Bayesian Approaches for Efficient and Uncertainty-Aware Prediction of Pressure Distributions

Mehdi Anhichem[*] and Sebastian Timme[†]
University of Liverpool, Liverpool, L69 3GH, United Kingdom

Jony Castagna[‡]
UKRI-STFC Hartree Centre, Warrington, WA4 4AD, United Kingdom

Andrew J. Peace[§] and Moira Maina[¶]
Aircraft Research Association Ltd, Bedford, MK41 7PF, United Kingdom

Computational fluid dynamics plays a vital role in aircraft design by providing invaluable insight into aerodynamic characteristics and allowing designers to improve aircraft performance and efficiency. High-fidelity simulations based on the (Reynolds-averaged) Navier–Stokes equations are still too expensive for tasks requiring many evaluations such as design optimisation or uncertainty quantification. Instead, deploying machine learning to build surrogate models offers approaches to approximate the quantities of interest in the design space. However, as these models are not exact, it is advantageous to have a measure of the model epistemic uncertainty. The Bayesian paradigm offers a rigorous framework to train and analyse uncertainty-aware models. This work focuses on evaluating the effectiveness of Bayesian surrogate models, specifically Bayesian neural networks and Gaussian processes, in predicting aerodynamic pressure distributions. Computational challenges arising from large data sets, necessitating either approximations such as stochastic variational inference or powerful parallelisation (or both), are addressed. A detailed comparison is made, considering both model accuracy and predicted uncertainty, for an aerofoil case. Our primary ambition is to assess the contribution and added value of Bayesian surrogate models as a tool in predicting surface pressure distributions.

Nomenclature

C_p	=	pressure coefficient
$\mathfrak{D}$	=	input space
d	=	dimension of the input space
f	=	scalar quantity of interest
$\mathbf{f}$	=	vector of observation values
$\mathbf{f}^*$	=	prediction vector
F_n	=	cumulative distribution function
$\mathbb{I}$	=	Boolean function
$k(\mathbf{x}, \mathbf{x})$	=	covariance function
l_k	=	lengthscale
λ	=	weight decay
$m(\mathbf{x})$	=	mean function
M	=	Mach number
MSE	=	mean squared error
N_{ind}	=	number of inducing points

[*]PhD Student, Centre for Doctoral Training in Distributed Algorithms, m.anhichem@liverpool.ac.uk, Student Member AIAA.
[†]Reader, School of Engineering, sebastian.timme@liverpool.ac.uk. Member AIAA.
[‡]Computational Scientist, jony.castagna@stfc.ac.uk
[§]Chief Scientist, Senior Member AIAA.
[¶]Technical Consultant.

N_{basis}	=	POD lower-dimensional basis
N_{points}	=	number of points per snapshot
N_{snap}	=	number of snapshots
N	=	number of training points
N_t	=	number of testing points
p	=	probability distribution
q	=	variational probability distribution
R^2	=	coefficient of determination
$\mathcal{S}$	=	snapshot matrix
SE_{95}	=	95th percentile of the squared error
SE_{99}	=	99th percentile of the squared error
T	=	number of Bayesian neural networks (BNN) forward passes
$\mathbf{u}$	=	vector of observation at inducing point locations
$\mathcal{V}$	=	matrix of POD coefficients
$\mathbf{x}$	=	input vector
X, Z	=	Cartesian coordinates
$\mathbf{X}$	=	matrix of training points
y	=	noisy observation scalar function
$\mathbf{y}$	=	noisy observation vector
$\hat{y}$	=	neural network output
$\mathbf{Z}$	=	matrix of inducing points
α	=	angle of attack
ε	=	Gaussian noise random variable
μ_{GP}	=	Gaussian process mean
μ_{MCD}	=	BNN with Monte Carlo dropout mean
$\mathbf{\Psi}$	=	matrix of POD modes
σ	=	output standard deviation
σ_{GP}	=	Gaussian process standard deviation
σ_{MCD}	=	BNN with Monte Carlo dropout standard deviation
τ	=	Gaussian noise variance
$\boldsymbol{\theta}$	=	hyperparameters vector

I. Introduction

Numerous experimental and numerical approaches are available for the analysis of aerodynamic problems. Their extensive use is driven by the need to account for multiple flight conditions and potential design variations. This need frequently occurs in applications such as shape optimisation or uncertainty quantification. Challenges arise as applying these approaches can be costly and time-consuming to execute in such context. As the desired accuracy level of the outcome increases, these difficulties become more pronounced. There is a demand for aerodynamic data models that are both fast and accurate, ensuring the capability to predict viable design alternatives with confidence. Many engineering design problems face similar challenges and aim to overcome these with approximate mathematical models called surrogate models. A surrogate model can be defined as an intermediate agent that mimics the relationship between a system input and its output and has the ability to make cost effective predictions about what the true response might look like in regions of the parameter space where no *a priori* knowledge exists. The emergence of machine learning techniques has paved the way for developing advanced surrogates that can accurately capture the fundamental characteristics of the response of complex engineering systems.

Various surrogate models have been discussed for aerodynamics. Historically, surrogate-based modelling has extensively used a method known as Gaussian process regression, also commonly referred to as kriging. Its effectiveness was demonstrated in flow predictions [1, 2] as well as aerofoil shape optimisation [3, 4]. Also, a significant number of applications within aerodynamics use a combination of the proper orthogonal decomposition with an interpolation method. Examples in compressible aerodynamics, transonic flows or aerodynamic loads evaluation are available in [5–8]. Equally, the growing popularity of deep learning in the machine learning community explains the relatively recent but successful application of neural networks for aerodynamics. Depending on the problem at hand, different types and architectures can be applied to aerodynamic problems. For instance, a multilayer perceptron and a convolutional neural network were used in [9] and [10], respectively, to compute the aerodynamic forces of an aerofoil. Unsteady (multidisciplinary) phenomena, such as flutter speed or buffet pressure loads predictions, were considered in [11–13]. A comparative study on pressure distribution predictions for various geometries and input space dimensions using Gaussian processes, interpolation with proper orthogonal bases and deep neural networks is provided in [14].

As surrogate models are approximation methods, it entails a focus on the quality of their predictions. The accuracy of a surrogate model is defined by the error between the surrogate model estimate and the raw experimental or numerical data [15–17]. Different metrics will be introduced in the following to quantify the model accuracy. However, this does not in any way reflect the accuracy of the data with respect to the real-world phenomena. Indeed, typically experiments and simulations are subject to epistemic and aleatoric uncertainties themselves. In other words, there is a first mismatch between the physics on the one hand and the experimental or numerical data on the other hand, and then a second discrepancy between these (experimental or numerical) data and the surrogate model. This discussion is out of scope for this work, and the only interest here is the epistemic uncertainty associated with the surrogate model with respect to its input data. It is essential to understand to what extent the surrogate model is likely to be wrong with respect to the data. This uncertainty is mostly associated with the fact that the training data set only covers a part of the design space and thus ignores possible variations of the modelled function (experiment or simulation output). The aleatory uncertainty (noise in the data) is considered to be null here. Bayesian methods allow the estimation of the epistemic uncertainty from a surrogate model. Thus, Bayesian surrogate models provide an estimate of both the output and its associated uncertainty. Two Bayesian models are considered in this paper; Gaussian processes, already mentioned earlier, and Bayesian neural networks (BNN). Previously, these were compared in the context of building energy modelling in [18]. BNN is a type of neural network that incorporates Bayesian inference to make probabilistic predictions. Traditional neural networks make deterministic predictions based on fixed weights and biases. In contrast, BNNs allow for uncertainty in the weights and biases by treating them as random variables with a prior probability distribution. BNNs were used to assist the design of pressure tap locations in [19], and they have also shown improvements with respect to their non-Bayesian equivalent not only on benchmark data sets [20] but also in healthcare [21] or for photovoltaic interval prediction [22].

A similar study is carried out here to assess the ability of Bayesian techniques to make fast and uncertainty-aware predictions of pressure distributions on an aerofoil. Noting the large volumes of data generated to study the pressure field, big data incarnations of the algorithms must be considered. Variational inference techniques enable the application of BNNs and Gaussian processes. The objective is to evaluate the modelling accuracy of the two methods and their ability to predict the epistemic uncertainty. Our study presents a comparison by integrating Bayesian surrogate models into the field of aerodynamics, offering options that combine accurate predictive capability with uncertainty quantification for pressure distributions. Section II introduces the two Bayesian surrogate models used in this study, along with an alternative model based on proper orthogonal decomposition (POD) with interpolation. The test case and the data used here are introduced in Section III, specifically the OAT15A aerofoil and Reynolds-averaged Navier–Stokes (RANS) simulations are considered to obtain a suitable data set. Section IV analyses the resulting models with respect to the different accuracy metrics and to the pressure distribution plots.

II. Bayesian Surrogate Models

A. Bayesian Paradigm

Bayesian modelling involves the application of Bayes' theorem to analyse data, which entails updating available knowledge regarding parameters in a model with information obtained from observed data as these become available. Let f be the surrogate model mapping the input vector $\mathbf{x} \in \mathcal{D}$ to the scalar observed output y, where $\mathcal{D} \subseteq \mathbb{R}^d$ is the design space of dimension d. It can be written $y = f(\mathbf{x}, \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is the vector of the model parameters. For instance, in the case of a linear regression, the model parameters are the intercept and the slope, while for deep learning applications these become the weights and biases of the neural network. Instead of having a single value, the parameters are described by probability distributions in Bayesian modelling that contain information explaining the association between $\mathbf{x}$ and y. Based on the prior knowledge of the parameters' distribution $p(\boldsymbol{\theta})$ and on the likelihood function $p(y|\mathbf{x}, \boldsymbol{\theta})$ indicating which parameter values are more likely to explain the observed data, Bayes' theorem is applied to compute the posterior distribution,

$$p(\boldsymbol{\theta}|y, \mathbf{x}) = \frac{p(y|\mathbf{x}, \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(y|\mathbf{x})} \tag{1}$$

where $p(y|\mathbf{x})$ is the marginal likelihood. It is the likelihood function that has been integrated over the parameter space $\boldsymbol{\theta}$. The posterior distribution can then be used to do predictions at unseen inputs $\mathbf{x}^*$ by substituting it into the expression of the predictive distribution,

$$p(y^*|\mathbf{x}^*, \mathbf{x}, y) = \int p(y^*|\mathbf{x}^*, \boldsymbol{\theta})\, p(\boldsymbol{\theta}|y, \mathbf{x})\mathrm{d}\boldsymbol{\theta} \tag{2}$$

Nevertheless, in most cases, the exact posterior distribution is intractable due to the complexity of the models and the amount of data considered. In big data regimes, variational inference is an approach that is frequently considered to approximate the true posterior distribution. The concept involves approximating the true posterior distribution p with a known-form variational distribution q. The objective is then to find the parameters of q that minimises the divergence with p, thus converting the inference issue into an optimisation problem. The divergence between the two probability distributions is quantified by the Kullback–Leibler (KL) divergence. Its reformulation leads to a lower bound of which the optimisation provides the model parameters and the variational parameters. The variational distribution is then used to make predictions replacing the true posterior distribution in the predictive distribution given by Eq. 2. More detailed explanations as well as application examples can be found in [23, 24]. In the two families of techniques considered in this paper, namely Gaussian processes and BNN, different formulations of variational inference are used to enable scalable applications and to obtain posterior distributions.

B. Bayesian Neural Network with Monte Carlo Dropout

Deep learning is a subset of machine learning, inspired by the structure and function of the human brain's neural networks. It provides solutions to complex tasks by automatically learning hierarchical representations of data. Deep neural networks consist of multiple layers (deep layers) of interconnected artificial neurons, which process and transform input data. These networks are trained on large data sets, adjusting their internal parameters through a process known as backpropagation to minimise prediction errors. Despite their popularity, standard neural networks are often prone to overfitting, making their generalisation difficult, and provide no measure of model uncertainty. Bayesian deep learning overcomes these limitations. The Bayesian neural network concept extends the standard network architectures to align with the Bayesian modelling approach that just has been introduced. Traditional neural networks make deterministic predictions based on fixed weights and biases. In contrast, Bayesian neural networks account for uncertainty in the weights and biases by treating them as random variables with a prior probability distribution. During training, the prior distribution is updated to a posterior distribution using Bayes' rule. This updated posterior distribution reflects the uncertainty of the model's parameters given the observed data. This updated posterior can be used to make predictions that incorporate this uncertainty, resulting in a distribution of possible outcomes rather than a single value. As stated before, the posterior distribution over the weights can be obtained using Bayesian inference techniques such as Markov Chain Monte Carlo or variational inference. A detailed mathematical introduction is available in [25] and extensive reviews of the training methods are provided in [26, 27].

One popular variational inference approach introduced in [20] for training BNN is Monte Carlo Dropout, which is a simple and efficient method for approximating the posterior distribution over the network weights. While Monte Carlo Dropout offers computational efficiency and ease of implementation, it may exhibit limitations in capturing complex forms of uncertainty and could pose challenges in achieving well-calibrated uncertainty estimates. A method called dropout regularisation is used during training, where a random subset of neurons is temporarily turned off during each forward pass of an epoch, i.e. a complete pass through the entire training data set. This has the effect of creating an ensemble of models, each with a different subset of active neurons. During the training phase, the parameters of the network, namely the weights and biases $\boldsymbol{\theta}$, are updated through the minimisation of the cost function (or objective function),

$$\mathcal{L}_{\text{dropout}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda||\boldsymbol{\theta}||^2 \tag{3}$$

The first term is the mean square error evaluated on the training data set y of size N and the neural network output $\hat{y}$. The second (penalty) term is an L_2 regularisation weight with a decay λ. During testing, instead of using a single forward pass through the network, multiple forward passes are made with different subsets of neurons dropped out. The outputs of these multiple forward passes are averaged to obtain a model prediction for a test input. They are also used to compute the predictive uncertainty of the model. Assuming a standard neural network with weights and biases $\boldsymbol{\theta}$, the output of the network for a given input $\mathbf{x}$ can be written as $\hat{y}(\mathbf{x}; \boldsymbol{\theta})$. For the BNN model here, the T forward passes made with different dropout masks result in a set of outputs $\{\hat{y}(\mathbf{x}; \boldsymbol{\theta}_t)\}_{t=1}^{T}$. The predictive distribution over the network's outputs is approximated by averaging over this set of outputs,

$$p(y^*|\mathbf{x}^*, \mathbf{x}, y) \approx \frac{1}{T}\sum_{t=1}^{T} p(y^*|\mathbf{x}^*, \boldsymbol{\theta}_t) \tag{4}$$

The mean and variance of the predictive distribution can be approximated by,

$$\mu_{\text{MCD}}(\mathbf{x}) \approx \frac{1}{T}\sum_{t=1}^{T} \hat{y}(\mathbf{x}; \boldsymbol{\theta}_t) \tag{5}$$

$$\sigma^2_{\text{MCD}}(\mathbf{x}) \approx \frac{1}{T-1}\sum_{t=1}^{T} \left(\hat{y}(\mathbf{x}; \boldsymbol{\theta}_t) - \mu_{\text{MCD}}\right)^2 \tag{6}$$

This approach is a computationally efficient technique, as it can be implemented using standard dropout layers in popular deep learning frameworks. The BNN implementation is adapted from the work of [20] * and built on TensorFlow using the Keras API [28]. Its architecture is discussed and justified in Section III.

C. Scalable Gaussian Processes

Gaussian processes are a powerful probabilistic model that can be used for regression [29]. A Gaussian process is characterised by $\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ where m is the mean function and k is the covariance function. The regression outcome strongly depends on the formulation used for $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$. Whereas the mean function is commonly assumed to be zero on the design space, $m(\mathbf{x}) = 0$, the covariance function incorporates the function's structure available under the Gaussian process prior. The covariance function expression will be defined in Section III. A Gaussian process defines a probability distribution over functions written as $f : \mathbf{x} \in \mathcal{D} \longmapsto f(\mathbf{x}) \in \mathbb{R}$, where any finite set of function values has a joint Gaussian distribution. Noisy observations can be considered and the observation variable is then written as $y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon$ with the noise ε being an independent identically distributed Gaussian variable with zero mean and variance τ^2. Given a set of N input points represented by the matrix $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_N]$, a set of corresponding output values $f(\mathbf{X}) = [f_1, f_2, ..., f_N]^\top$ and a matrix of evaluation points $\mathbf{X}^*$, the joint normal distribution over the random variable vectors $\mathbf{y}$ and $\mathbf{f}^* = f(\mathbf{X}^*)$ is given by,

$$p(\mathbf{y}, \mathbf{f}^*) = \mathcal{N}\left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{X}^*) \end{bmatrix}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) + \tau^2\mathbf{I} & k(\mathbf{X}, \mathbf{X}^*) \\ k(\mathbf{X}, \mathbf{X}^*)^\top & k(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix}\right) \tag{7}$$

The predictive distribution for $\mathbf{X}^*$ is derived from the multivariate Gaussian conditional rule which gives $p(\mathbf{f}^*|\mathbf{X}^*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mathbf{f}^*|\mu_{\text{GP}}(\mathbf{X}^*), \sigma_{\text{GP}}(\mathbf{X}^*))$ where:

$$\mu_{\text{GP}}(\mathbf{X}^*) = m(\mathbf{X}^*) + k(\mathbf{X}, \mathbf{X}^*)^\top \left[k(\mathbf{X}, \mathbf{X}) + \tau^2\mathbf{I}\right]^{-1} \left(\mathbf{y} - m(\mathbf{X})\right) \tag{8}$$

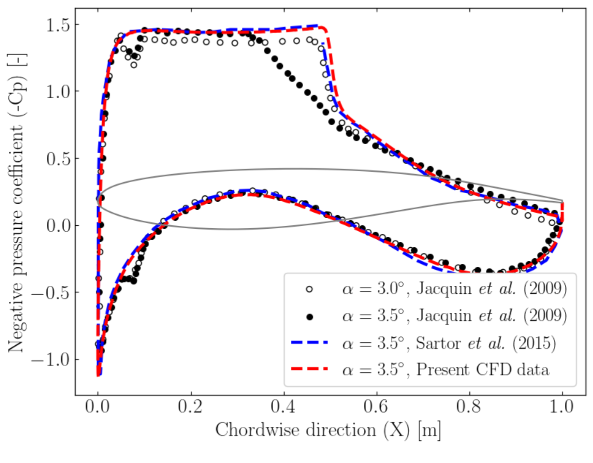
$$\sigma_{\text{GP}}(\mathbf{X}^*) = k(\mathbf{X}^*, \mathbf{X}^*) - k(\mathbf{X}, \mathbf{X}^*)^\top \left[k(\mathbf{X}, \mathbf{X}) + \tau^2\mathbf{I}\right]^{-1} k(\mathbf{X}, \mathbf{X}^*) \tag{9}$$

Training the model corresponds to finding an appropriate set of covariance function parameters, also called the model hyperparameters, similarly denoted by the vector $\boldsymbol{\theta}$. A commonly employed technique for hyperparameter learning involves maximising the likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ with gradient-based algorithms. By using the standard form of a multivariate Gaussian distribution, the log-likelihood is expressed as:

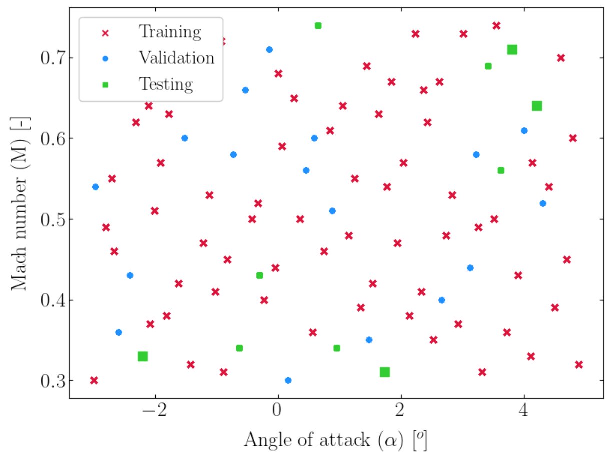
$$\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^\top [k(\mathbf{X}, \mathbf{X}) + \tau^2\mathbf{I}]^{-1}\mathbf{y} - \frac{1}{2}\log|k(\mathbf{X}, \mathbf{X}) + \tau^2\mathbf{I}| - \frac{N}{2}\log(2\pi) \tag{10}$$

Observing the functional form in Eqs. (8) through (10), it can be seen that the regression model involves the matrix inversion of $\left[k(\mathbf{X}, \mathbf{X}) + \tau^2\mathbf{I}\right]$. The number of operations for direct matrix inversion scales with N^3 operations. For large data sets, it will lead to scalability limits of the Gaussian process regression and requires additional consideration. Stochastic Variational inference for Gaussian Processes (SVGP) is used herein to enable the application to large data sets. SVGP is an approximate inference method that uses a small set of inducing points to approximate the true posterior

*http://www.github.com/yaringal/DropoutUncertaintyExps



(a)



(b)

Fig. 1 (a) Pressure coefficient results at $M = 0.73$ compared with experiments from [38] and steady-state two-dimensional aerofoil results from [39] and (b) and design of experiment for present OAT15A test case.

distribution over functions. This reduced set of latent variables with a size $N_{\text{ind}} \ll N$ represents the actual data set $\mathbf{X}$. These N_{ind} inducing points are evaluations $\mathbf{u} = f(\mathbf{Z})$ at the inducing point locations $\mathbf{Z}$. The inducing points are chosen to maximise a lower bound on the log marginal likelihood of the data, and the posterior distribution is approximated by a Gaussian distribution with a mean and covariance matrix that depend on the inducing points. More specifically, during the training of the SVGP, the inducing point locations $\mathbf{Z}$ and the covariance parameters $\boldsymbol{\theta}$ are optimised to minimise the KL divergence. The predictive distribution for a new point $\mathbf{X}^*$ can then be computed efficiently using the inducing points and their associated covariance matrix. This method was introduced in [30, 31] and applied to multifidelity data fusion of pressure distributions in [32]. A more general review of scalable Gaussian processes is presented in [33].

D. Proper Orthogonal Decomposition with Interpolation

To assess the BNN and SVGP approaches, we introduce a technique known in the literature as POD with interpolation, which amalgamates the computation of POD modes and their coefficients with the interpolation of these coefficients to generate new predictions [7, 14, 34–36]. Define N_{snap} as the number of snapshots (in this context, representing the number of flow conditions) and N_{points} as the number of points per snapshot (here, indicating the number of surface mesh points). The data can be organised into an $N_{\text{points}} \times N_{\text{snap}}$ matrix denoted as $\mathcal{S}$. Given this snapshot matrix, the POD method aims to identify a lower-dimensional basis ($N_{\text{basis}} < N_{\text{snap}}$), capable of effectively approximating those same snapshots. In most scenarios, the snapshot matrix has a tall and narrow structure ($N_{\text{snap}} \ll N_{\text{points}}$), signifying the presence of numerous degrees of freedom compared to the number of snapshots. Rather than directly applying singular value decomposition (SVD), which is computationally expensive, the method of snapshots [37] can be employed (working on $\mathcal{S}^T\mathcal{S}$ instead).

Thus, the matrix $\mathcal{S}$ can be expressed as a linear combination of the POD modes Ψ,

$$\mathcal{S} = \Psi\mathcal{V} \tag{11}$$

where $\mathcal{V}$ represents the resulting POD coefficients organised in an $N_{\text{snap}} \times N_{\text{snap}}$ matrix. More specifically, each individual snapshot can be reconstructed as a linear combination of the POD modes, denoted as $\mathcal{S}_i = \Psi\mathcal{V}_i$, where $\mathcal{S}_i$ and $\mathcal{V}_i$ correspond to the i-th column of $\mathcal{S}$ and $\mathcal{V}$, respectively. To make predictions in the design space (flow conditions) that are not part of the snapshot matrix, it is essential to determine the coefficients associated with each mode. The concept behind POD with interpolation is to select a limited number of POD modes, which capture the majority of the information, and then perform interpolation using the information contained in $\mathcal{V}$ to calculate their corresponding POD coefficients. Herein, a Gaussian process is trained for ten modes by employing the rows of $\mathcal{V}$ as training data, meaning that ten Gaussian processes are trained, with each one using the i-th row of $\mathcal{V}$ to train the Gaussian process associated with the i-th POD mode.

Table 1 Bayesian neural network hyperparameters.

Parameter	Range	Optimal value
Number of hidden layers	$\{1,2,...,12\}$	3
Number of neurons	$\{32k \mid k \in \{1,...,32\}\}$	384 (for $k = 12$)
Shrinkage rate	[0.25, 1.0]	0.661
Dropout rate	[1%, 5%]	1%

III. Problem Setup

A. OAT15A Aerofoil Numerical Simulations

ONERA's OAT15A profile has been studied widely since its first experimental investigation in [38], such as for numerical shock-buffet stability analyses in [40, 41]. To solve the non-linear governing equations, the TAU code of the German Aerospace Center (DLR), commonly used in the Eruopean aerospace industry, was chosen. It employs a cell-vertex second-order finite-volume spatial discretisation. The inviscid fluxes of the governing RANS equation are discretised via a central scheme that employs matrix artificial dissipation, whereas a first-order Roe upwind scheme is used for those of the turbulence model. For the computation of gradients of flow variables, required for viscous fluxes and the turbulence model's source term, the Green–Gauss theorem is employed. The no-slip adiabatic condition is strongly enforced on the solid walls of the aerofoil, and the far field is considered to be a free-stream flow through a characteristic boundary condition. The code's standard backward Euler scheme with the lower-upper symmetric Gauss–Seidel method, together with local time-stepping and multigrid for convergence acceleration, is chosen for obtaining steady-state solutions converged by 7 orders of magnitudes in the density residual norm.

For the two-dimensional aerofoil simulations herein, a baseline mesh was generated that has a quasi-structured region in the near-field with an O-type topology around the blunt trailing edge, while unstructured triangular cells are employed towards the circular far-field boundary with a radius of 100 chord lengths. The domain is discretised with approximately $80,000$ control volumes altogether, and the aerofoil surface is divided into 626 elements. Figure 1a illustrates the pressure coefficient in comparison to corresponding experimental data from [38] at two specific angles of attack, specifically around the onset of shock buffet. Additionally, it includes numerical results for the same two-dimensional aerofoil obtained from [39], using a very similar version of the turbulence model. The simulations at $\alpha = 3.5°$ demonstrate good agreement with the wind-tunnel data at $\alpha = 3.0°$. This discrepancy is often reported and attributed to the particular choice of turbulence model.

Along with the aerofoil surface coordinates (X, Z), the Mach number M and angle of attack α are considered as input space parameters. Following a Halton sequence [42], as outlined in [14], 98 steady-state simulations have been carried out within the parameter ranges $M = [0.3, 0.75]$ and $\alpha = [-3°, 5°]$, leading to a total of 61,348 data points. Figure 1b shows the sample locations of the numerical experiments which are divided into three subsets; specifically 70% for training, 20% for validation and 10% for testing.

B. Model Design

The selection of the appropriate structure and parameters for the two models (BNN and SVGP) is outlined next. The idea is to justify the design choices of the surrogate models that can learn a meaningful representation of the input data by making accurate predictions on validation data points. The architecture of a neural network, on the one hand, is mainly characterised by the number of hidden layers and the number of neurons in the input, hidden and output layers. The BNN considered here, and represented as an example in Fig. 2, requires in addition the definition of a dropout rate (ratio of neurons turned off during training), a shrinkage factor (ratio of neurons between two successive layers) and a batch size (number of training points presented to the model in one forward/backward pass). Selecting an appropriate set of hyperparameters is critical in achieving satisfactory predictive performance of the model. Table 1 summarises the parameter ranges considered (except the batch size which is fixed at 1024 throughout after initial testing). It can be noticed that the number of neurons is a multiple of 32. Indeed, when working with CUDA (NVIDIA's platform for GPU-accelerated parallel computing) and GPUs for deep learning tasks, it is a common best practice to choose hidden layer and batch sizes that are multiples of 32. This practice helps in fully leveraging the parallel processing capability and optimising the performance of neural network training and inference.

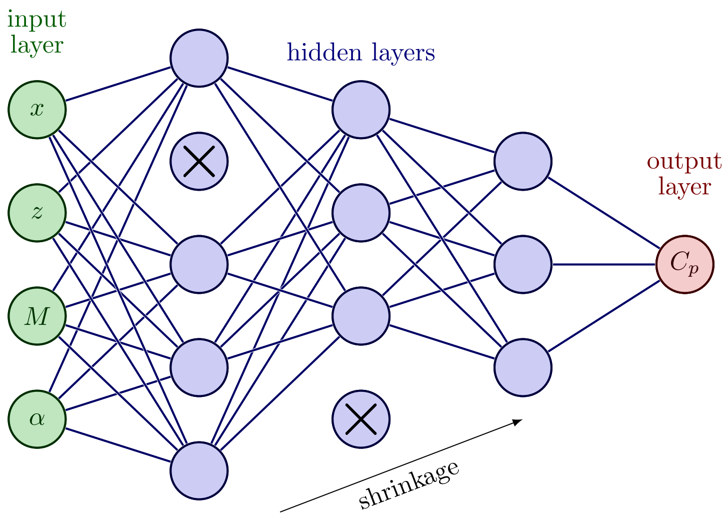
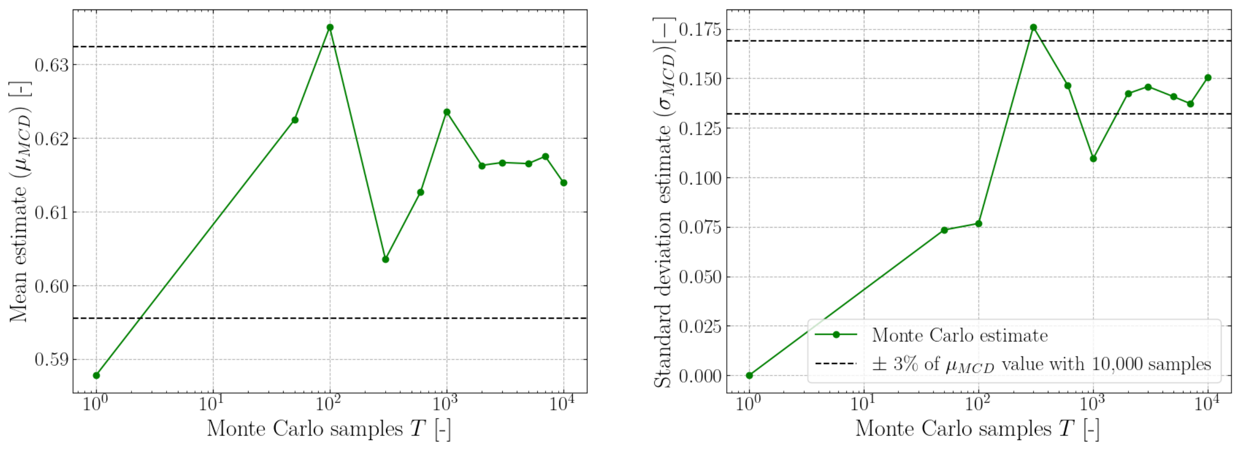


Fig. 2 Bayesian neural network with Monte Carlo dropout architecture example.

Bayesian optimisation is used to search the best model hyperparameters [43]. Instead of exhaustively trying all possible hyperparameter combinations, Bayesian optimisation uses probabilistic models to predict which configurations are most likely to yield better performance based on the network's previous evaluations. It iteratively refines its predictions, directing the search towards promising regions of the hyperparameter space, ultimately leading to improved neural network performance while requiring fewer trials compared to grid or random search. The optimisation is made on the neural network's cost function (here the mean-squared error defined below) evaluated on the validation set. A Gaussian process regression of the (negative) cost function is built describing its variation on the space of the hyperparameters. An acquisition function, herein the upper confidence bound defined as the sum of the posterior mean and standard deviation weighted by a coefficient, is then computed to find a next evaluation point on this space, specifically where the acquisition function is maximal. The coefficient weighting the standard deviation in the acquisition function is used to tune the exploitation vs exploration trade-off. When the coefficient is small, Bayesian optimisation prioritises solutions expected to perform well, characterised by a high posterior mean (i.e. low value of the cost function). Conversely, with a large coefficient, Bayesian optimisation encourages exploration of previously unexplored regions within the search space (i.e. high standard deviation of the cost function). Here, the Bayesian optimisation application used a default value of 2.6. A total of 250 combinations are computed and tested with respect to the validation set. The best combination is also shown in Table 1. It is essential to highlight that Bayesian optimisation is primarily concerned with identifying the model with the highest accuracy (i.e. lowest mean-squared error) on the validation data within predefined parameter ranges. However, it does not account for the assessment of output uncertainty, which can be viewed as a limitation when working with Bayesian surrogate models.

The final BNN architecture, acquired through Bayesian optimisation, undergoes training for 1000 epochs using the Adam optimiser [44] applied with the default learning rate of 1×10^{-3} on the mean-squared error. It is performed on a laptop with an Intel Core i7-11800H processor and an Nvidia RTX A2000 4GB Graphics Processing Unit (GPU). Then, the predictive distribution is approximated by computing T forward passes through the trained network with various dropout masks. Convergence of μ_{MCD} and σ_{MCD} predicted by this BNN is demonstrated in Fig. 3 with an increasing number of Monte Carlo samples. Both are computed on the validation set. It can be observed that they both converge to within a 3% range of μ_{MCD} after approximately 1000 samples.

For SVGP, on the other hand, the library GPflow was adopted, which is a Python module based on TensorFlow [28, 45]. Enhanced with GPU enabled capability, SVGP training makes use of a dedicated compute node having 48 AMD CPU cores with 256 GB of memory and equipped with two Nvidia Ampere A100 80 GB GPUs. The model is trained using 10,000 inducing points, 10,000 iterations of the Adam optimiser, and mini-batches of 100 samples. As the observations are deterministic, noise-free predictions are considered. Nevertheless, the noise term τ is fixed to a small value of 10^{-4} to ensure numerical stability. Two popular choices of covariance functions have been considered initially, specifically



(a) BNN mean **(b) BNN standard deviation**

Fig. 3 Analysis of the convergence of BNN estimations as the number of Monte Carlo dropout samples increases.

squared-exponential (SE) and Matern3/2 (M32), defined as,

$$k_{SE}(\mathbf{x},\mathbf{x}') = \sigma^2 \exp\left(-\sum_{k=1}^{d}\frac{(x_k - x'_k)^2}{2l_k^2}\right) \tag{12}$$

$$k_{M32}(\mathbf{x},\mathbf{x}') = \sigma^2\left(1+\sum_{k=1}^{d}\frac{\sqrt{3}|x_k - x'_k|}{l_k}\right)\exp\left(-\sum_{k=1}^{d}\frac{\sqrt{3}|x_k - x'_k|}{l_k}\right) \tag{13}$$

where the model hyperparameters are the output standard deviation σ and a length scale l_k per dimension, denoted by the vector $\boldsymbol{\theta} = [\sigma, l_k]$, previously defined. The length scale for each dimension of the design space was found through automatic relevance determination [46]. The highest accuracy on the validation set has been found with the Matern3/2 covariance function k_{M32}, which will therefore be discussed hereafter.

C. Evaluation criteria

In the context of Bayesian surrogate models, these models yield two essential quantities; a prediction made by the model and the corresponding measure of uncertainty. We introduce metrics for assessing both the accuracy of the model's predictions and the accuracy of its uncertainty estimates. To comprehensively evaluate the performance of these models, we consider the mean-squared error (MSE) and the coefficient of determination (R^2),

$$\text{MSE} = \frac{1}{N_t}\sum_{n=1}^{N_t}(y_n - \hat{y}_n)^2 \tag{14}$$

$$R^2 = 1 - \frac{\sum_{n=1}^{N_t}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N_t}(y_n - \bar{y})^2} \tag{15}$$

where $\bar{y}$ represents the mean of y and N_t is the number of testing points. For MSE, lower values are desirable, indicating a better fit while for R^2, a value closer to 1 suggests a higher proportion of explained variance. Additionally, we assess the 95th and 99th percentiles of the squared errors, denoted as SE_{95} and SE_{99}, respectively. Percentiles of squared error provide insights into the distribution of errors in a model. For example, the 95th percentile indicates the value below which 95% of squared errors fall, offering a glimpse into the robustness of the model. While accurate predictions are a crucial aspect of surrogate models, Bayesian models introduce the additional necessity of evaluating the reliability of uncertainty estimates. In a properly calibrated Bayesian model, these uncertainty estimates should accurately encapsulate the genuine data distribution, meaning that, for instance, a 90% posterior confidence interval should encompass the true simulation outcome in approximately 90% of cases. In the context of regression [47, 48], the surrogate model's

Table 2 Evaluation metrics of Bayesian models on test set.

Model	MSE	R^2	SE_{95}	SE_{99}
BNN	**0.0029**	**0.9900**	**0.0064**	**0.0563**
SVGP	0.0164	0.9441	0.0724	0.1906
POD+GP	0.0156	0.9467	0.0733	0.3086

uncertainty estimates are deemed well-calibrated when

$$\frac{1}{N_t}\sum_{n=1}^{N_t}\mathbb{I}\{y_n \leq F_n^{-1}(p)\} \rightarrow p \text{ for all } p \in [0,1] \tag{16}$$

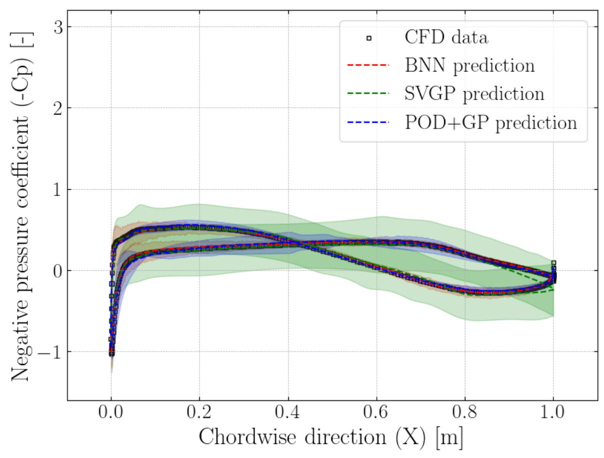
where $\mathbb{I}$ is the Boolean function, F_n is the cumulative distribution function targeting y_n. Here, F_n^{-1} denotes the quantile function $F_n^{-1}(p) = \inf\{y : p \leq F_n(y)\}$. In simpler terms, as the data set size approaches infinity, the empirical and predicted cumulative distribution functions should align. To evaluate the quality of calibration, we determine the proportion of test data observations that lie within the prediction confidence intervals obtained from the quantile function. We visualise the degree of calibration in Bayesian models using a calibration plot, where perfectly calibrated uncertainty estimates would align precisely with the diagonal line.

IV. Results

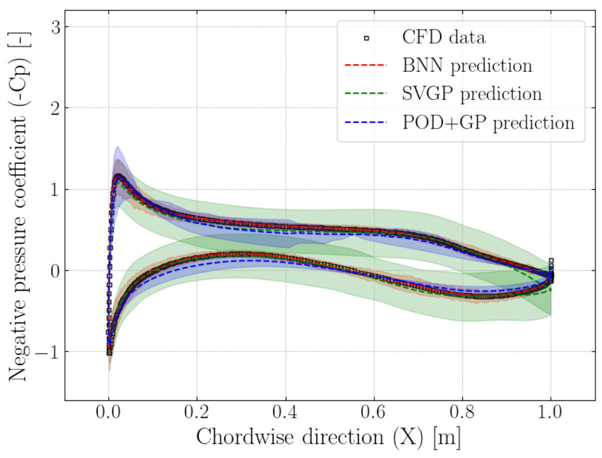
All three Bayesian models were implemented and applied for the purpose of studying their ability on a transonic flow problem. The POD with interpolation model is used as a reference model from the existing literature. We scrutinise the models with respect to their prediction accuracy and uncertainty approximation. Table 2 summarises the results for the metrics, defined in Section III.C, for the OAT15A test set. Predictions on the test set have been carried out to evaluate the models' accuracy on unseen data. The results reveal differences in performance among the methods. The BNN demonstrates an approximately ten-fold reduction in MSE, showing its better capability to closely approximate the correct values. Moreover, with a higher R^2 score, the BNN exhibits a stronger fit compared to the other methods. With respect to these two metrics, the POD coupled with Gaussian process regression (POD+GP) presents slightly better results than SVGP. However, in terms of robustness, SVGP has lower values of SE_{99} and the BNN model exhibits the highest performance, showcasing one-order-of-magnitude lower values in both SE_{95} and SE_{99}.

Figure 4 present a comparison of the predictions at fixed flow conditions from the Bayesian surrogate models. The predictions (compared with reference data) show the posterior mean and the confidence intervals associated with the posterior standard deviations ($\pm 1.96\sigma$) superimposed. It can be observed in Figs. 4a and 4b that both models under scrutiny provide reliable predictions for relatively low values of Mach number and angle of attack. They visually provide a closer match to the simulation data compared to the POD+GP model for these two flow conditions. At higher Mach numbers, as plotted in Figs. 4c and 4d, the prediction outputs are less accurate due to the existence of non-linearities inherent to transonic flows, specifically shock waves, and all models behave differently. The BNN surrogate captures the chordwise location of the shock as well as the pressure values of the supersonic part of the flow reasonably well, but the sharpness of the shock is underestimated resulting in a small difference with the simulation data. For the SVGP and POD+GP models, the prediction at these locations is clearly degraded. Indeed, the appearance of the pressure coefficient is smooth (linear), therefore not following the abrupt change due to the flow discontinuity. At the same time, all models tend to yield relatively accurate predictions for the lower surface of the wing. It can be asserted with confidence that the varying performance, reported in Table 2, can be attributed to differences in their capabilities to predict transonic flow characteristics.

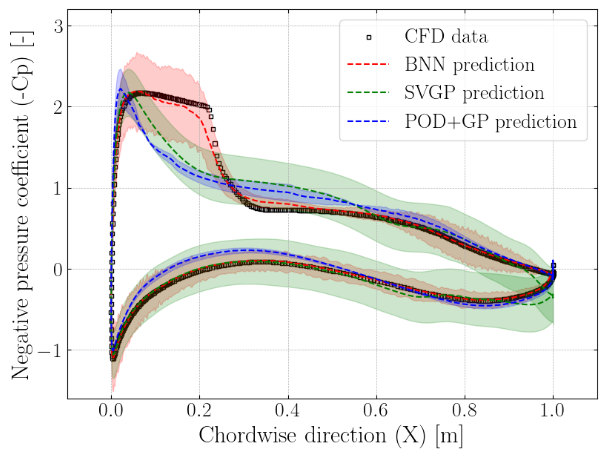
The advantage of using Bayesian models is also to obtain an estimate of the epistemic uncertainty plotted as transparent regions in Fig. 4, which gives us confidence in the predicted value away from the training points. As discussed in Section II, the methods from Section II.B and Section II.C are based on variational inference techniques. They are only approximations to the true posterior distributions and may not always capture the full uncertainty of the model. It is therefore necessary to analyse the uncertainties obtained through the standard deviation of the methods. In the figures, it can be seen that similar and satisfactory levels of confidence intervals are predicted with the exception of



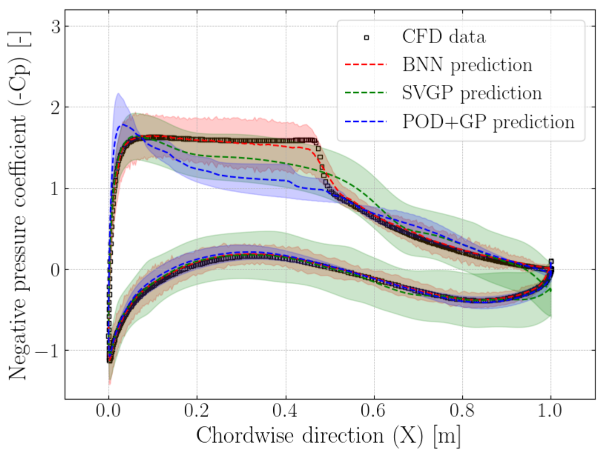
(a) α = −2.21°, M = 0.33



(b) α = 1.74°, M = 0.31



(c) α = 4.21°, M = 0.64



(d) α = 3.81°, M = 0.71

Fig. 4 A comparison between the predictions made by the models on a few points of the test set.

the supersonic region. For the SVGP model, the confidence interval is nearly constant along the chordwise direction on both lower and upper surfaces, whereas the BNN provides wider confidence intervals, particulary in the supersonic part of the flow, indicating a lack of confidence in the given prediction. For each experimental conditions in the training set, the critical pressure coefficient is computed and compared to the associated pressure coefficient distribution. The existence of a supersonic pocket is signalled if a negative pressure coefficient is less than the critical pressure coefficient. If a negative pressure coefficient is lower than the critical pressure coefficient, it indicates the presence of a supersonic pocket. In the training set, it is observed for five experimental conditions (located at the top right of Fig. 1b). Hence, among the sixty-eight conditions considered in the training set, only five demonstrate supersonic features. This observation underscores the difficulties in accurately predicting these features, contributing to the wider confidence intervals evident in the results depicted in Figs. 4c and 4d.

For a more rigorous uncertainty analysis, a calibration plot is shown in Fig. 5 in accordance with the definition from [47] as presented in Section III.C. Uncertainty is assessed by plotting the expected probabilities of observing an outcome against the empirically observed rates within a set of twenty intervals. A similar behaviour is noticeable in both SVGP and BNN, with their respective curves exhibiting an overestimation of confidence levels. Among these models, SVGP comes closest to achieving perfect calibration. The reference model (POD+GP) tends to underestimate confidence levels, as illustrated in Fig. 4, where the confidence intervals appear relatively narrow. While SVGP gives the best overall calibration, Fig. 4 reveals that its confidence intervals remain relatively uniform around the supersonic region, where the model's accuracy is compromised. In contrast, BNN displays an opposing behaviour, which is advantageous given that the primary aim of employing these models is to pinpoint areas where the model lacks certainty

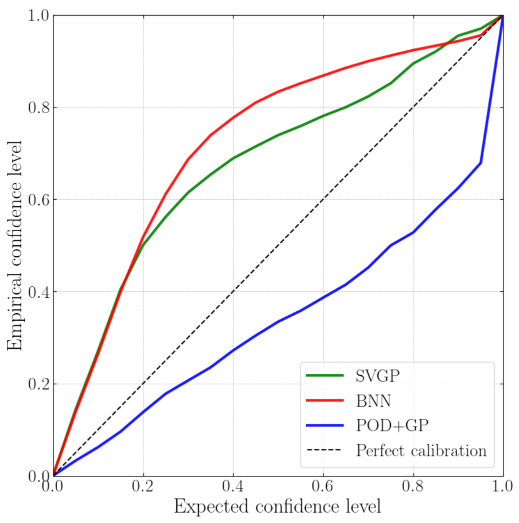


Fig. 5 Calibration plot assessing the different model calibrations.

in its predictions. Approaches for handling uncertainty play a pivotal role in the construction of precise and dependable machine learning models. Bayesian techniques offer a foundation for quantifying uncertainty. Nonetheless, due to the adoption of approximate inference, Bayesian uncertainty estimates can frequently lack accuracy.

V. Conclusions

While surrogate modelling approaches have been widely employed in aerospace engineering, this work proposes an innovative viewpoint by adopting Bayesian surrogates. Through the quantification of surrogate model (epistemic) uncertainty, the Bayesian approach recognises that surrogate models are imperfect approximations of the original physical models (simulations or experiments). This framework provides a valuable means to make informed decisions while accounting for the associated uncertainty. It allows us to leverage the significantly faster execution of surrogate models to generate engineering quantity of interest estimates, all while acknowledging the inherent approximations.

Through a comprehensive discussion of Bayesian models for pressure distribution modelling, three approaches are compared; BNN, SVGP and POD+GP. The analysis of accuracy is based on the evaluation of different performance metrics on a test set. Given that uncertainty is also a fundamental element of Bayesian models, their calibration is a subject of investigation as well. BNN using Monte-Carlo dropout demonstrates the most promising outcomes in terms of accuracy, showcasing the desired ability to predict the presence of shock waves in previously unseen data points. The other two models have demonstrated lower effectiveness on flow conditions that exhibit such compressibility effects. Concerning uncertainty calibration, both BNN and SVGP tend to overestimate the confidence intervals, while POD+GP underestimates them. In other words, the BNN and SVGP predict a higher level of uncertainty than is actually evident in the data. In future work, a calibration procedure for enhancing the calibration of any regression algorithm, based on the work in [47], could be implemented in this scenario to attain improved model calibration. In addition, the presented Bayesian surrogate models will be applied to higher dimensional, practical aerospace problems with more complex geometry. Specifically, previously we compiled a suitable (experimental and numerical) data set for a half wing-fuselage configuration representative of a commercial aircraft design [32, 49] that can be readily assessed.

Acknowledgements

The first author is grateful for the financial support by the Engineering and Physical Sciences Research Council (EPSRC) Centre for Doctoral Training in Distributed Algorithms (grant number 2447391 as part of EP/S023445/1) in partnership with Aircraft Research Association Ltd. We thank the German Aerospace Center for access to the TAU flow solver and the University of Liverpool for computing time on the high-performance computing system. We also acknowledge the technical support provided by the UK Research and Innovation (UKRI) Science and Technology Facilities Council (STFC) Hartree centre. The simulation data that support the findings of this study are available from the authors upon reasonable request.

References

[1] Chiplunkar, A., Bosco, E., and Morlier, J., “Gaussian Process for Aerodynamic Pressures Prediction in Fast Fluid Structure Interaction Simulations,” *12th World Congress on Structural and Multidisciplinary Optimization*, Braunschweig, Germany, 2017, pp. 221–233. URL `https://hal.science/hal-01828716`.

[2] Rajaram, D., Puranik, T. G., Renganathan, S. A., Sung, W., Fischer, O. P., Mavris, D., and Ramamurthy, A., “Empirical Assessment of Deep Gaussian Process Surrogate Models for Engineering Problems,” *Journal of Aircraft*, Vol. 58, No. 1, 2020, pp. 1–15. https://doi.org/10.2514/1.C036026.

[3] Forrester, A. I., and Keane, A. J., “Recent advances in surrogate-based optimization,” *Progress in Aerospace Sciences*, Vol. 45, No. 1-3, 2009, pp. 50–79. https://doi.org/10.1016/j.paerosci.2008.11.001.

[4] Han, Z.-H., Zhang, Y., Song, C.-X., and Zhang, K.-S., “Weighted Gradient-Enhanced Kriging for High-Dimensional Surrogate Modeling and Design Optimization,” *AIAA Journal*, Vol. 55, No. 12, 2017, pp. 4330–4346. https://doi.org/10.2514/1.J055842.

[5] Bui-Thanh, T., Damodaran, M., and Willcox, K., “Proper Orthogonal Decomposition Extensions for Parametric Applications in Compressible Aerodynamics,” *21st AIAA Applied Aerodynamics Conference*, American Institute of Aeronautics and Astronautics, Orlando, Florida, 2003. https://doi.org/10.2514/6.2003-4213.

[6] Amsallem, D., and Farhat, C., “Interpolation Method for Adapting Reduced- Order Models and Application to Aeroelasticity,” *AIAA Journal*, Vol. 46, No. 7, 2008, pp. 1803–1813. https://doi.org/10.2514/1.35374.

[7] Fossati, M., “Evaluation of Aerodynamic Loads via Reduced-Order Methodology,” *AIAA Journal*, Vol. 53, No. 8, 2015, pp. 2389–2405. https://doi.org/10.2514/1.J053755.

[8] Ripepi, M., Verveld, M. J., Karcher, N. W., Franz, T., Abu-Zurayk, M., Görtz, S., and Kier, T. M., “Reduced-order models for aerodynamic applications, loads and MDO,” *CEAS Aeronautical Journal*, Vol. 9, No. 1, 2018, pp. 171–193. https://doi.org/10.1007/s13272-018-0283-6.

[9] Moin, H., Zeeshan Iqbal Khan, H., Mobeen, S., and Riaz, J., “Airfoil’s Aerodynamic Coefficients Prediction using Artificial Neural Network,” *2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, IEEE, Islamabad, Pakistan, 2022, pp. 175–182. https://doi.org/10.1109/IBCAST54850.2022.9990112.

[10] Zhang, Y., Sung, W. J., and Mavris, D. N., “Application of Convolutional Neural Network to Predict Airfoil Lift Coefficient,” *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, Kissimmee, Florida, 2018. https://doi.org/10.2514/6.2018-1903.

[11] Wang, Y.-R., and Wang, Y.-J., “Flutter speed prediction by using deep learning,” *Advances in Mechanical Engineering*, Vol. 13, No. 11, 2021, p. 168781402110622. https://doi.org/10.1177/16878140211062275.

[12] Tekaslan, H. E., Demiroglu, Y., and Nikbay, M., “Surrogate Unsteady Aerodynamic Modeling with Autoencoders and LSTM Networks,” *AIAA SCITECH 2022 Forum*, American Institute of Aeronautics and Astronautics, San Diego, CA & Virtual, 2022. https://doi.org/10.2514/6.2022-0508.

[13] Zahn, R., Weiner, A., and Breitsamter, C., “Prediction of wing buffet pressure loads using a convolutional and recurrent neural network framework,” *CEAS Aeronautical Journal*, 2023. https://doi.org/10.1007/s13272-023-00641-6.

[14] Sabater, C., Stürmer, P., and Bekemeyer, P., “Fast Predictions of Aircraft Aerodynamics Using Deep-Learning Techniques,” *AIAA Journal*, Vol. 60, No. 9, 2022, pp. 5249–5261. https://doi.org/10.2514/1.J061234.

[15] Kleijnen, J. P., “Regression and Kriging metamodels with their experimental designs in simulation: A review,” *European Journal of Operational Research*, Vol. 256, No. 1, 2017, pp. 1–16. https://doi.org/10.1016/j.ejor.2016.06.041.

[16] Yondo, R., Andrés, E., and Valero, E., “A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses,” *Progress in Aerospace Sciences*, Vol. 96, 2018, pp. 23–61. https://doi.org/10.1016/j.paerosci.2017.11.003.

[17] Sun, G., and Wang, S., “A review of the artificial neural network surrogate modeling in aerodynamic design,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 16, 2019, pp. 5863–5872. https://doi.org/10.1177/0954410019864485.

[18] Westermann, P., and Evins, R., “Using Bayesian deep learning approaches for uncertainty-aware building energy surrogate models,” *Energy and AI*, Vol. 3, 2021, p. 100039. https://doi.org/10.1016/j.egyai.2020.100039.

[19] Shen, W., Huan, X., Zhou, B. Y., and Gauger, N. R., "Towards Design of Airfoil Pressure Tap Locations for Real-Time Predictions Under Uncertainty Using Bayesian Neural Networks," *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics, Orlando, FL, 2020. https://doi.org/10.2514/6.2020-0906.

[20] Gal, Y., and Ghahramani, Z., "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," *Proceedings of The 33rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 48, edited by M. F. Balcan and K. Q. Weinberger, PMLR, New York, New York, USA, 2016, pp. 1050–1059.

[21] Filos, A., Farquhar, S., Gomez, A. N., Rudner, T. G. J., Kenton, Z., Smith, L., Alizadeh, M., de Kroon, A., and Gal, Y., "A Systematic Comparison of Bayesian Deep Learning Robustness in Diabetic Retinopathy Tasks," , 2019.

[22] Wang, K., Du, H., Jia, R., and Jia, H., "Performance Comparison of Bayesian Deep Learning Model and Traditional Bayesian Neural Network in Short-Term PV Interval Prediction," *Sustainability*, Vol. 14, No. 19, 2022, p. 12683. https://doi.org/10.3390/su141912683.

[23] Wainwright, M. J., and Jordan, M. I., "Graphical Models, Exponential Families, and Variational Inference," *Foundations and Trends® in Machine Learning*, Vol. 1, No. 1–2, 2007, pp. 1–305. https://doi.org/10.1561/2200000001.

[24] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D., "Variational Inference: A Review for Statisticians," *Journal of the American Statistical Association*, Vol. 112, No. 518, 2017, pp. 859–877. https://doi.org/10.1080/01621459.2017.1285773.

[25] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D., "Weight Uncertainty in Neural Network," *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 37, edited by F. Bach and D. Blei, PMLR, Lille, France, 2015, pp. 1613–1622.

[26] Jospin, L. V., Buntine, W. L., Boussaïd, F., Laga, H., and Bennamoun, M., "Hands-on Bayesian Neural Networks - a Tutorial for Deep Learning Users," *CoRR*, Vol. abs/2007.06823, 2020.

[27] Magris, M., and Iosifidis, A., "Bayesian learning for neural networks: an algorithmic survey," *Artificial Intelligence Review*, Vol. 56, 2023. https://doi.org/10.1007/s10462-023-10443-1.

[28] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," , 2015. Software available from https://www.tensorflow.org/.

[29] Rasmussen, C., and Williams, C., *Gaussian Processes for Machine Learning*, Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, 2006.

[30] Hensman, J., Fusi, N., and Lawrence, N. D., "Gaussian Processes for Big Data," *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, AUAI Press, Arlington, VA, 2013, p. 282–290.

[31] Hensman, J., Matthews, A., and Ghahramani, Z., "Scalable variational Gaussian process classification," *Artificial Intelligence and Statistics*, PMLR, 2015, pp. 351–360.

[32] Anhichem, M., Timme, S., Castagna, J., Peace, A., and Maina, M., "Multifidelity Data Fusion Applied to Aircraft Wing Pressure Distribution," *AIAA AVIATION 2022 Forum*, American Institute of Aeronautics and Astronautics, Chicago, IL & Virtual, 2022. https://doi.org/10.2514/6.2022-3526.

[33] Liu, H., Ong, Y.-S., Shen, X., and Cai, J., "When Gaussian Process Meets Big Data: A Review of Scalable GPs," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 11, 2020, pp. 4405–4423. https://doi.org/10.1109/TNNLS.2019.2957109.

[34] Zimmermann, R., and Görtz, S., "Improved extrapolation of steady turbulent aerodynamics using a non-linear POD-based reduced order model," *The Aeronautical Journal*, Vol. 116, No. 1184, 2012, pp. 1079–1100. https://doi.org/10.1017/S0001924000007491.

[35] Fossati, M., and Habashi, W. G., "Multiparameter analysis of aero-icing problems using proper orthogonal decomposition and multidimensional interpolation," *AIAA Journal*, Vol. 51, No. 4, 2013, pp. 946–960. https://doi.org/10.2514/1.J051877.

[36] Zimmermann, R., Vendl, A., and Görtz, S., "Reduced-order modeling of steady flows subject to aerodynamic constraints," *AIAA Journal*, Vol. 52, No. 2, 2014, pp. 255–266. https://doi.org/10.2514/1.J052208.

[37] Sirovich, L., "Turbulence and the dynamics of coherent structures. I. Coherent structures," *Quarterly of Applied Mathematics*, Vol. 45, No. 3, 1987, pp. 561–571. https://doi.org/10.1090/qam/910462.

[38] Jacquin, L., Molton, P., Deck, S., Maury, B., and Soulevant, D., "Experimental Study of Shock Oscillation over a Transonic Supercritical Profile," *AIAA Journal*, Vol. 47, No. 9, 2009, pp. 1985–1994. https://doi.org/10.2514/1.30190.

[39] Sartor, F., Mettot, C., and Sipp, D., "Stability, receptivity, and sensitivity analyses of buffeting transonic flow over a profile," *AIAA Journal*, Vol. 53, No. 7, 2015, pp. 1980–1993. https://doi.org/10.2514/1.J053588.

[40] Crouch, J. D., Garbaruk, A., and Strelets, M., "Global instability in the onset of transonic-wing buffet," *Journal of Fluid Mechanics*, Vol. 881, 2019, p. 3–22. https://doi.org/10.1017/jfm.2019.748.

[41] He, W., and Timme, S., "Triglobal infinite-wing shock-buffet study," *Journal of Fluid Mechanics*, Vol. 925, 2021, p. A27. https://doi.org/10.1017/jfm.2021.678.

[42] Halton, J. H., "Algorithm 247: Radical-inverse quasi-random point sequence," *Communications of the ACM*, Vol. 7, No. 12, 1964, pp. 701–702. https://doi.org/10.1145/355588.365104.

[43] Garnett, R., *Bayesian Optimization*, Cambridge University Press, 2023.

[44] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, May 7-9, 2015, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun, 2015. URL http://arxiv.org/abs/1412.6980.

[45] Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J., "GPflow: A Gaussian process library using TensorFlow," *Journal of Machine Learning Research*, Vol. 18, No. 40, 2017, pp. 1–6. URL http://jmlr.org/papers/v18/16-537.html.

[46] Liu, K., Li, Y., Hu, X., Lucu, M., and Widanage, W. D., "Gaussian Process Regression With Automatic Relevance Determination Kernel for Calendar Aging Prediction of Lithium-Ion Batteries," *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 6, 2020, pp. 3767–3777. https://doi.org/10.1109/TII.2019.2941747.

[47] Kuleshov, V., Fenner, N., and Ermon, S., "Accurate Uncertainties for Deep Learning Using Calibrated Regression," *CoRR*, Vol. abs/1807.00263, 2018. URL http://arxiv.org/abs/1807.00263.

[48] Scalia, G., Grambow, C. A., Pernici, B., Li, Y.-P., and Green, W. H., "Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction," *Journal of Chemical Information and Modeling*, Vol. 60, No. 6, 2020, pp. 2697–2717. https://doi.org/10.1021/acs.jcim.9b00975.

[49] Masini, L., Timme, S., and Peace, A. J., "Analysis of a civil aircraft wing transonic shock buffet experiment," *Journal of Fluid Mechanics*, Vol. 884, 2020, p. A1. https://doi.org/10.1017/jfm.2019.906.