GPPS | JOURNAL

JOURNAL OF THE GLOBAL POWER AND PROPULSION SOCIETY
journal.gpps.global

Physics guided machine learning modelling of compressor stall flutter

Original article

Article history:
Submission date: 24 December 2023
Acceptance date: 26 April 2024
Publication date: 13 August 2024

This is the updated version of a paper originally presented at the Global Power and Propulsion Technical Conference, GPPS Hong Kong23, October 17–19, 2023.

Check for updates

Marco Rauseo[1,*], Fanzhou Zhao[1], Mehdi Vahdati[1]

[1]*Imperial College London, Exhibition Road, SW7 2AZ London, UK*

*Correspondence:
MR: mr618@ic.ac.uk

Peer review:
Single blind

Keywords:
flutter; compressor; machine learning

Citation:
Rauseo M., Zhao F., and Vahdati M. (2024). Physics guided machine learning modelling of compressor stall flutter. Journal of the Global Power and Propulsion Society. 8: 295–309.
https://doi.org/10.33737/jgpps/187996

Abstract

Modern aircraft engines need to meet ever more stringent requirements that greatly increase the complexity of design, which strives for enhanced performance, reduced operating costs, emissions and noise simultaneously. The drive for performance leads to the development of thin, lightweight, highly loaded fan and compressor blades which are increasingly more prone to incur high, sustained vibratory stresses and aeroelastic problems such as flutter. The current practice employs preliminary design tools for flutter that are often based on empiricism or simplified analytical models, requiring extensive use of computational fluid dynamics to verify aeroelastic stability. As the industry moves to new designs, fast and accurate prediction tools are needed. In this work, data-driven techniques are employed to model the aeroelastic response of compressor blades. Machine learning has been applied to a plethora of engineering problems, with particular success in the field of turbulence modelling. However, conventional, black-box data-driven methods based on simple input parameters require large databases and are unable to generalise. In this work a combination of machine learning techniques and reduced order models is proposed to address both limitations at the same time. Previous knowledge of flutter is introduced in the physics guided framework by formulating relevant, steady state input features, and by injecting results from low-fidelity analytical models. The models are tested on several unseen cascades and it is found that training on even a single geometry yields accurate results. The models developed here allow flutter prediction of fan and compressor flutter stability based on the steady state flow only without a need for any CPU intensive unsteady simulations. Hence, one can predict flutter stability of a given blade for different mechanical properties (mode shape, frequency) at near zero additional cost once the mean flow is known.

Introduction

Stall flutter occurs at part-speed regime, when airfoils operate at an incidence higher than nominal (Dowell, 2015); it is normally associated with high steady loading and low reduced frequency, for blades vibration in the first flap (1F), sometimes referred to as flapwise bending (also simply bending), or torsion (1T) mode (Srinivasan, 1997), in forward travelling assembly modes, with positive low nodal diameters from 1 to 6 (Vahdati et al., 2001). Vahdati et al. (2011) state there are two types of mechanisms that drive stall flutter: flow driven and acoustic driven.

For the flow driven mechanism, the inception of stall itself is not a necessary condition, though the flow separation, and its interaction with the suction side shock wave, is a key component for this type of instability. On the other hand, the acoustic driven mechanism is driven by the interaction of blade vibration with an upstream running wave caused by the blade vibration itself and the reflected wave from the intake. These conclusions are confirmed by a number of previous and subsequent studies.

Isomura and Giles (1998) carried out a numerical study of a transonic fan and concluded that the source of the instability is the shock wave/boundary layer interaction (SBLI) which primes the shock to oscillate, destabilising the blade. Vahdati et al. (2001) also performed a numerical study on a different fan blade, finding that actually the shock wave has a stabilising effect, while the opposite is true of the flow separation. It is therefore clear that these flow features can both exacerbate and eradicate flutter, depending on geometry, flow field, modeshape and frequency. Vahdati et al. (2011) carried out a numerical campaign to investigate the mechanism of wide-chord fan flutter. They found that SBLI only is not sufficient to explain the loss of stability near stall, and that this event is driven by acoustic waves propagating upstream being reflected at the intake highlight. They name this loss of stability "flutter bite". Vahdati and Cumpsty (2015) performed a numerical analysis of a state-of-the-art fan blade, and concluded that flow separation, followed by radial migration of flow along the span towards the casing, decreases aerodynamic damping. They found that aerodynamic damping reduces as the twist component in the first flap mode increases, similar to what Panovsky and Kielb (1999) concluded for turbines. Finally, they state that flutter occurs in the frequency and nodal diameter range where the generated acoustic waves are cut-on upstream and cut-off downstream.

Building on this knowledge, Stapelfeldt and Vahdati (2019) presented two strategies to improve the flutter margin of an unstable fan blade. The first addresses the flow driven mechanism by restaggering sections of the blade, so that the radial pressure distribution is altered and the flow separation, with consequent radial flow migration, is mitigated. The second strategy involves bleeding a small amount of fluid at different chord locations, in order to attenuate the upstream running pressure wave from trailing edge to leading edge, thus addressing the acoustic driven mechanism. Finally, Rendu et al. (2019) presented a radial decomposition method of the blade vibration to identify the flutter source of a fan blade. They show that the fan blade is rendered unstable due to the highly destabilising contribution of the tip section of the blade, where the separation is most severe. By dividing the blade into radial panels and vibrating them individually, they found that the instability though is not caused by the tip itself, but rather is the vibration of lower panels that induces a destabilising unsteady pressure on the tip.

The challenge of predicting compressor flutter lies in the complex interactions between aerodynamic flow, structural mechanics and aeroacoustics which involves a large number of parameters. Analytical and computational methods for the prediction of compressor stall flutter have reached maturity, and development of new efficient techniques seems to have come to a halt. Machine learning offers a large set of tools to explore and model complicated phenomena with many variables involved, and it has proved successful in turbulence modelling, bridging the gap between closure models and high fidelity methods in a number of applications.

Data-driven models for fluid mechanics applications, though, often share a fallacy: they are not reusable, as predictions for the quantities of interest are only possible on training geometry or flow condition. The difficulties with extrapolation from the training space are well known in machine learning and efforts have been made to address these limitations with varying degrees of success. All the paradigms are based on introducing physical or domain knowledge into the process, and a brief summary of such approaches is in order to justify the choices made in this work.

Raissi et al. (2019) introduced the physics informed neural network (PINN) concept and applied it to solve a number of problems, including two dimensional shedding around a cylinder. The PINN framework includes the differential form of the governing equations, and relative boundary conditions, as loss terms inside the cost function so that predictions are consistent with the underlying physical problem, allowing the model to extrapolate to unseen conditions. However, even in their state-of-the-art form, PINNs have only been applied to relatively simple fluid flow problems (Jin et al., 2021; Oldenburg et al., 2022) and, more importantly, are still geometry dependent, i.e. a model trained on one geometry cannot be used on another.

Physical understanding can be incorporated through domain knowledge as well. Manepalli et al. (2019) built a model to predict snow accumulation on mountains and have incorporated certain domain knowledge via additional penalty terms into the cost function, which penalise the model for predicting snow accumulation on water surfaces. Daw et al. (2020) used an LSTM to predict lake water density and, as density can only increase with depth, they constrained their model to only predict monotonic, increasing trends. Frey Marioni et al. (2021) seeked to improve the modelling of the eddy viscosity term in the Boussinesq assumption, using neural networks trained on DNS data, with the ultimate goal of improving flow field predictions in a turbomachinery cooling flow. Rather than directly training on the complicated flow, they approximated it with a serpentine passage and employed hierarchical agglomerative clustering to classify and divide flow regions. They showed improved results by training only on regions without flow separation. Frey Marioni et al. (2022) built on their work to model wakes in an LPT cascade. Once again they showed that dividing the domain into simpler subdomains and employing clustering improves the predictive capabilities of their ML models.

Particularly in turbulence modelling, physical knowledge is incorporated in the formulation of meaningful input features. Ling and Templeton (2015) developed a number of classifiers to identify regions of high uncertainty in RANS computations, and ensured that the models had good generalisation properties by providing several features that are rotationally invariant and based on physical intuition. Their set of features has since been adopted by other researchers (Wang et al., 2017; He et al., 2022).

Finally, Pawar et al. (2021) introduced the concept of physics guided machine learning (PGML) and applied it to the prediction of lift coefficient on 4 and 5 digits NACA airfoils. The PGML framework in this case refers to a fully connected neural network (FCNN) where, at some hidden layer, results from a reduced order model (ROM) of the process are injected, thus augmenting the knowledge of the algorithm and "guiding" it to a physically consistent formulation. The same framework has been applied to modelling of dynamical systems with good degree of success Robinson et al. (2022), and both studies show good generalisation properties. This framework is appealing as it is computationally inexpensive, easily implemented, builds incrementally on previous, physically sound knowledge and acts as a regulariser, avoiding excessive overfitting even in a *small-data* regime.

Considering the discussion above, we seek to build a machine learnt model using the PGML framework with meaningful input and output features, based on physical understanding of stall flutter. Each component of the PGML model developed in this work will now be discussed.

Methodology

Flow solver

In this study, the data necessary to build our PGML model are obtained solely with computational methods. The CFD solver used in this work, LUFT, is developed by Dr. Paul Petrie-Repar from RPMTurbo. The steady-state solver employs a cell-centered finite volume scheme to solve the 3D RANS equations, on a computational grid composed exclusively of hexahedral elements. The fluxes are calculated by means of an upwind AUSM scheme and, at cell interfaces, the flow is reconstructed with a MUSCL interpolation coupled with the van Albada limiter. Finally, a pseudo-time stepping with residual smoothing is applied to march the solution to convergence.

The unsteady flow solver is linearised. The time dependent flow equations are cast in the frequency domain, so that the flow is assumed to be harmonically oscillating about its steady-state

$$U = \bar{U} + \Re\{\tilde{U} e^{j\omega t}\} \tag{1}$$

U is the unsteady flow solution, $\bar{U}$ is the steady-state solution and $\tilde{U}$ is an unknown complex number corresponding to the small amplitude flow perturbation. The linear system of equations arising from linearisation is solved with a preconditioned GMRES scheme. Throughout the study, a linearised Spalart-Allmaras turbulence model without wall functions is used and 2D non-reflecting boundary conditions are applied. Further information and validation against other established codes can be found in (Petrie-Repar et al., 2006; Frey et al., 2019).

Test case

The test case selected for this study is the Standard Configuration 10 (Frannson and Verdon, 1991), a 2D compressor cascade which consists of modified, cambered NACA0006 airfoils. The geometry is fixed, with a chord length, c, of 0.1 meters, a solidity s of 1.0 and a stagger angle ξ of 45°. In this study, the total inlet pressure and temperature are fixed at 101.3 KPa and 300 K respectively and, at design point, the cascade operates with inlet Mach number $M_1 = 0.7$ and inflow angle $\alpha_1 = 55°$. Throughout the rest of the paper, we will refer to incidence of flow onto the blade as $\beta_1 = \alpha_1 - 55°$: positive values indicate incidence higher than nominal, therefore moving the cascade closer to stall.

The grid used for both steady and unsteady computations is quasi-2D, i.e. one cell layer along the span, it has been obtained through a convergence study and it consists of 42, 268 nodes with a $y^+ < 1$ at the blade surface. Total pressure, temperature and flow angle are used as inlet boundary conditions, whereas static pressure is imposed at the domain outlet. The boundaries are located 1.5 chord lengths away from leading and trailing edge. The computational domain is a single passage; periodic and phase lagged boundary conditions are applied for steady and unsteady computations respectively.

Machine learning algorithm

The backbone of the PGML model is a fully connected neural network. The choice of an FCNN is motivated by its layered structure, which allows full control over the location where the low-fidelity results are injected. A schematic representation of the FCNN is given in Figure 1.

A number of hyperparameters need to be specified now to build the model, such as number of hidden layers, number of neurons per hidden layer, regularisation coefficient, bias value, number of iterations, minibatch and step size, activation function and location of low-fidelity results injection. In this case, a full factorial exploration is unfeasible to say the least, therefore the standard practice of random search is employed. A set number of FCNNs will be trained with hyperparameters chosen randomly within pre-established ranges; the loss metrics are evaluated and the best models are then selected for a posteriori assessment.

In this work, we aim for small, simple FCNN architectures, therefore a large number of hyperparameters can be tested in parallel, even on conventional CPUs. The optimisation algorithm is Adam (Kingma and Ba, 2014), and the cost function is the standard mean squared error (MSE). The model is implemented in PYTHON with the TENSORFLOW (Abadi et al., 2015) library.

Reduced order model

The reduced order model we are seeking must:

- be efficient, thus adding minimum computational effort
- take into account some, if not all, geometrical features of the cascade
- model compressibility effects, due to the Mach number range of interest for stall flutter
- be validated

Unlike the machine learning algorithm, there are not many reduced order models for the aeroelastic response of turbomachinery cascade. The most viable option, fitting our requirements, is the semi-analytical model LINSUB, based on a theory by Whitehead (1972) for linearised, subsonic, inviscid, unsteady flow through an infinite cascade of two-dimensional flat plates. The theory ignores flow deflection, thickness and camber effects, thus ignoring steady blade loading. LINSUB calculates a matrix of unsteady load coefficients induced by simple rigid body modes, plunging normal to the chord and pitching about the leading edge.

Output formulation

The same coefficients computed by LINSUB can be calculated using CFD. Two computations are necessary to calculate the unsteady pressures induced by each mode, which are consequently integrated along the blade to find forces and moments. Equation (2) shows the definition of two of the coefficients, for sake of brevity. The superscript refers to the mode causing the unsteady loads, while the subscript refers to the modal force, thus, L for lift when projecting the loads onto the plunge mode and M for moment, which is the modal force for a pure

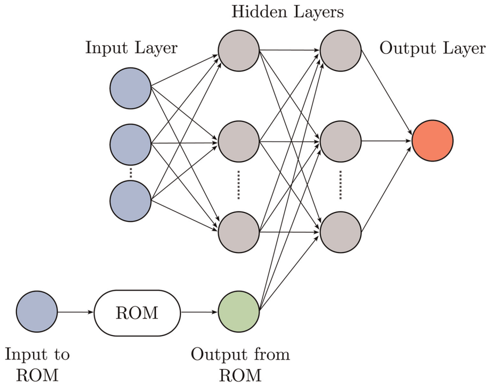

Figure 1. A schematic representation of a PGML network with two hidden layers and one-dimensional output.

pitch.

$$C_L^h = \frac{1}{S(p_0 - p_1)\bar{h}} \int_S \hat{p}^{gth} \boldsymbol{n} \cdot \boldsymbol{\Phi}^h dS; \quad C_M^\alpha = \frac{1}{cS(p_0 - p_1)\bar{\alpha}} \int_S (\boldsymbol{r} \times \hat{p}^\alpha \boldsymbol{n}) \cdot \boldsymbol{\Phi}^\alpha dS; \quad \boldsymbol{C} = \begin{bmatrix} C_L^h & C_L^\alpha \\ C_M^h & C_M^\alpha \end{bmatrix} \tag{2}$$

S is the blade surface area, c its chord, (*p*0–*p*1) is the difference between inlet total and static pressure equals to the dynamic head, $\boldsymbol{\Phi}$ is the modeshape and $\bar{h}$, $\bar{\alpha}$ the modal amplitudes, $\boldsymbol{n}$ is the blade surface unit vector, $\hat{p}$ the complex unsteady pressure and, finally, $\boldsymbol{r}$ is the position vector from the pitching axis to the blade.

The interest of this work lies in stall flutter which, mostly, affects blades vibrating in first flap mode. A flap mode though is a rotation about a generic axis located at x chords from the leading edge, therefore, one can convert the matrix $\boldsymbol{C}$ into the new reference frame centered at x

$$\boldsymbol{H} = 2 \begin{bmatrix} 1 & 0 \\ x & 1 \end{bmatrix} \begin{bmatrix} C_L^h & C_L^\alpha \\ C_M^h & C_M^\alpha \end{bmatrix} \begin{bmatrix} 1 & x \\ 0 & 1 \end{bmatrix} \tag{3}$$

Finally, the obtained unsteady moment coefficient can be translated into an aerodynamic damping coefficient ζ (see Eq. 15 in Frannson and Verdon (1991)), resulting in

$$\zeta = -\Im\{C_M^\alpha\} + (\Im\{C_M^h\} + \Im\{C_L^\alpha\})x - \Im\{C_L^h\}x^2 = \zeta_M^\alpha + \zeta_M^h + \zeta_L^\alpha + \zeta_L^h \tag{4}$$

where, again, the superscript refers to the mode causing the unsteady loads, while the subscript refers to the induced force.

The quantity of interest (QoI) of the PGML model will therefore be the matrix $\boldsymbol{C}$, from which plunge, pitch and any flap mode stability can be evaluated.

Input formulation

The purpose of the proposed model is to predict unsteady force coefficients for different geometries. The most basic avenue to achieve this goal requires an expansion of the training database through the inclusion of several different geometries, so that the surrogate model can then interpolate to produce predictions on cases that fall within the training space. This approach is appealing as, once the surrogate model is trained, it bypasses the CFD solver completely and only boundary conditions and geometry are needed to obtain the set of unsteady forces. Unfortunately, this approach is also unfeasible as the number of training samples needed would be extremely large, moreover, even if one were to simulate a large number of cases, the definition of these geometries would most likely need to be parametrised through an autoencoder, which is a task with pitfalls in itself (Gambitta et al., 2022; Yonekura et al., 2022). Therefore, the approach followed in this work relies on a different strategy.

In order to predict unsteady forces on the blade, the surrogate model needs information about the geometry and flow variables. If the geometry is to be explicitly passed as an input, clearly several cascades need to be used to build a training set. On the other hand, the flow features, and their relationship, form a set of dependent variables which is defined by the boundary conditions and geometry themselves. It is thus argued that, rather than passing the geometry explicitly, the PGML model can build a functional mapping for the QoI from a set of unsteady boundary conditions and steady state features, which already encode the geometry in their mutual dependence. Effectively, the steady CFD solver acts as an encoder to translate the geometry into an input the PGML can work with. This approach is somewhat similar to that of Ling and Templeton (2015). The drawback of this solution is that a run of steady state solver is needed to calculate the input features at prediction time. This is deemed an acceptable price to pay, as the aerodynamic damping for different modeshapes, frequencies and interblade phase angles, can be calculated at no extra cost.

The careful choice of input features is crucial to develop a model that can generalise. The features must not only be relevant for stall flutter, but must also carry information about the domain, without harming the learning process: too many input features will introduce information that might be irrelevant and add dimensions to the data; on the other hand, if the number of inputs is too low, the model will not be able to capture a change in geometry or flow conditions and will thus be unable to generalise.

Steady state features

A number of constraints are established to reduce the space of possible features, which would otherwise grow too large. The features must be continuous and relevant for both subsonic and transonic flow, thus no binary variables, e.g. shock/no shock or pressure jump at shock front. They must also be relevant for both attached and separated flow, which denies the inclusion of quantities such as size of separation. Finally, these features must be either non-dimensional or rescaled by a relevant quantity. The last point will be explained more carefully as we list them.

It is stressed that the absence of quantities relating "directly" to the presence of shocks and separation does not imply that these flow features are unimportant, but simply that their effect can be represented by other flow features. The following features have been chosen based on physical intuition.

Inlet Mach number

The inlet Mach number is one of the most important non-dimensional groups in turbomachinery and it is also an input to LINSUB.

Chordwise center of pressure

The steady pressure distribution on the blade is important to determine stability, thus a parameter to capture loading is needed. Force and moment coefficients give a good indication of loading, but vary widely between geometries and cannot be used for generalisation unless normalised by some reference value which is hard to define. The same applies to pressure ratio, pressure rise and flow coefficients, and similar parameters. Wong (1997) and Kiss (2021) both showed strong correlation between aerodynamic damping and location of chordwise center of pressure x_{cp}. The parameter carries a lot of information regarding the state of the pressure field around the blade and for a typical compressor section, unless the flow is severely stalled or choked, it is bounded between 0 and 1, corresponding to leading and trailing edge respectively.

Blockage near trailing edge

Blockage is the reduction in flow area caused by local velocity deficit which is a result of the displacement thickness associated with boundary layers. It is a crucial variable for compressor design and a great indicator of either increasing incidence onto the blades or flow separation due to a strong shock wave. The blockage definition used in this work is adapted from Khalid et al. (1999) and is briefly discussed here. First, the authors define a main flow direction to select a velocity component, u_m, to use in blockage definition. As we are interested in the blockage due to the boundary layers in the bladed passage, the main flow direction is the one aligned with the stagger angle ξ. Second, they identify the edge of the velocity deficit region by recognising that near solid walls, the gradients are stronger than in the core flow. The edge is defined as the region where the magnitude of the gradient $\vec{\nabla}\rho u_m$, normalised by inlet density, inlet axial velocity and blade chord, is equals to a target value. The target is set to 3 as in the original paper. As our test case is two-dimensional, the expression for blockage area becomes a blockage width δ defined as

$$\delta = \int \left(1 - \frac{\rho u_m}{\rho_e U_e}\right) \cdot dl \tag{5}$$

where dl refers to integration along a line normal to the blade surface, while ρ_e and U_e are the values of density and velocity at the blockage edge.

The blockage can only grow on the blade, therefore it is calculated near the trailing edge where it reaches its maximum, both for suction and pressure side, because the relationship between the two changes depending on the operating conditions. In subsonic flow, the blockage on suction and pressure side are, respectively, directly and inversely proportional to incidence; in transonic flow this is not necessarily the case as, even with low incidence, the shock wave forming on the suction side will grow the blockage while the pressure side is effectively unchanged (assuming the flow is not choked). This feature is bounded by the passage width but, at the same time, the boundary layer growth is also a function of the chord length. Moreover, in the limiting case of null mass flow rate, there should not be any blockage at all. It is found that the best approach to account for these observations is to rescale the blockage size by the solidity s, i.e. chord to pitch ratio.

Wake momentum thickness

A further indicator of the state of boundary layers and losses of the cascade is the wake momentum thickness. This quantity, similar to blockage, is proportional to incidence and inlet Mach number. Moreover, it correlates to the diffusion factor $u_{\max}/u$ (Lieblein and Roudebush, 1956), which gives an indication of fluid velocities on the suction side, but unlike the diffusion factor, the same considerations put forth regarding blockage size can be applied here, therefore this feature can also be rescaled by the solidity. The definition of momentum thickness used here is found in Dixon and Hall (2013)

$$\Theta = \int \frac{u}{U}\left(1 - \frac{u}{U}\right) dy \tag{6}$$

where U and u are velocity outside and inside the wake region respectively, and dy indicates integration in the pitchwise direction.

Stagnation pressure loss coefficient

Losses in cascades are commonly expressed in terms of stagnation loss coefficient.

$$\bar{\omega} = \frac{p_{01} - p_{02}}{p_{01} - p_1} \tag{7}$$

where 1 and 2 indicate inlet and outlet. This quantity is easy to calculate and correlates well with other loss parameters.

Unsteady features

The unsteady features are the interblade phase angle σ and the blade reduced frequency k. Arguably, these quantities could be better represented by their ratio with cut-off frequencies (for an exhaustive explanation of acoustic modes in turbomachinery ducts see Tyler and Sofrin (1962)), but for any given combination of frequency and interblade phase angle, there are multiple cut-off conditions. It is thus unclear which one would be the most appropriate to choose for normalisation and for this reason, the values of σ and k are used directly. The periodicity with interblade phase angle is enforced by simply duplicating training samples on either side of the $\pm 180°$ range.

Finally, one would argue that a feature able to capture the change in passage shape is missing. Although this is true, we can make two considerations. First, the change in passage shape is partially modelled in LINSUB. Second, and more importantly, this effect becomes highly relevant for torsional modes at high interblade phase angle, which is a condition that very rarely exhibits stall flutter.

Training data

The Latin Hypercube sampling method (McKay et al., 1979) is applied to generate five independent databases of input parameters. The databases have size 128, 256, 512, 1024, 2048, and are referred to as db1 to db5. Each input sample corresponds to a set of computations, one steady and two unsteady, of the time-linearised solver on the SC10 geometry. The range for inlet Mach number is chosen to be symmetric about the nominal value of 0.7, to study both subsonic flow, where no shock appears, and transonic flow conditions, where a shock is present on the suction side. M_1, thus, changes between 0.5, anything lower is discarded as excessively low speed, and 0.9, so that the shock stays on the suction side and does not move too far upstream. The incidence of incoming flow onto the blade is varied between 0° and 6°, which is the largest value before inception of stall for all the Mach numbers studied in this work. The interblade phase angle σ ranges from $-180°$ to 180°, while the reduced frequency k ranges from 0.5 to 1.0. This choice of intervals for the steady state variables is based on the literature shown in the previous sections. The unsteady computations are performed with two basic modeshapes: a plunge orthogonal to the chord line and a pitch about the leading edge. The force coefficients are found integrating the dot product of modeshape and unsteady pressures as shown in Equation (2).

Validation data

The validation set employed to evaluate the generalisation capabilities of the PGML model is composed of 25 different cascades, each obtained by varying one parameter of the original Standard Configuration 10 (7 cases with different airfoil thickness, 4 airfoil cambers, 5 stagger angles and 9 solidity values). For each cascade, a total of 35 steady state computations is run (7 inlet flow angles and 5 inlet Mach numbers). For each steady state flow, a total of 444 time-linearised computations is run (2 modeshapes, 6 reduced frequency values, 37 interblade phase angles). Therefore, 388, 500 time-linearised computations are needed to compute the validation set.

Results

Model selection

The process of selecting the best hyperparameter setting is composed of several stages.

First, the goal of this process is to have four PGML networks, each predicting one of the quantities of interest, i.e. the components of the unsteady force matrix $\boldsymbol{C}$. This is to alleviate the burden of each machine learnt model and produce better results. Also, this is justified as mathematically these four coefficient values are decoupled.

Second, the learning dataset is composed only of the five SC10 databases, while the large dataset of geometry variations is held out as a validation set. This means that no knowledge of change in geometry is introduced in the learning process, neither explicitly, i.e. using the data for training, nor implicitly, i.e. by finding the hyperparameters that best fit the validation set. The choice of such an approach is the true test of our hypothesis and of the quality of the input features.

Third, as per standard machine learning practice, a cross validation procedure is needed to find the hyperparameters. In this work, the common K-fold cross validation is used: the learning dataset is randomly partitioned in to K groups called folds, the PGML is trained on $K-1$ folds and the left out fold is used as test data. The number of folds is fixed at $K=5$, and as the gradient descent algorithm can fall into local optima depending on weights initialisation and data partition, the computations are repeated 16 times for each network. The output is averaged to generate a prediction, and the parameter setting yielding the best error metric is used to train the final PGML model, using all of the learning dataset. The error, or rather, the performance metric used here is the coefficient of determination R^2

$$R^2(q,\hat{q}) = 1 - \frac{\sum_{i=0}^{m-1}(q_i - \hat{q}_i)^2}{\sum_{i=0}^{m-1}(q_i - \bar{q})^2} \tag{8}$$

where q, $\bar{q}$, $\hat{q}$ are the QoI, its mean across samples and surrogate model prediction respectively, and m is the number of samples. In this case, “best” means the greatest R^2 value. The maximum attainable value of the coefficient is $R^2 = 1$, which corresponds to a perfect model and null prediction error over all samples. Other error or performance metrics can be utilised with no change to the outcome.

Finally, as the space of possible settings is too large to be explored with a full factorial, the cross validation is run using a sample of 1024 hyperparameter combinations generated at random, by drawing from uniform distributions within the specified ranges. The cross-validation R^2 for the four QoIs are given in Table 1.

Figure 2a shows the R^2 values for the four QoIs. The performance on the training set exhibits negligible variation across the QoIs, while the PGML framework improves the prediction capabilities of the simple FCNN ever so slightly; on the other hand, the PGML performance on the validation set varies noticeably, ranging from a maximum of $R^2 \approx 0.92$ when predicting the plunge lift coefficient, to a minimum of $R^2 \approx 0.75$ associated to the predictions of pitch moment coefficient. Furthermore, PGML performs consistently better than the FCNN, while LINSUB predicts poorly across all samples.

The error distribution of PGML on the validation set can be visualised as a probability density function (PDF). The relative error ϵ is defined as

$$\epsilon(q,\hat{q}) = \frac{|q-\hat{q}|}{\bar{q}} \tag{9}$$

Figure 2b reports the relative error distribution for each quantity of interest. For most of the validation samples, the relative errors are within 7.5%. Comparing across the QoIs, the PGML shows a generally higher accuracy in predicting plunge induced coefficients over their pitch induced counterparts, and also PGML is able to better predict unsteady lift over unsteady moments. This result is reassuring for our purposes because, as

Table 1. Cross validation R^2 values.

	C_L^h	C_M^h	C_L^α	C_M^α
R^2	0.952	0.95	0.945	0.947

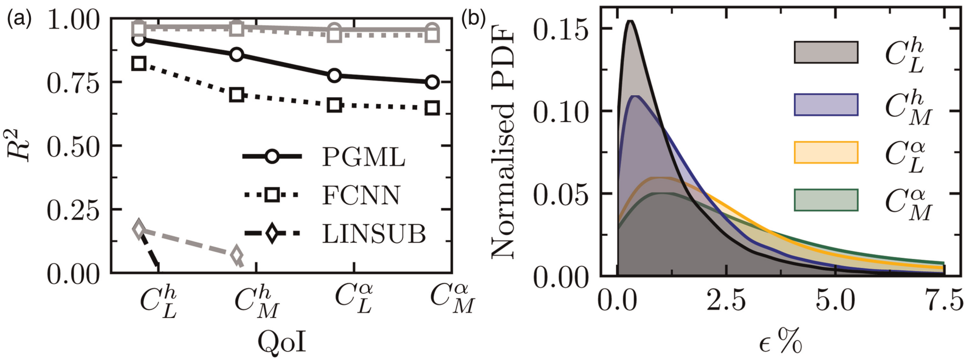

Figure 2. Performance of PGML, FCNN and LINSUB: (a) R^2 values on training (—) and validation (—) sets; (b) normalised distribution of PGML relative error on validation set.

mentioned earlier, Equation (4) shows that for flap modes, i.e. $x \geq 1$, the contribution of plunge induced lift to stability is dominant. These observations are in agreement with the presented R^2 values. The loss in prediction accuracy for pitch induced coefficients was anticipated in the previous section, and it is attributable to the absence of input features pertaining to the change in passage shape. As this effect is particularly relevant in pitch dominated modes, the predictions of the PGML model, although enhanced by LINSUB, result in greater errors, especially at high σ. The interblade phase angle argument is easily confirmed by plotting the QoI values from CFD against the prediction from PGML. Figure 3 illustrates such a plot for for plunge induced lift and pitch induced moment, which are the best and worst predicted QoIs, respectively. The contour is coloured according to the absolute value of interblade phase angle and it clearly shows that, while the model performs really well at low interblade phase angles, as $|\sigma|$ is increased, the agreement between CFD and PGML degrades.

Although imperfect, the presented model fits its intended purpose: stall flutter occurs largely in the regimes where PGML performs well, i.e. flap modes, which are mostly plunge dominated, and at low interblade phase angles.

Prediction on unseen geometries

In this section, the selected machine learnt models will be examined more in depth and compared against the validation set results obtained with CFD. The validation set is composed of all the computations performed on the cascades obtained by varying the SC10 parameters one at a time. It is reiterated that, on the other hand, the training set is constituted of only SC10 computations, therefore all of the following results constitute an extrapolation in terms of geometry.

The plots in Figure 4 show the behaviour of plunge induced lift and pitch induced moment with increasing solidity. The steady state conditions, $M_1 = 0.85$, $\beta_1 = 3°$, are the same for all panels. The interblade phase angle is also kept constant at $\sigma = 20°$. From left to right, the reduced frequency is respectively $k = 0.5$, $k = 0.7$, $k = 1.0$. At low interblade phase angle, as the reduced frequency is increased, the CFD and LINSUB results shift towards more negative coefficient values, corresponding to an increasingly stable configuration. This behaviour is replicated by both PGML and FCNN.

The FCNN produces results that are generally close to CFD and it is able to reproduce the behaviour with increasing solidity at low frequencies, although the agreement seems to degrade both as the frequency is increased

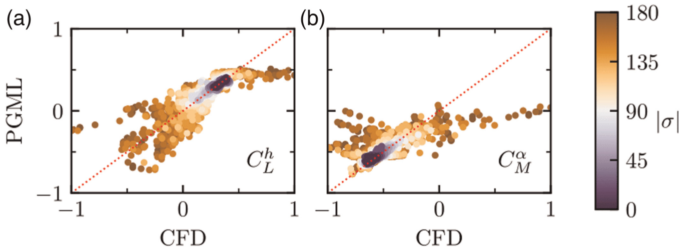

Figure 3. Scatter plot of normalised QoI form validation set. The CFD and PGML predictions are on horizontal and vertical axis respectively. The QoIs are reported on each panel.

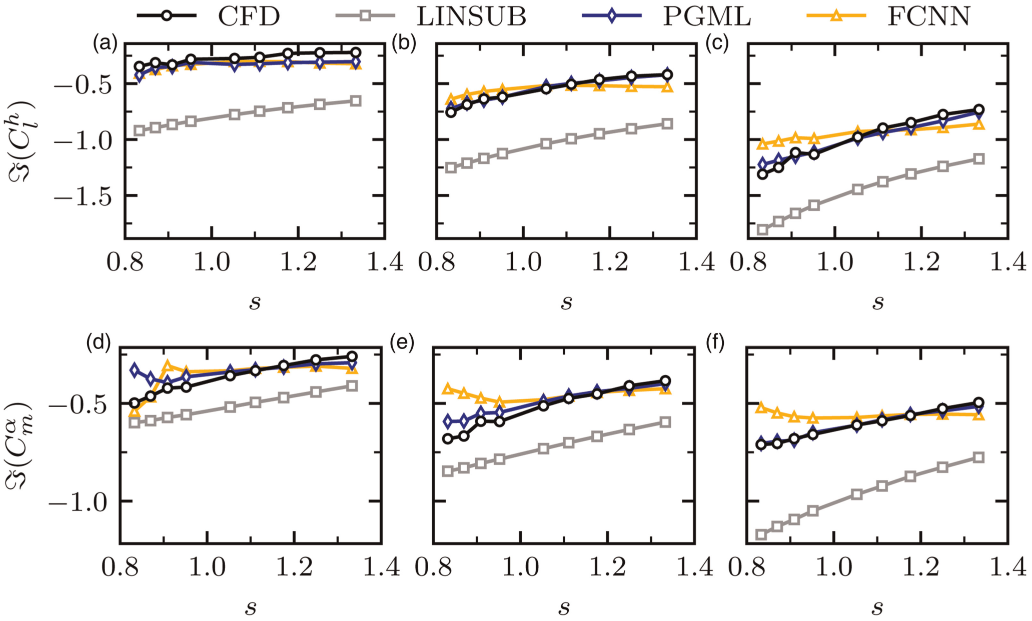

Figure 4. Predictions of plunge lift and pitch moment coefficients from CFD, PGML, FCNN and LINSUB against solidity. The steady state conditions are constant at $M_1 = 0.85$ and $\beta_1 = 3°$. The panels, from left to right, are run with reduced frequency $k = 0.5$, $k = 0.7$, $k = 1.0$, respectively, while the interblade phase angle is constant at $\sigma = 20°$.

and as the cascade operates at a solidity increasingly further from $s = 1.0$. This is a foreseeable consequence of extrapolating further from the training set, i.e. a cascade with solidity $s = 1.0$.

On the other hand, the PGML model follows closely the FCNN at lower frequencies, but its predictions are visibly rectified by LINSUB at higher k, to the point that, in panels **(c)** and **(f)**, the PGML predictions become parallel to those by LINSUB and overlap with the CFD almost perfectly, while the FCNN predicts a nearly constant behaviour. Furthermore, the slope of CFD and LINSUB predictions are nearly identical throughout the frequency range and are only offset by a value.

We now investigate a case with larger interblade phase angle. The plots in Figure 5 show the behaviour of plunge induced lift and pitch induced moment with increasing solidity. The steady state conditions, $M_1 = 0.85$,

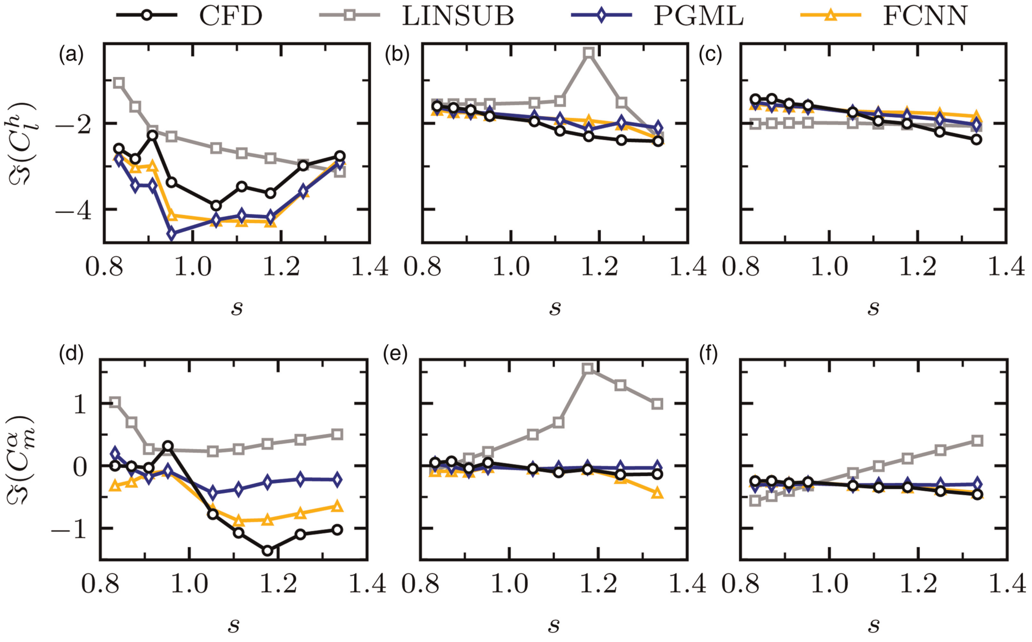

Figure 5. Predictions of plunge lift and pitch moment coefficients from CFD, PGML, FCNN and LINSUB against solidity. The steady state conditions are constant at $M_1 = 0.85$ and $\beta_1 = 3°$. The panels, from left to right, are run with reduced frequency $k = 0.5$, $k = 0.7$, $k = 1.0$, respectively, while the interblade phase angle is constant at $\sigma = 150°$.

$\sigma = 150°$, are the same for all panels. The interblade phase angle is kept constant at $\sigma = 150°$. From left to right, the reduced frequency is respectively $k = 0.5$, $k = 0.7$, $k = 1.0$. Unlike the low interblade phase angle case, the results from CFD and LINSUB are not simply offset by a factor, but behave rather differently with solidity.

With $k = 0.5$ (panels **(a)**, **(d)**), the upstream pressure field changes from cut-on to cut-off at $s \approx 0.95$, causing a spike in aerodynamic forcing, followed by a dip. After reaching a minimum, the trend is inverted and the coefficient value moves closer to zero. This non-monotonic trend is not well predicted by LINSUB, because it does not take into account the inflow angle and flow turning, hence predicting cut-off frequency at a different solidity value. PGML and FCNN approximate the shape of the CFD curve, though with some discrepancy on the predicted value. As the frequency increases, PGML and FCNN predictions improve and nearly overlap with CFD. LINSUB produces results that move closer to CFD, but ultimately predicts a different trend with solidity.

The plots in Figures 6 and 7 show the behaviour of plunge induced lift and pitch induced moment with increasing solidity with $\sigma = 20°$ and $\sigma = 150°$, respectively. Similarly to the case with increasing solidity, the PGML outperforms the naive FCNN and closely follows the results from CFD. Especially for the case of interest of low interblade phase angle, PGML and CFD nearly overlap for all investigated frequencies.

Finally, a comparison of results obtained with a sweep in camber is shown in Figure 8. The steady and unsteady conditions are the same as the ones shown previously. Once again, the PGML is able to correctly predict the QoIs value across a range of conditions. We can appreciate only a small difference between the results from FCNN and PGML, and that is because the latter has learnt that LINSUB does not provide any significant guidance as the camber, and thus the loading, changes.

The input from LINSUB does not help capturing the change in camber, this makes intuitive sense as LINSUB is a flat plate model, though we can see that its effect is constant throughout the range, “pulling” the PGML predictions towards more negative values compared to the FCNN. The following conclusions can be drawn:

- the PGML relies largely on the physics guidance to produce predictions when a change of solidity or stagger angle is taking place, as their effect is modelled in LINSUB; on the other hand, the FCNN has to capture these effects solely through the steady state features provided, hence producing poorer results;
- the extent to which PGML relies on LINSUB depends on the combination of interblade phase angle and reduced frequency;

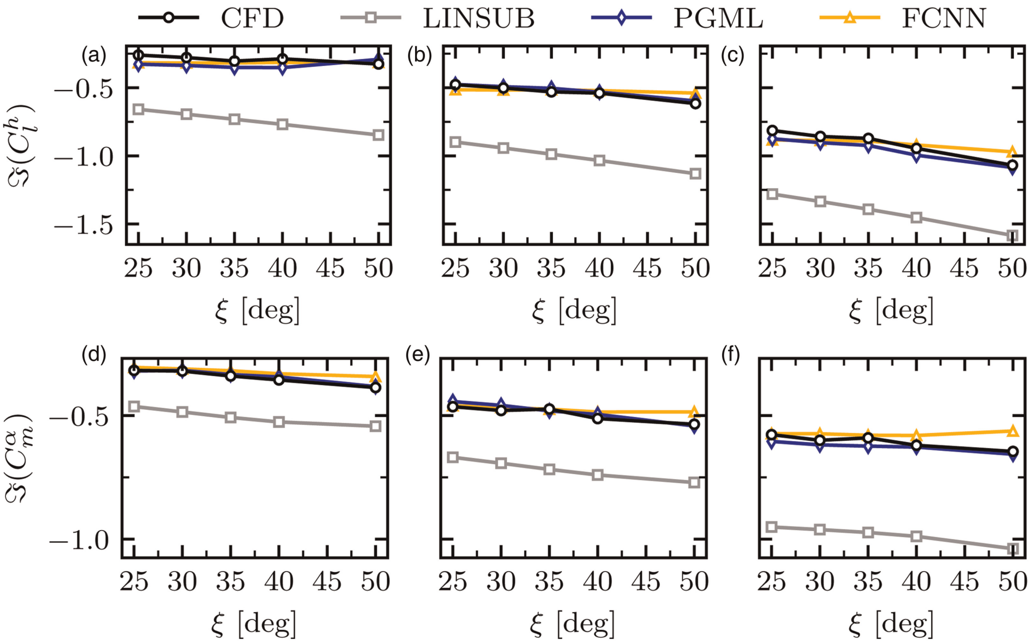

Figure 6. Predictions of plunge lift and pitch moment coefficients from CFD, PGML, FCNN and LINSUB against stagger angle. The steady state conditions are constant at $M_1 = 0.85$ and $\beta_1 = 3°$. The panels, from left to right, are run with reduced frequency $k = 0.5$, $k = 0.7$, $k = 1.0$, respectively, while the interblade phase angle is constant at $\sigma = 20°$.

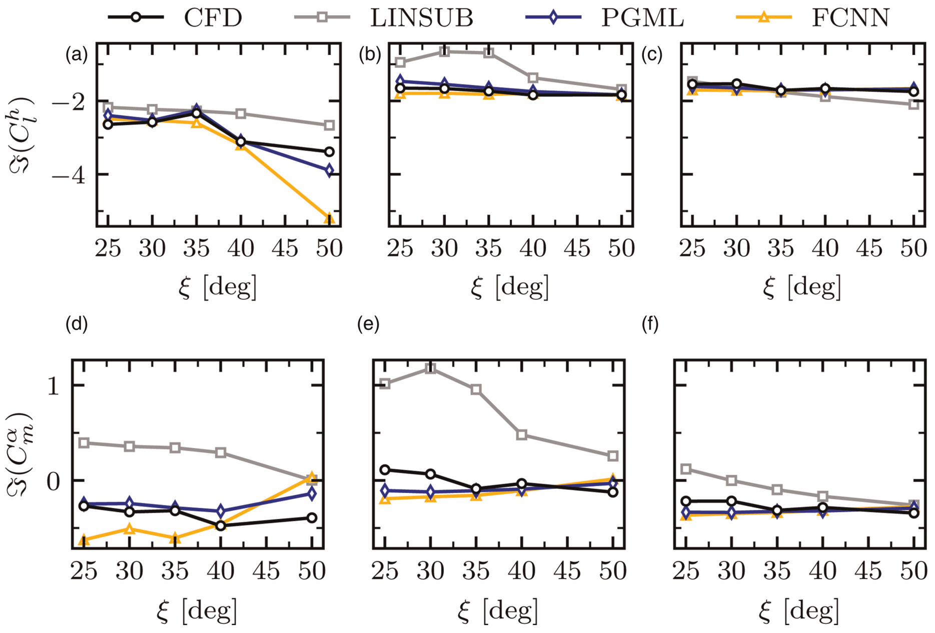

Figure 7. Predictions of plunge lift and pitch moment coefficients from CFD, PGML, FCNN and LINSUB against stagger angle. The steady state conditions are constant at $M_1 = 0.85$ and $\beta_1 = 3°$. The panels, from left to right, are run with reduced frequency $k = 0.5$, $k = 0.7$, $k = 1.0$, respectively, while the interblade phase angle is constant at $\sigma = 150°$.

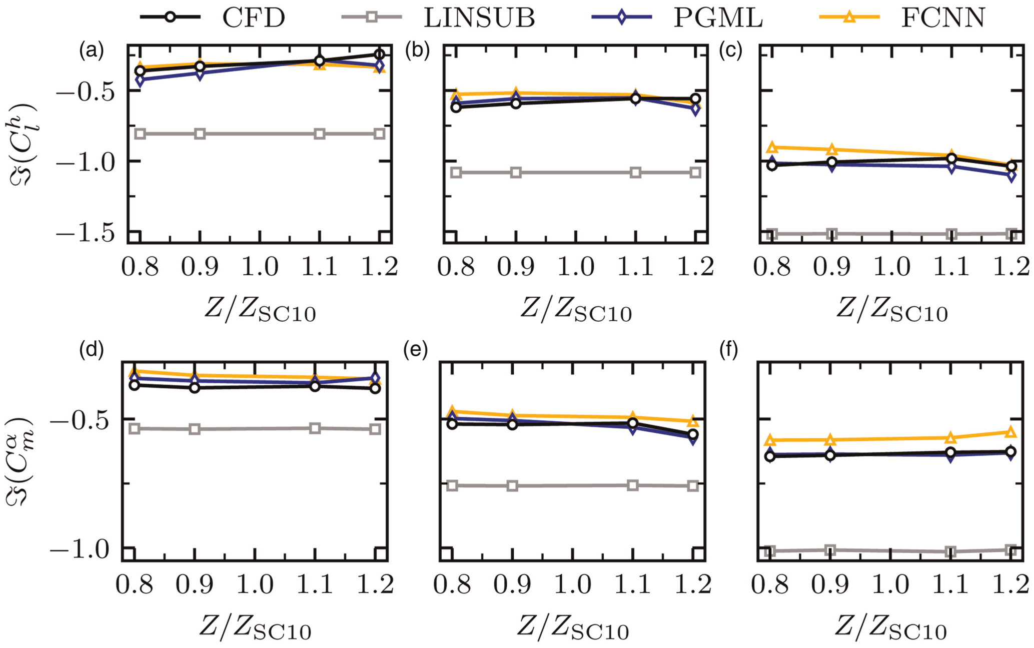

Figure 8. Predictions of plunge lift and pitch moment coefficients from CFD, PGML, FCNN and LINSUB against camber. The steady state conditions are constant at $M_1 = 0.85$ and $\beta_1 = 3°$. The panels, from left to right, are run with reduced frequency $k = 0.5$, $k = 0.7$, $k = 1.0$, respectively, while the interblade phase angle is constant at $\sigma = 20°$.

- the physics input has little contribution when a change in airfoil shape is taking place, as this effect is not modelled in LINSUB; nevertheless, it still provides useful guidance by rectifying the predictions, e.g. in Figure 8f we can see the PGML output being “corrected” towards more negative values.

Conclusions and future work

In this work, a physics guided neural network model has been presented.

The model leverages the PGML framework which is comprised of a fully connected neural network in which predictions of the QoIs from a reduced order model (ROM) are injected at some hidden layer. The ROM employed in this work is LINSUB, a semi-analytical method which models the aeroelastic response of a cascade of unloaded blades, accounting for important parameters such as cascade geometry and flow compressibility. The required data for model training was generated using CFD. The input features to the PGML are steady state quantities that are relevant for stall flutter and carry information regarding the cascade and blade geometry as well as blade loading. The unsteady input features are interblade phase angle and reduced frequency. The QoIs are four unsteady quantities, two lift and two moment coefficients, induced by two basic modeshapes: a plunge orthogonal to the chord line, and a pitch about the leading edge. In this way, the QoIs can be combined to predict aerodynamic damping for any flapwise bending modeshape. The current implementation of the PGML thus requires a steady CFD computation at prediction time to obtain the input features. Such an approach allows us to train the model on a reduced number of training samples as there is no need to pass the geometry directly as an input feature, which would require an unfeasible number of computations to build the training database.

The PGML is first trained with SC10 data only. The hyperpameters are tuned with a random search approach and the best model is selected with a 5-fold cross validation which, again, only employs SC10 data. Consequently, the model is tested on a large database of cascades and it is found that the PGML can predict the QoIs with good degree of accuracy, especially in the low interblade phase angle region, which is of most interest for stall flutter. It is also found that the prediction accuracy for plunge induced coefficients is higher than the pitch induced ones, due to the lack of an input feature pertaining to change in passage shape.

The authors are currently working on testing the model on cascade geometries resembling a tip section of a fan blade.

Appendix - PGML hyperparameters

Table 2. Hyperparameters of PGML networks.

Hyperparameter		C_L^h	C_M^h	C_L^α	C_M^α
Activation	$g(z)$	ReLU	ReLU	Swish	Swish
Batch Size	$\|B\|$	512	256	256	96
Epochs		5,000	4,000	4,000	5,000
Initialisation		He-$\mathcal{N}$	Xavier-$\mathcal{U}$	He-$\mathcal{N}$	Xavier-$\mathcal{U}$
Hidden Layers	L	5	5	6	5
Learning Rate	η	0.003	0.006	0.002	0.004
Neurons per layer	N_L	9	6	8	8
Physics Layer		4	1	5	4
Regularisation	$\log_{10}(\lambda)$.	−4	−4	−5	−5
Scaler		Standard	Standard	Min-Max	Standard

Competing interests

Marco Rauseo declares that he has no conflict of interest. Fanzhou Zhao declares that he has no conflict of interest. Mehdi Vahdati declares that he has no conflict of interest.

References

Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. https://www.tensorflow.org/.

Daw A., Thomas R. Q., Carey C. C., Read J. S., Appling A. P., and Karpatne A. (2020). Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling. In Proceedings of the 2020 SIAM International Conference on Data Mining (SDM), pp. 532–540.

Dixon S. and Hall C. (2013). *Fluid Mechanics and Thermodynamics of Turbomachinery*. Linacre House, Jordan Hill, Oxford: Elsevier Science. https://books.google.co.uk/books?id=wZoTAAAAQBAJ.

Dowell E. H. (2015). *A Modern Course in Aeroelasticity*, 5th edn. Dordrecht: Springer, pp. 428–429.

Fransson T. and Verdon J. (1991). Update Report on Standard Configurations for Unsteady Flow through Vibrating Axial-flow Turbomachine Cascades. Progress report.

Frey C., Ashcroft G., Kersken H.-P., and Schlüß D. (2019). Flutter analysis of a transonic steam turbine blade with frequency and time-domain solvers. *International Journal of Turbomachinery, Propulsion and Power*. 4 (2): 15–26. https://doi.org/10.3390/ijtpp4020015.

Frey Marioni Y., de Toledo Ortiz E. A., Cassinelli A., Montomoli F., Adami P., and Vazquez R. (2021). A machine learning approach to improve turbulence modelling from dns data using neural networks. *International Journal of Turbomachinery, Propulsion and Power*. 6 (2): 17–29. https://doi.org/10.3390/ijtpp6020017.

Frey Marioni Y., Adami P., Vazquez Diaz R., Cassinelli A., Sherwin S., and Montomoli F. (2022). Development of Machine-Learnt Turbulence Closures for Wake Mixing Predictions in Low-Pressure Turbines, Vol. Volume 10C: Turbomachinery — Design Methods and CFD Modeling for Turbomachinery; Ducts, Noise, and Component Interactions of Turbo Expo: Power for Land, Sea, and Air. V10CT32A032. https://doi.org/10.1115/GT2022-82531.

Gambitta M., Beirow B., and Schrape S. (2022). A Digital Twin of Compressor Blisk Manufacturing Geometrical Variability for the Aeroelastic Uncertainty Quantification of the Aerodynamic Damping, Vol. Volume 8A: Structures and Dynamics of *Turbo Expo: Power for Land, Sea, and Air*. V08AT21A021. https://doi.org/10.1115/GT2022-82935.

He X., Tan J., Rigas G., and Vahdati M. (2022). On the explainability of machine-learning-assisted turbulence modeling for transonic flows. *International Journal of Heat and Fluid Flow*. 97: 109038. https://doi.org/10.1016/j.ijheatfluidflow.2022.109038.

Isomura K. and Giles M. B. (1998). A numerical study of flutter in a transonic fan. *Journal of Turbomachinery*. 120 (3): 500–507. https://doi.org/10.1115/1.2841746.

Jin X., Cai S., Li H., and Karniadakis G. (2021). Nsfnets (navier-stokes flow nets): physics-informed neural networks for the incompressible navier-stokes equations. *Journal of Computational Physics*. 426: 109951. https://doi.org/10.1016/j.jcp.2020.109951.

Khalid S. A., Khalsa A. S., Waitz I. A., Tan C. S., Greitzer E. M., et al. (1999). Endwall blockage in axial compressors. *Journal of Turbomachinery*. 121 (3): 499–509. https://doi.org/10.1115/1.2841344.

Kingma D. P. and Ba J. (2014). Adam: A method for stochastic optimization. https://arxiv.org/abs/1412.6980.

Kiss A. L. A. (2021). Forced Response System Identification of Gas Turbine Fan Flutter. PhD thesis, Massachusetts Institute of Technology. http://dspace.mit.edu/handle/1721.1/7582.

Lieblein S. and Roudebush W. H. (1956). Theoretical Loss Relation for Low-Speed 2D Cascade, Technical Report 3662, NACA.

Ling J. and Templeton J. (2015). Evaluation of machine learning algorithms for prediction of regions of high reynolds averaged navier stokes uncertainty. *Physics of Fluids*. 27 (8): 085103. https://doi.org/10.1063/1.4927765.

Manepalli A., Albert A., Rhoades A., and Feldman D. (2019). Emulating numeric hydroclimate models with physics-informed cgans. In NeurIPS 2019 Workshop on Tackling Climate Change with Machine Learning. https://www.climatechange.ai/papers/neurips2019/39.

McKay M. D., Beckman R. J., and Conover W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*. 21 (2): 239–245. http://www.jstor.org/stable/1268522.

Oldenburg J., Borowski F., Öner A., Schmitz K., and Stiehm M. (2022). Geometry aware physics informed neural network surrogate for solving navier–stokes equation (gapinn). *Advanced Modeling and Simulation in Engineering Sciences*. 9. https://doi.org/10.1186/s40323-022-00221-z.

Panovsky J. and Kielb R. E. (1999). A design method to prevent low pressure turbine blade flutter. *Journal of Engineering for Gas Turbines and Power*. 122 (1): 89–98. https://doi.org/10.1115/1.483180.

Pawar S., San O., Aksoylu B., Rasheed A., and Kvamsdal T. (2021). Physics guided machine learning using simplified theories. *Physics of Fluids*. 33 (1): 011701. https://doi.org/10.1063/5.0038929.

Petrie-Repar P. J., McGhee A., Jacobs P. A., and Gollan R. (2006). Analytical maps of aerodynamic damping as a function of operating condition for a compressor profile. *Turbo Expo: Power for Land, Sea, and Air*. 5: 1133–1144.

Raissi M., Perdikaris P., and Karniadakis G. (2019). Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*. 378: 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.

Rendu Q., Vahdati M., and Salles L. (2019). Radial decomposition of blade vibration to identify a stall flutter source in a transonic fan. *Journal of Turbomachinery*. 141 (10): 10–17. https://doi.org/10.1115/1.4044484.

Robinson H., Pawar S., Rasheed A., and San O. (2022). Physics guided neural networks for modelling of non-linear dynamics. *Neural Networks*. 154: 333–345. https://doi.org/10.1016/j.neunet.2022.07.023.

Srinivasan A. V. (1997). Flutter and resonant vibration characteristics of engine blades. *Journal of Engineering for Gas Turbines and Power*. 119 (4): 742–775. https://doi.org/10.1115/1.2817053.

Stapelfeldt S. and Vahdati M. (2019). Improving the flutter margin of an unstable fan blade. *Journal of Turbomachinery*. 141 (7): 071006. https://doi.org/10.1115/1.4042645.

Tyler J. M. and Sofrin T. G. (1962). Axial flow compressor noise studies. In SAE Technical Paper, SAE International.

Vahdati M. and Cumpsty N. (2015). Aeroelastic instability in transonic fans. *Journal of Engineering for Gas Turbines and Power*. 138 (2): 022604. https://doi.org/10.1115/1.4031225.

Vahdati M., Sayma A. I., Marshall J. G., and Imregun M. (2001). Mechanisms and prediction methods for fan blade stall flutter. *Journal of Propulsion and Power*. 17 (5): 1100–1108. https://doi.org/10.2514/2.5850.

Vahdati M., Simpson G., and Imregun M. (2011). Mechanisms for wide-chord fan blade flutter. *Journal of Turbomachinery*. 133 (4): 041029. https://doi.org/10.1115/1.4001233.

Wang J.-X., Wu J.-L., and Xiao H. (2017). Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data. *Physical Review Fluids*. 2: 034603. https://doi.org/10.1103/PhysRevFluids.2.034603.

Whitehead D. S. (1972). Vibration and Sound Generation in a Cascade of Flat Plates in Subsonic Flow, *British A.R.C.* R. & M. No. 3685.

Wong M. T. M. (1997). System Modeling and Control Studies of Flutter in Turbomachinery. Masters thesis, Massachusetts Institute of Technology.

Yonekura K., Wada K., and Suzuki K. (2022). Generating various airfoils with required lift coefficients by combining naca and joukowski airfoils using conditional variational autoencoders. *Engineering Applications of Artificial Intelligence*. 108: 104560. https://doi.org/10.1016/j.engappai.2021.104560.