⁵² est u/u_∞ at these two locations being affected. Such information can be used to form manufacturing
³⁵³ tolerances, or to decide whether to scrap components, as is done in^{22,51,52}.

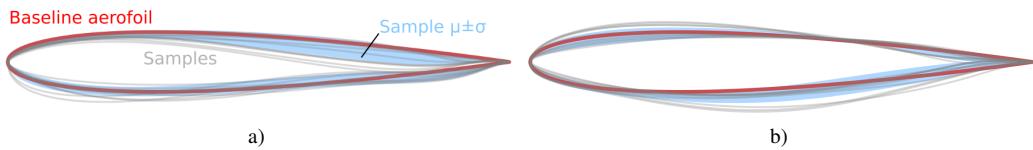


FIG. 10. Design envelopes obtained by sampling the inactive subspaces of the ridge approximations for u/u_∞ at locations a) and b) in Figure 6. 1000 samples are generated using a hit-and-run algorithm⁵⁰, with the grey lines corresponding to the blade profiles of 10 randomly selected samples.

C. Physical Insight

In addition to exploring the design space, ridge functions can also provide important physical insights. As an example, we consider the stagnation pressure around the aerofoil. In an ideal (inviscid) incompressible flow, the stagnation pressure, $p_0 = p + 0.5\rho U^2$, is constant along streamlines[‡]. Streamline curvature results in a reversible exchange between the static pressure p and dynamic pressure $0.5\rho U^2$. However, in a real flow, irreversible processes such as viscous effects cause a loss in p_0 , which it is important to control. Take the case of an aircraft; high losses over the wing are synonymous with high drag[§], while in the engines, high losses lead to decreased component efficiencies, both of which lead to a higher fuel consumption. The pressure loss at a given location \mathbf{s} can be quantified through the loss coefficient

$$Y_p(\mathbf{s}) = \frac{p_{0\infty} - p_0(\mathbf{s})}{0.5\rho_\infty U_\infty^2}, \quad (19)$$

where $p_{0\infty}$ is the freestream stagnation pressure, and $Y_p = 0$ indicates no loss in stagnation pressure, while $Y_p = 1$ indicates a total loss in stagnation pressure.

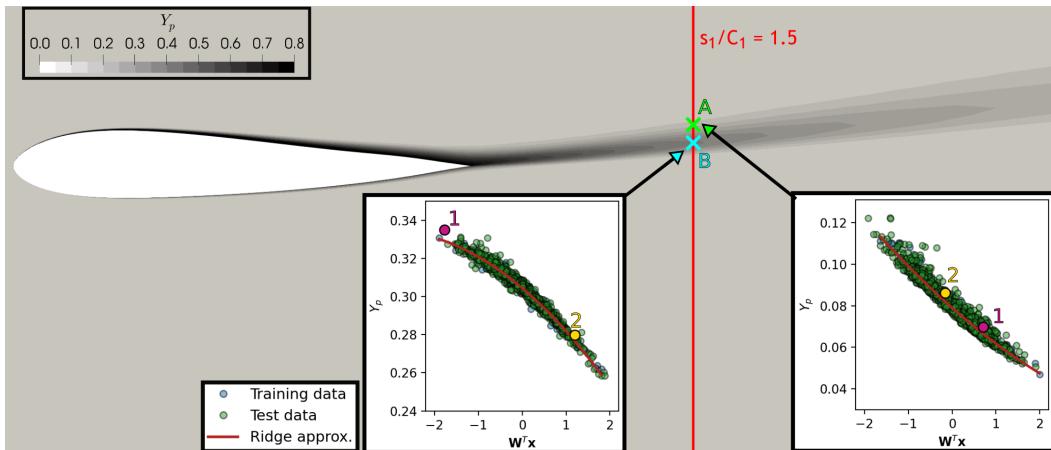


FIG. 11. Contours of loss coefficient, Y_p , for a deformed aerofoil from the test set (“Design 1”) at an angle of incidence of $\alpha_\infty = 10^\circ$. A line at 50% axial chord downstream of the trailing edge is highlighted, and sufficient summary plots are shown for two points within the wake, on this line. Design 1, and a second design, are highlighted in the summary plots.

The loss coefficient Y_p is calculated from the CFD data for 500 training designs, and the approximations for a randomly selected test design are visualised in Figure 11. In Figure 12a, profiles of Y_p across the $s_1/C_1 = 1.5$ plane are plotted for the two designs labelled 1 and 2 in Figure 11. For both designs, the high loss in the wake (around $0 \leq s_2/C_1 \leq 0.15$) is accurately predicted by the ridge functions.

The sufficient summary plots in Figure 11 show how Y_p at probe locations A and B change as the design space is traversed. Furthermore, the dimension reducing subspace vector \mathbf{W} can be examined. Recalling that in the present framework \mathbf{x} contains the Hicks-Henne bump amplitudes, the elements of \mathbf{W} show us how the bump amplitudes affect the reduced coordinates $\mathbf{W}^T \mathbf{x}$. In Figure 12a, the scaled elements of \mathbf{W} for the Y_p ridge functions at A and B are projected onto the baseline aerofoil geometry[§]. The weights offer important physical insights here; they show us how the aerofoil surfaces must be deformed in order to affect the loss coefficient Y_p at probes A and B.

[‡]The stagnation pressure is constant everywhere if the flow is inviscid and irrotational.

[§]Each element of the subspace weight vector is plotted at the surface location of its associated bump.

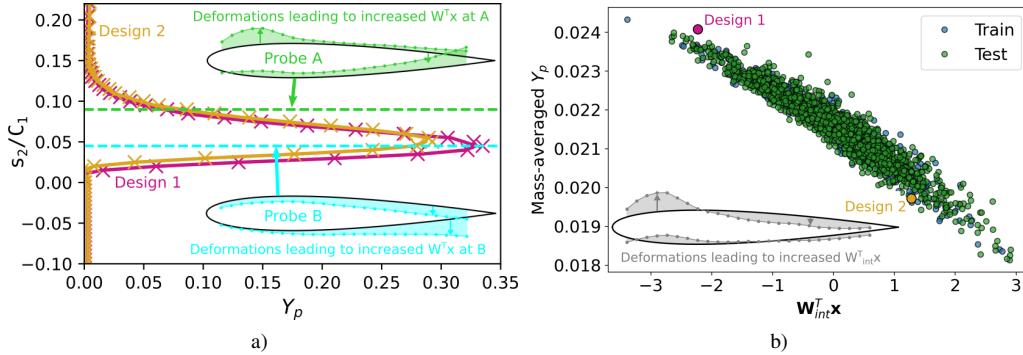


FIG. 12. a) Profiles of loss coefficient Y_p across $s_1/C_1 = 1.5$, and their ridge approximations, for the two designs highlighted in Figure 11. Also shown are the s_2 locations of probes A and B from Figure 11 (dashed lines), along with the projections of \mathbf{W} at these two locations onto the baseline aerofoil. b) Sufficient summary plot for the mass-averaged loss coefficient across the $s_1/C_1 = 1.5$ line, with designs 1 and 2 highlighted, and the associated subspace vector \mathbf{W}_{int} projected onto the baseline aerofoil.

378 D. Obtaining Global Quantities

379 Understanding the sensitivity of local flow variables to a design can be invaluable to a designer.
 380 But, sometimes, a designer may also be interested in global quantities of interest. For example, in
 381 the present case, the mass averaged loss across the $s_1/C_1 = 1.5$ line in Figure 11 is an important
 382 quantity, and is given by

$$\bar{Y}_p = \frac{\int u(s_1, s_2) Y_p(s_1, s_2) ds_2}{\int u(s_1, s_2) ds_2}, \quad (20)$$

383 where $u(s_1, s_2)$ and $Y_p(s_1, s_2)$ denote the local axial velocity and loss coefficient respectively. The
 384 mass averaged loss represents the total pressure loss caused by the aerofoil. A surrogate model for
 385 \bar{Y}_p could be obtained by preprocessing the CFD data, and then computing a ridge approximation
 386 for the precomputed \bar{Y}_p values. However, alternatively, the local ridge functions themselves can be
 387 integrated following the approach outlined in Appendix B. A dimension reducing subspace for \bar{Y}_p ,
 388 \mathbf{W}_{int} , is obtained directly from the ridge approximations for $u(s_1, s_2)$ and $Y_p(s_1, s_2)$. The resulting
 389 one dimensional sufficient summary plot for \bar{Y}_p is shown in Figure 12b. It is apparent that designs
 390 1 and 2 lie at either end of the design space with regards to their mass averaged loss, indicating that
 391 design 2 is objectively the better design (with regards to loss). Similarly to before, in Figure 12b
 392 the elements of \mathbf{W}_{int} are projected onto the baseline aerofoil. This shows that the total loss can be
 393 reduced by deforming the suction surface outwards near the leading edge, and inwards near to the
 394 trailing edge.

395 To understand why design 2 has a lower loss, a second law analysis can be performed. Herwig
 396 and Schmandt⁵⁴ show that the overall entropy generation rate

$$S' = \frac{\rho(v + v_t)}{T} \left(\frac{\partial u_i}{\partial s_j} + \frac{\partial u_j}{\partial s_i} \right) \frac{\partial u_i}{\partial s_j}, \quad (21)$$

397 can be used to understand the sources of drag or pressure loss within a flow. The contours of S' in
 398 Figure 13 show that, at $\alpha = 10^\circ$, the majority of the entropy generation comes from the suction sur-
 399 face boundary layer and the wake region[¶]. In Figure 13 the delta between the y -averaged entropy
 400 generation for design 2 and the baseline design ($\mathbf{W}^T \mathbf{x} = 0$) is shown. Deforming the suction surface

[¶]Decomposing Equation 21 into entropy generation due to laminar viscosity v , and that due to turbulent viscosity v_t , shows that laminar irreversibilities dominate over the aerofoil surfaces, while turbulent mixing dominates in the wake.

401 outwards near the leading edge, and flattening the aft portion (see Fig. 12b), appears to increase
 402 entropy generation over the first quarter of the suction surface. But, this is offset by a reduction
 403 in entropy generation in the wake and leading edge stagnation region, leading to a net decrease in
 404 the entropy generation. An experienced aerodynamicist may have foreseen this conclusion. How-
 405 ever, the present example serves to illustrate the potential of an easily interpretable surrogate model
 406 which can be readily integrated.

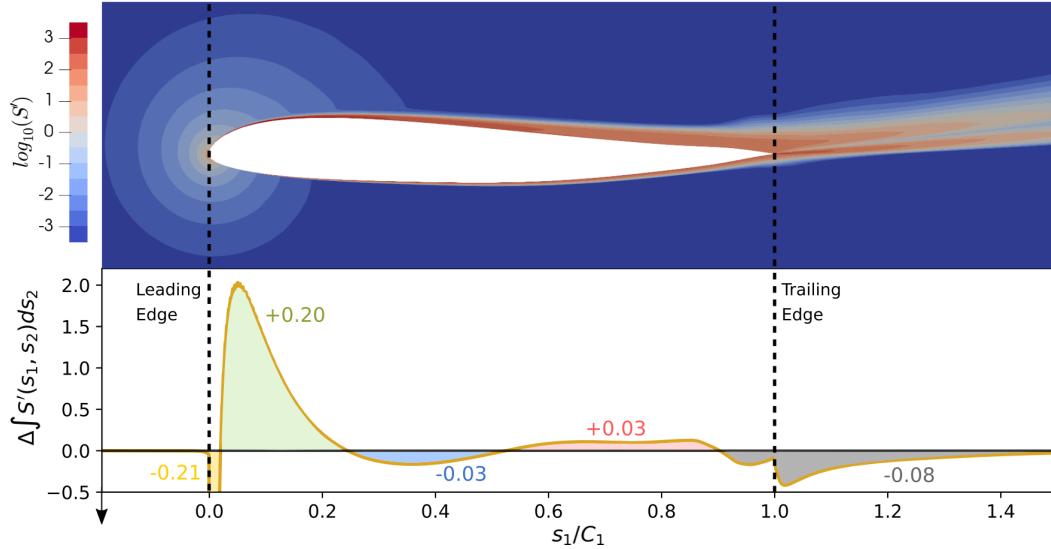


FIG. 13. Contours of total entropy generation rate, S' , for design 2 (see Figs. 11- 12), and axial profile of the difference in y-averaged generation rate between design 2 and the baseline design. The areas under the profile at various locations are also shown. The units of S' are $Wm^{-3}K^{-1}$.

407 E. Exploiting Spatial Correlation

408 Up to this point, flowfield predictions have been made using ridge approximations at every grid
 409 point. For larger grids, such as three dimensional ones, obtaining N number of ridge approximations
 410 might be computationally intractable. Taking a similar approach to how the CNN takes advantage
 411 of spatial coherence in the flowfield, the method outlined in Section II D exploits spatial correlations
 412 to reduce the number of ridge approximations required. In Figure 14, ridge approximations are ob-
 413 tained for $J = 1000$ subsampled points (a sampling rate of $J/N = 3.5\%$). The covariance matrix \mathbf{K} ,
 414 computed on the training data, is then used to *upsample* the ridge approximations at the subsampled
 415 points back to the remaining $N - J$ points.

416 The colouring of the subsampled points in Figure 14 shows that there are a small number of
 417 points with poor quality ridge approximations. Preliminary work³⁸ showed that these poor qual-
 418 ity approximations can cause spurious noise in the flowfield estimates when upsampled. In³⁸ we
 419 use Gaussian ridge functions³¹, which provide an uncertainty estimate (the posterior variance) in
 420 addition to a mean prediction. The posterior variance of each ridge can be added to the ridge's asso-
 421 ciated element in the \mathbf{K}_{22} matrix, which has the effect of *unweighting* poor ridge predictions during
 422 upsampling (due to the \mathbf{K}_{22}^{-1} term in (14)). However, since in the present work polynomial ridge
 423 functions are employed, such uncertainty estimates are not available. Instead, the training R^2 score
 424 is used as a heuristic to measure the quality of each ridge. Ridges with $R^2 < 0.6$ are ignored during
 425 upsampling, with their grid points added back into the $N - J$ set of points. As seen in Figure 14, this
 426 strategy results in qualitatively accurate flowfield predictions using ridge approximations at only
 427 3.5% of the grid points, reducing the training time by a factor of approximately 29.

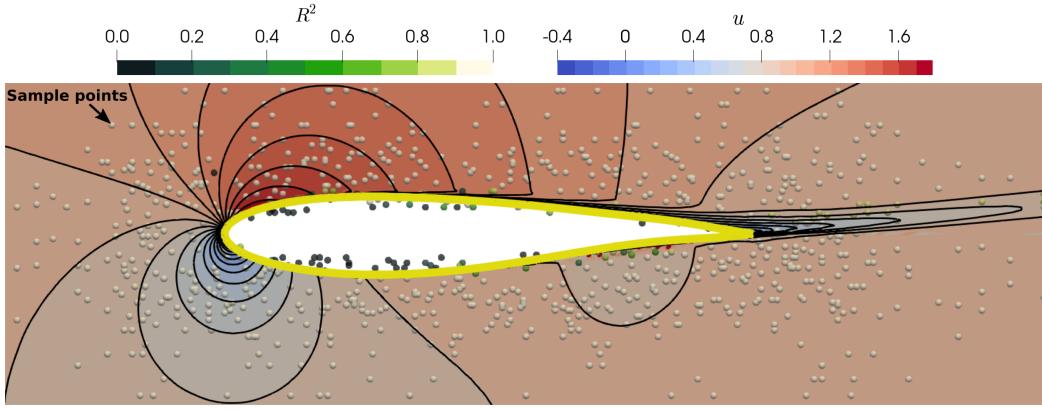


FIG. 14. Flowfield estimate of axial velocity, u/U_∞ , obtained by upsampling ridge approximations at $J = 1000$ randomly subsampled points. The aerofoil is from the test set, and the angle of incidence is $\alpha_\infty = 10^\circ$. Colour contours show upsampled ridge approximations, while the iso-lines show the true CFD solution. The subsampled points are coloured by the mean R^2 score of their ridge approximations over the training data.

428 V. TEST CASE 2: VARYING FREESTREAM CONDITIONS FOR A TRANSONIC WING

429 Following on from the success of the first test case, we now turn to the transonic wing test case
 430 described in Section III A 2 in order to expose the limitations of polynomial ridge functions. Here
 431 the geometry is fixed but the boundary conditions are varied. The transonic flow's transonic nature,
 432 and the possibility of extrapolating beyond the training dataset, make this test case particularly
 433 challenging. The input vector \mathbf{x} only consists of two dimensions in this case, Ma_∞ and α_∞ . We could
 434 once again obtain one dimensional ridge functions. However, two dimensional ridge functions are
 435 still simple to visualise. Recalling the ridge function form

$$f(\mathbf{x}) \approx g(\mathbf{W}^T \mathbf{x}) \quad \text{where} \quad \mathbf{W} \in \mathbb{R}^{d \times n} \quad \text{and} \quad \mathbf{x} \in \mathbb{R}^d, \quad (22)$$

436 we set $\mathbf{W} = \mathbf{I}$ (the identity matrix), resulting in two dimensional ridge functions ($n = d$).

437 A. Interpolation versus extrapolation

438 In this case, the behaviour of the ridge functions as we extrapolate away from the training
 439 boundary conditions is important. To explore this, polynomial ridge functions are trained on the
 440 $M_{train} = 36$ point DoE shown in the inset of Figure 4. The transonic nature of these flows results
 441 in a noticeable step change in the flow behaviour as the freestream Mach number increases, and
 442 at least fifth degree polynomials are found to be necessary to properly capture the surface Mach
 443 number's response. The two dimensional ridge functions are used to estimate the Mach number on
 444 the wing surface at the $s_3/S = 0.2$ span-wise slice highlighted in Figure 4, with the estimates for
 445 two different DoE points from the *test* set shown in Figures 15a and 15b. The corresponding two
 446 dimensional ridges for the locations labelled A and B are then visualised in Figures 15c and 15d.
 447 From these visualisations it is clear that DoE point 1 lies within the training distribution, hence the
 448 it's predictions (Fig. 15a) can be considered to be interpolations. In this case, the predicted Mach
 449 number distribution agrees well with the true distribution.

450 Moving on to DoE point 2, here the ridge approximations must extrapolate away from the flow
 451 conditions seen during training. Unsurprisingly, the predicted distribution in Figure 15b agrees less
 452 well with the true distribution in this case, especially on the aft portion of the wing. This high-
 453 lights a note of caution regarding polynomial ridge functions; as with many function approximation
 454 techniques, caution must be taken when using the underlying polynomials to extrapolate beyond
 455 the training data. If such a task is routinely necessary, one may be better of selecting a different
 456 class of models to use for the ridge approximations. For example, the Gaussian process based ridge

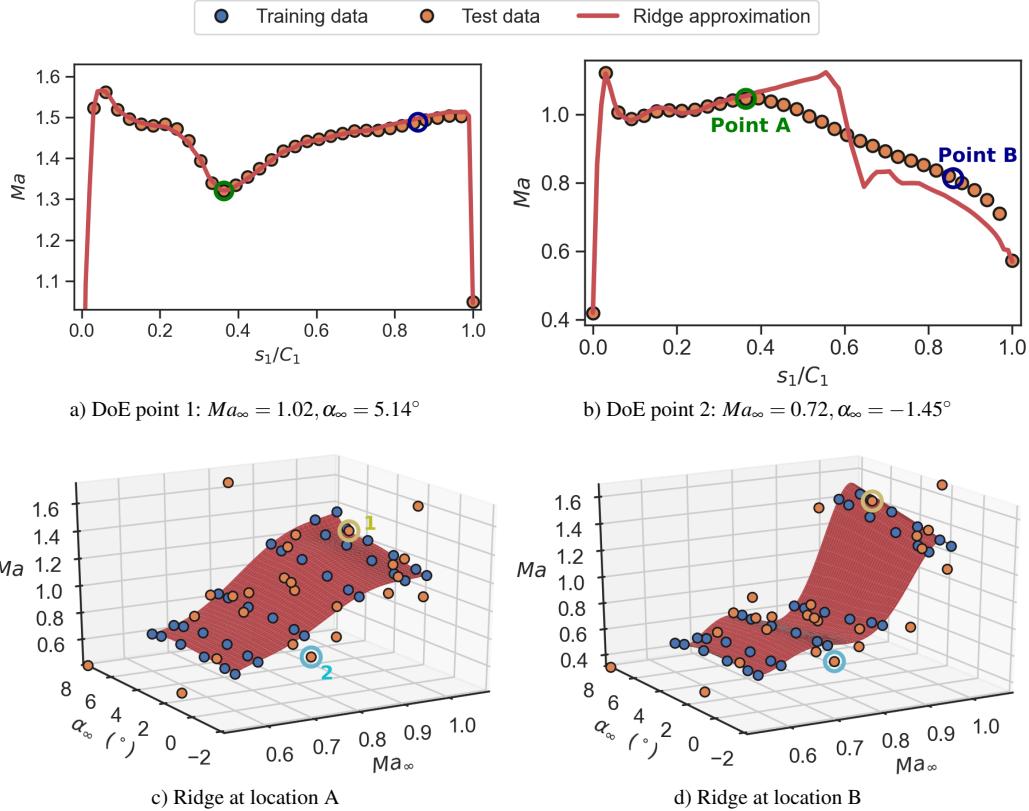


FIG. 15. Polynomial ridge approximations for Mach number on the surface of the transonic wing at the $s_3/S = 0.2$ span-wise slice. The predicted Mach distributions for two DoE points from the *test* set are shown in a) and b), while the corresponding ridges for two s_1/C_1 locations are visualised in c) and d). The two DoE points are also labelled in c) and d).

functions used in^{31,38} exhibit more controllable behaviour when extrapolating, as well as offering principled uncertainty quantification, at the cost of increased training times. Since polynomial ridge functions don't offer uncertainty quantification, a statistical distance measure such as the Mahalanobis distance could instead be used to detect excessive extrapolation (as an example, see⁵⁵).

461 B. Three-dimensional flowfield predictions

When extending the framework to three dimensional flowfields, computational efficiency is crucial. To achieve a surrogate model for the full flowfield the subsampling and upsampling strategy is used once again. Ridge functions are fitted at the $J = 5000$ randomly sampled points (a sampling rate of 4.6%), with a covariance matrix used to upsample the ridge approximations to full three dimensional flowfield estimates. A slice of a flowfield estimate for a *test* DoE point is shown in Figure 16. Comparing the colour contours and iso-lines, the prediction appears to be reasonable, with only a slight disagreement on the aft section of the suction surface.

The above result suggests the subsampling strategy is effective for three dimensional flows in addition to two dimensional ones. The inset figure within Figure 16 shows the effect of sampling rate on the prediction error. For the present case, sampling rates of lower than $\approx 4\%$ lead to significantly increased errors. Nevertheless, sampling rates of 4-10% still allow for significant computational cost savings.

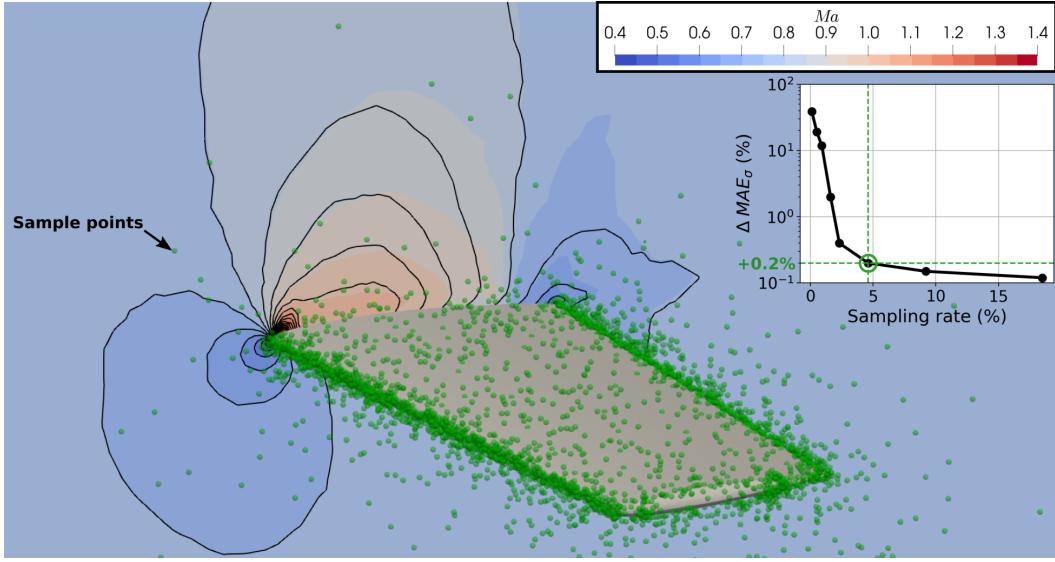


FIG. 16. A span-wise slice ($s_3/S = 0.2$) of the upsampled transonic wing flowfield estimates of Mach number, for a DoE point from the *test* set, with $Ma_\infty = 0.77$ and $\alpha_\infty = 4.6^\circ$. The $J = 5000$ randomly subsampled points are shown in green. The inset figure shows the difference in normalised MAE between the subsampled-upsampled estimates, and the estimates with no subsampling ($J = N$), for different sampling rates.

474 VI. CONCLUSIONS

475 This paper exposes the idea of using spatially correlated ridge functions to rapidly estimate flow-
 476 fields. The resulting data-driven framework could be trained on existing CFD data where available,
 477 or it could be integrated within a wider design of experiment, with new CFD data generated specif-
 478 ically for training. Once trained, the framework provides rapid flowfield predictions, which can be
 479 used for design space exploration, design optimisation tasks, performance predictions, or even to
 480 guide CFD mesh design.

481 On the subsonic aerofoil test case, comparisons with the state-of-the-art convolutional neural
 482 network (CNN) suggest that the ridge function framework is able to achieve competitive predictive
 483 accuracy on unseen aerofoil designs. In addition to serving as a surrogate model, the learned ridge
 484 functions can aid understanding of the existing training data. *Sufficient summary plots* can be viewed
 485 for any given point. The reduced dimensional nature of these plots lends itself to easy visualisation,
 486 allowing for easy comparison between designs, and new physical insights. Furthermore, the local
 487 ridge functions can be integrated to obtain new ridge functions for integral quantities such as a loss
 488 or drag coefficient. This allows new interpretable surrogate models for integral quantities to be
 489 generated on demand, without the need for further pre-processing of the CFD data. Such capability
 490 is advantageous in scenarios where the long-term storage of the full CFD dataset is problematic.
 491 The ridge functions can also provide *insensitivity* information, informing us how a design can be
 492 altered without influencing the flow at a given location.

493 The independent nature of each ridge function means that their training can be viewed as an em-
 494 barrassingly parallel task. This makes the ridge function framework trivial to implement in a parallel
 495 fashion, allowing for excellent scaling with problem size. However, for larger problems, computing
 496 ridge functions for every single grid point is undesirable. Instead, ridge functions are obtained at a
 497 much smaller number of randomly subsampled points. The flow physics encoded within covariance
 498 matrices, computed from the training data, can be used to upsample the ridge functions' predictions
 499 back to the rest of the flowfield. The computation of the covariance matrices, and the subsequent
 500 use of them, involves a variety of linear algebra operations which can be implemented efficiently.
 501 For the three dimensional transonic wing test case, sampling rates around 5% are achieved with
 502 minimal additional errors, allowing for significant reductions in training time.

503 The transonic wing test case demonstrates how ridge functions can be used to estimate flowfields
 504 with varying boundary conditions. In such cases, there is a danger of extrapolating too far from the
 505 training data, and the underlying polynomials' predictions can not be trusted here. If extrapolation is
 506 important, there is scope to replace the polynomials used in this work with alternative models, such
 507 as Gaussian processes. In addition to exploring this further, a worthwhile area of future work would
 508 be to combine the approaches used in the two test cases, incorporating geometric and boundary
 509 condition input parameters together.

510 ACKNOWLEDGMENTS

511 This research was supported in part through computational resources provided by The Alan Tur-
 512 ing Institute and with the help of a generous gift from Microsoft Corporation. The authors were
 513 supported by Wave 1 of The UKRI Strategic Priorities Fund under the EPSRC Grant EP/T001569/1,
 514 particularly the “Digital twins for complex systems engineering” theme within that grant and The
 515 Alan Turing Institute, and by the Lloyd’s Register Foundation-Alan Turing Institute programme on
 516 Data-Centric Engineering under the LRF grant G0095.

517 Appendix A: Convolutional Neural Network Setup

518 This appendix provides more details on the convolutional neural network setup in Figure 5. This
 519 network is a modified form of that proposed by Thuerey *et al.*¹⁴, which is based upon the U-Net
 520 architecture first proposed by Ronneberger, Fischer, and Brox⁴⁷. Since the boundary conditions are
 521 scalar values, it might appear wasteful to repeat the values over the entirety of the $128 \times 128 \times 3$
 522 boundary condition input channels, and some CNN flowfield prediction frameworks such as those
 523 of Bhatnagar *et al.*¹³ inject the scalar values into the feature vector instead. However, Thuerey
 524 *et al.*¹⁴ claim that, while the network would eventually propagate the boundary information via the
 525 convolutional bow tie structure, specifying the redundant boundary condition information with skip
 526 connections allows for a more efficient training process.

527 The general design features of the convolutional blocks from Thuerey *et al.*¹⁴ are retained; in the
 528 encoder leaky ReLU activations functions are chosen to avoid the *dying ReLU problem*⁵⁶, whilst
 529 standard ReLU functions are selected for the decoder, both with a slope of 0.2. Convolutional filter
 530 kernels of various sizes and strides are then used for encoding and decoding (See Fig. 5), with
 531 nearest neighbour upsampling followed by a regular convolution for the decoder blocks. To help
 532 mitigate overfitting, batch normalisation is used in all the blocks except for the first and last ones
 533 (1c and 1d), in addition to having a slight dropout rate of 0.01 for all layers.

534 1. Learning process

535 To obtain the the weights (and biases) of the network, the learning process uses the *Adam op-*
 536 *timiser*⁵⁷ to minimise the Huber loss function given in (16), with the loss averaged over all M_{train}
 537 number of training designs and $N = 128^2$ grid points. A learning rate of 4×10^{-4} is used, which is
 538 then reduced by a factor of ten after 500 epochs, and a batch size of ten is chosen.

539 The learning curve for the 488k parameter network is plotted in Figure 17; the convergence
 540 behaviour appears to be satisfactory, and the small difference between train and test loss suggests
 541 the network isn't suffering from excessive overfitting. Similar behaviour is observed for the other
 542 network sizes explored in this paper.

543 2. Effect of network size

544 To allow for a fair comparison between the ridge function framework and the CNN, the per-
 545 formance for various size/complexity networks is examined in order to find an optimal size. The

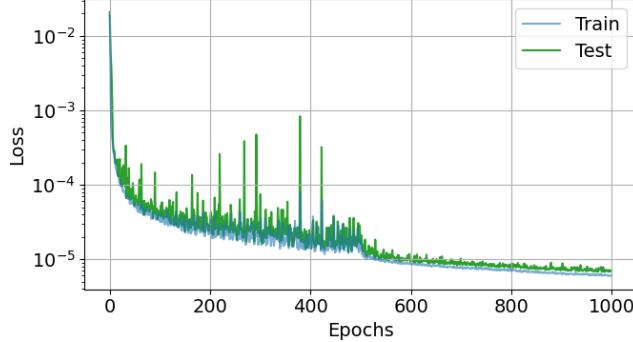


FIG. 17. Convergence history for the 488k parameter CNN trained on the $\alpha_\infty = 10^\circ$ dataset, with $M_{train} = 500$.

network's size is altered by changing the number of channels, with the number of channels in the i^{th} layer given by $C_i = 2^e c_i$. The constants c_i are defined in Table II, and the network's size is varied by adjusting the exponent e , with values of $e = \{2, 3, 4, 5\}$ tested. The variation in performance

TABLE II. Multiplier c used to set the number of channels in each layer of the convolutional neural network. The number of channels in the i^{th} layer is given by $C_i = 2^e c_i$, where the exponent e is a scalar parameter used to set the model complexity.

	Layer	1c	2c	3c	4c	5c	6c	7c
Encoder	c_i	1	2	2	4	8	8	8
Decoder	Layer	1d	2d	3d	4d	5d	6d	7d
	c_i	8	16	16	8	4	4	4

The normalised MAE, defined in (17), is measured on the training and test sets for the four different network sizes, with the results plotted in Figure 18. Generally, the larger networks achieve lower predictive errors. However, this is at the cost of longer training times (Fig. 18d), with the 1.94M parameter network taking up to 80 minutes to train. The larger networks also exhibit more stable predictions, with less variance in their MAE scores across the three train/test folds. To achieve a reasonable compromise between accuracy and training cost, the 488k parameter is chosen as the benchmark for comparisons elsewhere in this paper. A similar pattern between MAE, training times, and the number of training designs is also observed. The chosen value of $M_{train} = 500$ is seen to offer a good compromise between low MAE scores and short training time.

Appendix B: Integrating Ridge Functions

This appendix presents a procedure for identifying a dimension reducing subspace for the mass-averaged pressure loss coefficient \bar{Y}_p , defined in (20). A more general treatment on the subspaces of an *integral of a scalar field quantity* can be found in Wong *et al.*²³. For simplicity, we stick to a two dimensional domain, but the approach is readily generalisable to three dimensions. Recalling (20), we wish to obtain the mass-averaged loss coefficient \bar{Y}_p , at a given axial chord location C_x

$$\bar{Y}_p = \frac{1}{\dot{m}} \int_{C_1} u(s_1, s_2) Y_p(s_1, s_2) ds_2, \quad (B1)$$

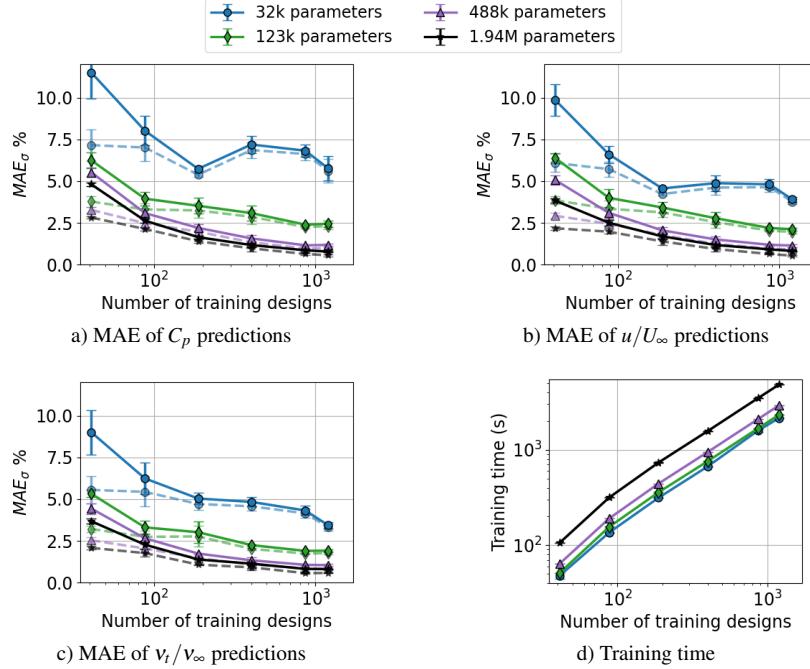


FIG. 18. Prediction errors and training time for different size convolutional neural networks, for the $\alpha_\infty = 10^\circ$ dataset. Solid lines indicate test errors, and dashed lines training errors. Results are averaged over three train/test folds, and the error bars denote the standard deviation in MAE_σ across these folds. The size of the network is altered by adjusting the number of channels via the exponent e .

where the mass flow rate across the plane, $\dot{m} = \int_{C_1} u(s_1, s_2) ds_2$, is taken to be a constant here^{||}. The local axial velocity and loss coefficient are approximated with two ridge approximations

$$\begin{aligned} u(s_1, s_2) &\approx g_{s_1, s_2} (\mathbf{W}_{s_1, s_2}^T \mathbf{x}) \\ Y_p(s_1, s_2) &\approx h_{x,y} (\mathbf{U}_{s_1, s_2}^T \mathbf{x}). \end{aligned} \quad (B2)$$

Here the input vector $\mathbf{x} \in \mathbb{R}^d$ has its usual meaning, whilst $\mathbf{W}^T \in \mathbb{R}^{n_w \times d}$ and $\mathbf{U}^T \in \mathbb{R}^{n_u \times d}$ denote the subspaces for axial velocity and loss coefficient respectively, where naturally $n_w \ll d$ and $n_u \ll d$. Substituting (B2) into (B1) yields

$$\bar{Y}_p = \rho_\infty \int_{C_1} g_{s_1, s_2} (\mathbf{W}_{s_1, s_2}^T \mathbf{x}) h_{s_1, s_2} (\mathbf{U}_{s_1, s_2}^T \mathbf{x}) ds_2. \quad (B3)$$

To identify the dimension reducing subspace for \bar{Y}_p with respect to its constituent input parameters \mathbf{x} , we need to compute the *averaged outer product of its gradient*³⁷ – a covariance matrix given by

$$\mathbf{C} = \int_{\mathbb{R}^d} (\nabla_{\mathbf{x}} \bar{Y}_p(\mathbf{x})) (\nabla_{\mathbf{x}} \bar{Y}_p(\mathbf{x}))^T \omega(\mathbf{x}) d\mathbf{x}, \quad (B4)$$

where $\omega(\mathbf{x})$ denotes the distribution associated with the input parameters \mathbf{x} . Assuming the existence of a design of experiment with M distinct CFD evaluations for different input parametrisations, and taking $\omega(\mathbf{x})$ to be a uniform distribution here, we can approximate \mathbf{C} with its finite sample estimate

$$\mathbf{C} \approx \frac{1}{M} \sum_{m=1}^M (\nabla_{\mathbf{x}} \bar{Y}_p(\mathbf{x}_m)) (\nabla_{\mathbf{x}} \bar{Y}_p(\mathbf{x}_m))^T. \quad (B5)$$

^{||}The standard deviation of \dot{m} across the C_1 plane is less than 0.2% of the mean.

⁵⁷⁴ The gradient evaluations can be written as

$$\begin{aligned}\nabla_{\mathbf{x}} \bar{Y}_p(\mathbf{x}_m) &= \rho_\infty \int_{\mathbb{R}^d} \frac{\partial}{\partial \mathbf{x}_m} [g_{s_1, s_2}(\mathbf{W}_{s_1, s_2}^T \mathbf{x}_m) h_{s_1, s_2}(\mathbf{U}_{s_1, s_2}^T \mathbf{x}_m)] ds_2 \\ &= \rho_\infty \int_{\mathbb{R}^d} [\mathbf{W}_{s_1, s_2} \nabla_{\mathbf{w}} g_{s_1, s_2}(\mathbf{w}_m) h_{s_1, s_2}(\mathbf{u}_m) + \mathbf{U}_{s_1, s_2} \nabla_{\mathbf{u}} h_{s_1, s_2}(\mathbf{u}_m) g_{s_1, s_2}(\mathbf{w}_m)] ds_2,\end{aligned}\quad (\text{B6})$$

⁵⁷⁵ where $\mathbf{w}_m = \mathbf{W}^T \mathbf{x}_m$ and $\mathbf{u}_m = \mathbf{U}^T \mathbf{x}_m$ are the reduced dimensional co-ordinates for the m^{th} design in
⁵⁷⁶ the $Y_p(s_1, s_2)$ and $u(s_1, s_2)$ subspaces respectively. If we further assume that spatially, there are N
⁵⁷⁷ nodes along which this integration has to be performed, in which case we have the set

$$\{g_i, h_i, \mathbf{W}_i, \mathbf{U}_i, \nabla_{\mathbf{w}} g_i, \nabla_{\mathbf{u}} h_i, \}_{i=1}^N, \quad (\text{B7})$$

⁵⁷⁸ we approximate

$$\nabla_{\mathbf{x}} \bar{Y}_p(\mathbf{x}_m) \approx \rho_\infty \sum_{i=1}^N [\mathbf{W}_i \nabla_{\mathbf{w}} g_i h_i + \mathbf{U}_i \nabla_{\mathbf{u}} h_i g_i] \varphi_i \quad (\text{B8})$$

⁵⁷⁹ where φ_i denotes appropriately selected quadrature weights depending on the spatial location of the
⁵⁸⁰ nodal centres. Plugging this into (B5) yields

$$\begin{aligned}\mathbf{C} &\approx \mathbb{E} \left[\left(\rho_\infty \sum_{i=1}^N [\mathbf{W}_i \nabla_{\mathbf{w}} g_i h_i + \mathbf{U}_i \nabla_{\mathbf{u}} h_i g_i] \varphi_i \right) \left(\rho_\infty \sum_{k=1}^N [\mathbf{W}_k \nabla_{\mathbf{w}} g_k h_k + \mathbf{U}_k \nabla_{\mathbf{u}} h_k g_k] \varphi_k \right)^T \right] \\ &= \rho_\infty^2 \sum_{i=1}^N \sum_{k=1}^N \varphi_i \varphi_k \mathbb{E} [h_i h_k \mathbf{W}_i \nabla_{\mathbf{w}} g_i \nabla_{\mathbf{w}} g_k^T \mathbf{W}_k^T] + \rho_\infty^2 \sum_{i=1}^N \sum_{k=1}^N \varphi_i \varphi_k \mathbb{E} [h_i g_k \mathbf{W}_i \nabla_{\mathbf{w}} g_i \nabla_{\mathbf{u}} h_k^T \mathbf{U}_k^T] + \\ &\quad \rho_\infty^2 \sum_{i=1}^N \sum_{k=1}^N \varphi_i \varphi_k \mathbb{E} [g_i h_k \mathbf{U}_i \nabla_{\mathbf{u}} h_i \nabla_{\mathbf{w}} g_k^T \mathbf{W}_k^T] + \rho_\infty^2 \sum_{i=1}^N \sum_{k=1}^N \varphi_i \varphi_k \mathbb{E} [g_i g_k \mathbf{U}_i \nabla_{\mathbf{u}} h_i \nabla_{\mathbf{u}} h_k^T \mathbf{U}_k^T]\end{aligned}\quad (\text{B9})$$

⁵⁸¹ This equation is easy to implement, as long as the gradients $\nabla_{\mathbf{w}} g$ and $\nabla_{\mathbf{u}} h$ can be computed.
⁵⁸² When applying this approach in Section IV C, g and h are taken to be the underlying polynomials
⁵⁸³ of the ridge function approximations, rather than the posterior mean of the ridge kernels as is the
⁵⁸⁴ case in the rest of the paper. This is done since the gradients of orthogonal polynomials are readily
⁵⁸⁵ available. Finally, an eigendecomposition of \mathbf{C} is performed, and the leading eigenvectors with the
⁵⁸⁶ largest eigenvalues are chosen to form \mathbf{W}_{int} , the dimension reducing subspace for \bar{Y}_p . In the present
⁵⁸⁷ case, the first eigenvalue was found to contribute 99.3% of the trace. Therefore, a one dimensional
⁵⁸⁸ dimension reducing subspace was deemed to be sufficient.

⁵⁸⁹ ¹T. Kobayashi and M. Tsubokura, “CFD application in automotive industry,” in *Notes on Numerical Fluid Mechanics and Multidisciplinary Design* (Springer Berlin Heidelberg, 2009) pp. 285–295.

⁵⁹⁰ ²F. Ding, A. Kareem, and J. Wan, “Aerodynamic tailoring of structures using computational fluid dynamics,” *Structural Engineering International* **29**, 26–39 (2019).

⁵⁹¹ ³R. Ramponi and B. Blocken, “CFD simulation of cross-ventilation for a generic isolated building: Impact of computational parameters,” *Building and Environment* **53**, 34–48 (2012).

⁵⁹² ⁴G. K. Saha, M. H. I. Maruf, and M. R. Hasan, “Marine propeller modeling and performance analysis using CFD tools,” in *8th BSME International Conference on Thermal Engineering* (AIP Publishing, 2019).

⁵⁹³ ⁵T. Dbouk and D. Drikakis, “On coughing and airborne droplet transmission to humans,” *Physics of Fluids* **32**, 053310 (2020).

⁵⁹⁴ ⁶T. Dbouk and D. Drikakis, “On respiratory droplets and face masks,” *Physics of Fluids* **32**, 063303 (2020).

⁵⁹⁵ ⁷K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, “Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns,” *IEEE Control Systems* **38**, 63–86 (2018).

⁵⁹⁶ ⁸S. Discetti, M. Raiola, and A. Ianiro, “Estimation of time-resolved turbulent fields through correlation of non-time-resolved field measurements and time-resolved point measurements,” *Experimental Thermal and Fluid Science* **93**, 119–130 (2018).

⁵⁹⁷ ⁹N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz, “Shallow neural networks for fluid flow reconstruction with limited sensors,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **476**, 20200097 (2020).

⁵⁹⁸ ¹⁰M. Raissi and G. E. Karniadakis, “Hidden physics models: Machine learning of nonlinear partial differential equations,” *Journal of Computational Physics* **357**, 125–141 (2018).

- ⁶¹⁰ ¹¹L. Agostini, "Exploration and prediction of fluid dynamical systems using auto-encoder technology," Physics of Fluids **32**, 067103 (2020).
- ⁶¹¹ ¹²X. Guo, W. Li, and F. Iorio, "Convolutional neural networks for steady flow approximation," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, 2016).
- ⁶¹² ¹³S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, "Prediction of aerodynamic flow fields using convolutional neural networks," Comput. Mech. (2019), 10.1007/s00466-019-01740-0, arXiv:1905.13166.
- ⁶¹³ ¹⁴N. Thuerey, K. Weißerow, L. Prantl, and X. Hu, "Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows," AIAA J. **58**, 25–36 (2020), arXiv:1810.08217.
- ⁶¹⁴ ¹⁵J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin, "Accelerating Eulerian fluid simulation with convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 70, edited by D. Precup and Y. W. Teh (PMLR, International Convention Centre, Sydney, Australia, 2017) pp. 3424–3433.
- ⁶¹⁵ ¹⁶V. Sekar, M. Zhang, C. Shu, and B. C. Khoo, "Inverse design of airfoil using a deep convolutional neural network," AIAA Journal **57**, 993–1003 (2019).
- ⁶¹⁶ ¹⁷X. Jin, P. Cheng, W.-L. Chen, and H. Li, "Prediction model of velocity field around circular cylinder over various reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder," Physics of Fluids **30**, 047105 (2018).
- ⁶¹⁷ ¹⁸T. J. Sejnowski, "The unreasonable effectiveness of deep learning in artificial intelligence," Proceedings of the National Academy of Sciences , 201907373 (2020).
- ⁶¹⁸ ¹⁹Q. Zhang, Y. N. Wu, and S. Zhu, "Interpretable convolutional neural networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018) pp. 8827–8836.
- ⁶¹⁹ ²⁰A. Pinkus, *Ridge functions*, Vol. 205 (Cambridge University Press, 2015).
- ⁶²⁰ ²¹Z. del Rosario, P. Constantine, and G. Iaccarino, "Developing design insight through active subspaces," in *19th AIAA Non-Deterministic Approaches Conference* (American Institute of Aeronautics and Astronautics, 2017).
- ⁶²¹ ²²A. D. Scillitoe, B. Ubald, P. Seshadri, and S. Shahpar, "Design space exploration of stagnation temperature probes via dimension reduction," in *Proc. ASME Turbo Expo* (2020).
- ⁶²² ²³C. Y. Wong, P. Seshadri, G. T. Parks, and M. Girolami, "Embedded ridge approximations," Computer Methods in Applied Mechanics and Engineering **372**, 113383 (2020).
- ⁶²³ ²⁴K.-C. Li, "Sliced inverse regression for dimension reduction," Journal of the American Statistical Association **86**, 316–327 (1991).
- ⁶²⁴ ²⁵K.-C. Li, "On principal hessian directions for data visualization and dimension reduction: Another application of stein's lemma," Journal of the American Statistical Association **87**, 1025–1039 (1992).
- ⁶²⁵ ²⁶B. Li, H. Zha, F. Chiaromonte, *et al.*, "Contour regression: a general approach to dimension reduction," The Annals of Statistics **33**, 1580–1616 (2005).
- ⁶²⁶ ²⁷R. D. Cook and S. Weisberg, "Sliced inverse regression for dimension reduction: Comment," Journal of the American Statistical Association **86**, 328–332 (1991).
- ⁶²⁷ ²⁸Y. Xia, H. Tong, W. K. Li, and L.-X. Zhu, "An adaptive estimation of dimension reduction space," Journal of the Royal Statistical Society: Series B (Statistical Methodology) **64**, 363–410 (2002).
- ⁶²⁸ ²⁹P. G. Constantine, A. Eftekhari, J. Hokanson, and R. A. Ward, "A near-stationary subspace for ridge approximation," Computer Methods in Applied Mechanics and Engineering **326**, 402–421 (2017).
- ⁶²⁹ ³⁰J. M. Hokanson and P. G. Constantine, "Data-driven polynomial ridge approximation using variable projection," SIAM Journal on Scientific Computing **40**, A1566–A1589 (2018).
- ⁶³⁰ ³¹P. Seshadri, S. Yuchi, and G. T. Parks, "Dimension reduction via gaussian ridge functions," SIAM/ASA Journal on Uncertainty Quantification **7**, 1301–1322 (2019).
- ⁶³¹ ³²X. Liu and S. Guillas, "Dimension reduction for gaussian process emulation: An application to the influence of bathymetry on tsunami heights," SIAM/ASA Journal on Uncertainty Quantification **5**, 787–812 (2017).
- ⁶³² ³³P. Seshadri, S. Shahpar, P. Constantine, G. Parks, and M. Adams, "Turbomachinery active subspace performance maps," Journal of Turbomachinery **140** (2018).
- ⁶³³ ³⁴P. Seshadri, A. Narayan, and S. Mahadevan, "Effectively subsampled quadratures for least squares polynomial approximations," SIAM/ASA Journal on Uncertainty Quantification **5**, 1003–1023 (2017).
- ⁶³⁴ ³⁵C. Y. Wong, P. Seshadri, and G. Parks, "Extremum sensitivity analysis with polynomial monte carlo filtering," Reliability Engineering & System Safety , 107609 (2021).
- ⁶³⁵ ³⁶G. Golub and V. Pereyra, "Separable nonlinear least squares: the variable projection method and its applications," Inverse problems **19**, R1 (2003).
- ⁶³⁶ ³⁷P. G. Constantine, *Active subspaces: Emerging ideas for dimension reduction in parameter studies*, Vol. 2 (SIAM, 2015).
- ⁶³⁷ ³⁸A. D. Scillitoe, P. Seshadri, and C. Y. Wong, "Instantaneous flowfield estimation with gaussian ridges," in *AIAA Scitech 2021 Forum* (2021) p. 1138.
- ⁶³⁸ ³⁹B. Diskin, J. L. Thomas, C. L. Rumsey, and A. Schwöppe, "Grid-convergence of reynolds-averaged navier–stokes solutions for benchmark flows in two dimensions," AIAA Journal **54**, 2563–2588 (2016).
- ⁶³⁹ ⁴⁰R. M. Hicks and P. A. Henne, "Wing design by numerical optimization," Journal of Aircraft **15**, 407–412 (1978).
- ⁶⁴⁰ ⁴¹T. D. Economou, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, "SU2: An open-source suite for multi-physics simulation and design," AIAA Journal **54**, 828–846 (2016).
- ⁶⁴¹ ⁴²P. Spalart and S. Allmaras, "A one-equation turbulence model for aerodynamic flows," in *30th Aerospace Sciences Meeting and Exhibit* (American Institute of Aeronautics and Astronautics, 1992).
- ⁶⁴² ⁴³C. Sullivan and A. Kaszynski, "PyVista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK)," Journal of Open Source Software **4**, 1450 (2019).
- ⁶⁴³ ⁴⁴V. Schmitt and F. Charpin, "Pressure distributions on the onera m6 wing at transonic mach numbers," in *Experimental data base for computer program assessment. Report of the Fluid Dynamics Panel, Working Group 04* (AGARD, 1979).

- 677 ⁴⁵P. Seshadri and G. Parks, "Effective-Quadratures (EQ): Polynomials for Computational Engineering Studies," *J. Open*
678 *Source Softw.* **2**, 166 (2017), arXiv:1601.05470.
- 679 ⁴⁶C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg,
680 N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson,
681 P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array
682 programming with NumPy," *Nature* **585**, 357–362 (2020).
- 683 ⁴⁷O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture*
684 *Notes in Computer Science* (Springer International Publishing, 2015) pp. 234–241.
- 685 ⁴⁸P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics* **35**, 73–101 (1964).
- 686 ⁴⁹A. D. Scillitoe, P. G. Tucker, and P. Adami, "Numerical Investigation of Three-Dimensional Separation in an Axial Flow
687 Compressor: The Influence of Freestream Turbulence Intensity and Endwall Boundary Layer State," *J. Turbomach.* **139**,
688 021011 1–10 (2016).
- 689 ⁵⁰P. Constantine, R. Howard, A. Glaws, Z. Grey, P. Diaz, and L. Fletcher, "Python active-subspaces utility library," *Journal*
690 *of Open Source Software* **1**, 79 (2016).
- 691 ⁵¹C. Y. Wong, P. Seshadri, A. Scillitoe, A. B. Duncan, and G. Parks, "Blade envelopes part i: Concept and methodology,"
692 (2021), arXiv:2011.11636 [cs.CE].
- 693 ⁵²C. Y. Wong, P. Seshadri, A. Scillitoe, B. N. Ubald, A. B. Duncan, and G. Parks, "Blade envelopes part ii: Multiple
694 objectives and inverse design," (2020), arXiv:2012.15579 [cs.CE].
- 695 ⁵³W. Wang, J. Wang, H. Liu, and B. yan Jiang, "CFD prediction of airfoil drag in viscous flow using the entropy generation
696 method," *Mathematical Problems in Engineering* **2018**, 1–15 (2018).
- 697 ⁵⁴H. Herwig and B. Schmandt, "How to determine losses in a flow field: A paradigm shift towards the second law analysis,"
698 *Entropy* **16**, 2959–2989 (2014).
- 699 ⁵⁵J. L. Wu, J. X. Wang, H. Xiao, and J. Ling, "A Priori Assessment of Prediction Confidence for Data-Driven Turbulence
700 Modeling," *Flow, Turbul. Combust.* **99**, 25–46 (2017), arXiv:1607.04563.
- 701 ⁵⁶L. Lu, "Dying ReLU and initialization: Theory and numerical examples," *Communications in Computational Physics* **28**,
702 1671–1706 (2020).
- 703 ⁵⁷D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," (2014), arXiv:1412.6980.