Aerospace Science and Technology 155 (2024) 109689

Contents lists available at ScienceDirect

Aerospace Science and Technology

journal homepage: www.elsevier.com/locate/aescte

Point-enhanced convolutional neural network: A novel deep learning method for transonic wall-bounded flows

Fernando Tejero *, Sanjeeth Sureshbabu, Luca Boscagli, David MacManus

Centre for Propulsion Engineering, School of Aerospace, Transport and Manufacturing, Cranfield University, Bedfordshire, MK43 0AL, United Kingdom

ARTICLE INFO

Communicated by Mehdi Ghoreyshi

Keywords:
Deep learning
CNN
PointNet
PCNN
Flow prediction

ABSTRACT

Low order models can be used to accelerate engineering design processes. Ideally, these surrogates should meet the conflicting requirements of large design space coverage, high accuracy and fast evaluation. Within the context of aerospace applications at transonic conditions, this can be challenging due to the associated non-linearity of the flow regime. Different methods have been investigated in the past to predict the flow-field around shapes such as airfoils or cylinders. However, they usually have reduced spatial resolution, limiting the prediction capabilities within the boundary layer which is of interest for transonic wall-bounded flows. This work proposes a novel Point-Enhanced Convolutional Neural Network (PCNN) method that combines the advantages of the well-established PointNet and convolutional neural network approaches. The PCNN model has relatively low memory requirements in the training process, preserves the spatial correlation in the domain and has the same resolution as a traditional computational method. The architecture is used for the flow-field prediction of civil aero-engine nacelles in which it is demonstrated that the flow features of peak isentropic Mach number (M_{is}), pre-shock isentropic Mach number and shock location (X/L_{nac}) are captured within $\Delta M_{is} = 0.02$, $\Delta M_{is} = 0.04$, $\Delta X/L_{nac} = 0.007$, respectively. The PCNN model successfully predicts the integral parameters of the boundary layer, in which the incompressible displacement thickness, momentum thickness and shape factor are typically within 5% of the CFD. Overall, the PCNN method is demonstrated for transonic wall-bounded flows for a range of flow physics that include shock waves and shock-induced separation.

1. Introduction

Within an industrial design environment, the ability to rapidly iterate through engineering designs is required to ensure innovation and competitiveness [1]. For this reason, low order methods are typically used in the preliminary design phase [2]. In this respect, they should fulfil the conflicting requirements of quick evaluations, high accuracy, and sufficient coverage of the design space. To meet these requirements emerging approaches that include multi-fidelity methods [3], dimensionality reduction techniques [4] or advanced surrogate modelling [5] are being considered. Over the last years, data-driven methods have been developed for aerodynamic design [6]. They have been successfully demonstrated for a wide range of applications such us geometric design space definition, prediction of regression or classification aerodynamic metrics, flow-field modelling and surrogate-based optimization [7]. For example, Li et al. [8] used a deep convolutional generative adversarial network to produce realistic airfoil shapes by learning the underlying features from an existing airfoil database. The model was able to detect abnormal geometries with expected poor aerodynamic performance without CFD evaluations, and filtered them from a CFD-based optimisation process. For a transonic case with a freestream Mach number of 0.75, this resulted in a reduction in computational cost by a factor of two. Many researchers have used a range of deep learning techniques to build surrogate models and predict regression or classification metrics [7]. They have been successfully applied to aerodynamic applications at transonic operating conditions, which establishes confidence as a feasible method for building preliminary design capabilities.

A range of approaches has been proposed for flow-field prediction [9–12]. In this respect, two different modelling techniques have been considered. The first method focuses on the flow characteristics along the surface of interest. This is referred in this paper as on-surface flow-field prediction. The second metamodelling method is based on the full domain, and is referred to as off-surface flow-field prediction in this work. Within this context, different algorithms have been proposed for the on-surface approach. Franz et al. [9] developed an isomap manifold learning method to predict steady transonic flows along the surfaces

* Corresponding author.
E-mail address: f.tejero@cranfield.ac.uk (F. Tejero).

https://doi.org/10.1016/j.ast.2024.109689
Received 20 April 2024; Received in revised form 17 October 2024; Accepted 21 October 2024
Available online 23 October 2024

Nomenclature

Roman symbols

f	Non-dimensional factor
h	Boundary layer shape factor
L	Length
M	Mach Number
P	Pressure
P	Velocity
r	Radius

Greek symbols

β	Boat-tail angle
ϵ	Vorticity magnitude
ω	Vorticity
σ	root mean square error

Superscripts and subscripts

hi	Highlight
i	Incompressible
if	Initial Forebody
is	Isentropic
max	Maximum radius
nac	Nacelle
te	Trailing Edge

Acronyms

ANN	Artificial Neural Network
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
DSE	Design Space Exploration
MFCR	Mass Flow Capture Ratio
MSE	Mean Square Error
PCNN	Point-Enhanced Convolutional Neural Network
RANS	Reynolds Averaged Navier Stokes
RSM	Response Surface Model

of 2D airfoils and 3D wings. For this study, the geometries were fixed and the flight Mach number and angle of attack were the independent variables. The work considered transonic conditions with a maximum freestream Mach number of 0.82. Relative to a conventional proper orthogonal decomposition (POD) approach, it was found that this novel architecture was more accurate in the prediction of the shock location and pre-shock Mach number. Typically, the isomap method reduced the error by a factor of around 3 relative to POD. Immordino et al. [10] proposed a deep learning framework for the prediction of pressure and skin friction distribution along wings and airframes. The method was tested for the Benchmark Super Critical Wing (BSCW), ONERA M6 wing and the wing-body configurations of the NASA Common Research Model (CRM). Independent low order models were generated for each test-case in which the input parameters were the aerodynamic variables of freestream Mach number (M) and angle of attack (AoA). For this study, M and AoA varied within 0.70 - 0.84 and 0° - 5°, respectively. Relative to numerical simulations, it was demonstrated that a fully connected neural network captures the transonic non-linear characteristics. To further improve the prediction of surface pressure distributions in complex cases, Hines and Bekemeyer [11] developed a novel graph neural network approach. This was also used on the NASA CRM model within the range of flight Mach number of 0.5 and 0.9. The predicted flow-field was used to calculate the lift (C_L) and pitching moment (C_{My}) coefficients. The mean absolute error in the validation snapshots was 0.0046 and 0.0058 for C_L and C_{My}, respectively

Several studies have proposed deep learning methods for off-surface flow-field prediction in airfoil applications at subsonic conditions or moderate transonic regimes. Sekar et al. [12] combined fully connected and convolutional neural networks (CNN) for subsonic cases at Reynolds numbers below 2500. The CNN was used for the geometry parameterization and the fully connected network for the flow estimation. Overall, the prediction error was approximately 3% in the pressure and velocity fields. Duru et al. [13] developed a deep learning methodology based on convolutional neural networks for the flow-field prediction of airfoils at constant freestream Mach number of M = 0.7. The model used the airfoil shape with a distance map as the shape descriptor and predicted the pressure and Mach number distributions around the aerodynamic shapes. Overall, it was demonstrated that the CNN predicts the main flow features within the investigated design space. It was highlighted that the CNN approach has limitations in terms of the predictions near the airfoil wall due to the inherent resolution capability of the convolutional neural network. Wu et al. [14] presented a data augmented generative adversarial network (GAN) for flow prediction with sparse data. The architecture used as inputs the airfoil shape and the freestream Mach number within the range of M = 0.73 - 0.76. The approach predicted the shock location within $\Delta x/c \approx 0.01$. Previous studies for off-surface flow-field prediction are mainly based on models that require as input a Cartesian grid to represent the aerodynamic shape. This results in a notable loss of accuracy near the wall regions, and is a clear drawback of these methods for aerodynamic applications where it is important to quantify the boundary layer characteristics. To improve the predictive accuracy near the aerodynamic surface of interest, Kashefi et al. [15] proposed a point-cloud deep learning method which uses the grid vertices, required in a traditional CFD domain, as a point-cloud input in a PointNet architecture [16]. This approach represents the geometry with the same resolution as a CFD-based simulation and avoids the need to interpolate the data onto a Cartesian grid. The method was successfully demonstrated for incompressible laminar flow across a wide range of aerodynamic shapes, where the root mean square error was typically within 0.1% for the velocity and pressure fields. However, comparisons of the boundary layer profiles were not provided. It is important to note that this study was based on a mesh resolution with only 1k mesh cells. For transonic applications, the cell count is expected to be several orders of magnitude larger, making the PointNet unfeasible due to the associated computational overhead required to train the model. Previous studies using deep learning methods highlighted the suitability for flow-field prediction in a variety of aerodynamic design problems. However, for off-surface predictions it was either not suitable for wall-bounded flows due to the methods used (e.g. CNN) or the mesh resolution was not representative for aircraft applications at transonic conditions (e.g. PointNet). Although the focus of this work is on the broader development of a new deep learning approach that meets both requirements, the method is demonstrated in a transonic aero-engine nacelle configuration.

Future civil aero-engines are anticipated to operate with higher bypass ratios than current in-service configurations [17]. The objective is to lower the specific thrust, enhance the overall propulsive efficiency, and thereby, decrease the engine-specific fuel consumption [18]. These novel aero-engines might encompass larger fan diameters, which presents numerous design challenges. If traditional design rules are used and the housing components are only scaled to accommodate the new fan size, the nacelle drag, overall weight, and the aerodynamic coupling with the airframe will increase. As a result, the next generation of civil turbofan engines is expected to use compact nacelles to realize the advantages offered by the new engine cycles [19,20]. Within the context of low order models for flow-field prediction in aero-engine nacelle applications, there is limited information in the open literature [21–23]. Wang et al. [21] developed a framework for inverse design. The method

was based on an improved generative adversarial network (GAN) coupled with an artificial neural network (ANN) for the prediction of the pressure distribution along 3D non-axisymmetric nacelles. The approach was used to redesign a transonic nacelle at a flight Mach number of M = 0.8, with a drag of reduction of 13.8% compared with a baseline configuration. Similarly, Tejero et al. [22] studied two different meta-modelling approaches for on-surface prediction of compact nacelles. The methods were based on artificial neural networks and a combination of singular value decomposition and ANN. For mid-cruise conditions at M = 0.85, where the various flow features include shock-wave boundary layer interaction or shock induced separation, both methods had an overall uncertainty in the isentropic Mach number prediction of about 0.02. The ANN method had better accuracy to predict pre-shock Mach numbers and shock-wave locations. The only work in the literature for off-surface flow-field prediction in aero-engine nacelles was carried out by Sureshbabu et al. [23]. The study considered a compact civil aero-engine nacelle at mid-cruise conditions (M = 0.85) and the low order model was based on a convolutional neural network. To improve the predictive accuracy around the shock regions, the method was coupled with a Sobel filter for edge detection. Relative to a baseline CNN built with guidelines from previous work in transonic applications, the mean square error for Mach number prediction was reduced by 75%. However, the near-wall resolution of the method was low due to the inherent limitations of the CNN approach.

Deep learning is a suitable method to accelerate the aerodynamic optimisation of complex configurations at early stages of the design process. In this respect, accurate low order models for off-surface flow-field prediction may change the current design practices. For example, they may be an enabling technology for the prediction of real-time flow solutions [24] that can be used within a multidisciplinary process [25]. For transonic wall-bounded flows, it is required to accurately predict the boundary layer. Current methods based only on convolutional neural networks are limited to a low resolution near the walls of interest. Although the PointNet approach preserves the spatial resolution of the training data, it is computationally unfeasible for domains with large number of cells which are expected for transonic applications. The novelty of this paper is in the development of a deep learning capability for flow-field prediction that combines the PointNet and CNN methods. Whilst it is acknowledged that similar architectures have been used in the past for object classification [26], segmentation [27] or detection [28], their application in flow-field prediction has never been demonstrated. By integrating the strengths of PointNet and CNNs, it is ensured high resolution near the wall, improved spatial correlation across the domain, and reduced computational memory requirements during training. The contribution to knowledge is in the quantification of the predictive accuracy of the proposed method with a particular emphasis on the near wall boundary layer flows. Although this method is showed for aero-engine nacelle applications, the findings have broader implications for the aerospace industry, where fast and accurate low-order models are essential for preliminary design studies.

2. Methodology

2.1. Problem description

The data used in this study to build models for flow-field prediction was generated with the process developed by Tejero et al. [29] for 2D axisymmetric aero-engine nacelles. It uses a parametric geometry definition based on the intuitive Class Shape Transformation (iCST) method to define the nacelle shape [30]. Overall, 7 geometric intuitive parameters are used: $L_{nac}, r_{hi}, r_{te}, r_{max}, f_{max}, r_{if}$ and β_{nac} (Fig. 1). The computational grids are based on a well-established multiblock structured strategy [29] with an overall cell count of approximately 30,000 cells. The CFD method solves the viscous and compressible steady Favre-averaged Navier-Stokes equations [31] with the $k-\omega$ Shear-Stress Transport (SST) turbulence model, and a double-precision, implicit, and

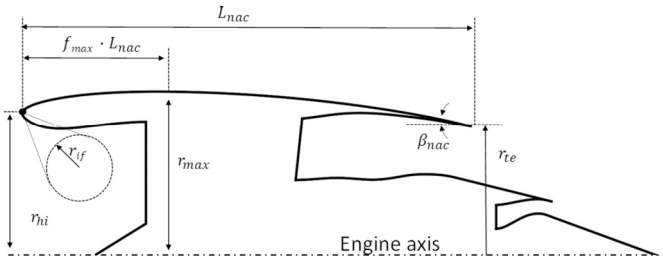


Fig. 1. Geometry definition of the aero-engine nacelle.

Table 1
Bounds of the nacelle design variables.

Variable	Lower bound	Upper bound
r_{max}/r_{hi}	1.1	1.3
f_{max}	0.3	0.5
$f_i f = r_{if} \frac{f_{max} L_{nac}}{(r_{max}-r_{hi})^2}$	0.6	1.5
$beta_{nac}$	9.0	15.0

density-based approach. The first cell height is adjusted for a y^+ of approximately 50. A Green-Gauss node based scheme with a second-order upwind spatial discretisation is used. The thermal conductivity is computed with the kinetic theory and the dynamic viscosity is calculated from Sutherland's law. A pressure-farfield boundary condition models the freestream, where the Mach number and static pressure and temperature are imposed. All the housing components of the aero-engine, i.e. nacelle, intake, spinner, bypass duct and the core duct surfaces, are modelled as no-slip adiabatic walls. The fan-face is defined with a pressure-outlet boundary condition in which the massflow is imposed according to the required massflow flow capture ratio (MFCR). The flow conditions at the bypass and core duct inlets are represented as pressure-inlets using prescribed total pressure and temperature values. This computational approach has been extensively used in previous nacelle design studies, where the CFD validation was reported [32].

As part of this work, the flow-field around the aero-engine nacelles are evaluated with CFD or deep-learning methods (Fig. 2). As discussed below, the low order models are based on convolutional neural networks (CNN), multilayer perceptrons (MLP) or a combination of both (Section 2.3).

2.2. Data preparation

A wide range of different aero-engine nacelle shapes can be generated by changing the geometric intuitive parameters (Fig. 1). This work was based on a compact nacelle with a fixed $L_{nac}/r_{hi} = 3.1$ and r_{te}/r_{hi} 0.91, and the other 4 intuitive parameters were varied, i.e. r_{max}, f_{max}, r_{if} and β_{nac}. The bounds of these variables are based on previously published work [32,34] (Table 1). Across this design space, a wide range of different transonic aerodynamics are encountered, which include attached flows with single and double shock-topologies and shock-induced separation. Therefore, it is a challenging problem to test deep-learning architectures for flow-field prediction.

The generated CFD database is pre-processed to train the investigated deep-learning models (Section 2.3). The inputs are the spatial coordinates of the domain and the output is the prediction of a flow variable, e.g. pressure or Mach number. To improve the performance and training stability of the deep learning models [29], the data was normalised between 0 and 1 (Eq (1)).

$$v_{norm} = \frac{(v - v_{min})}{(v_{max} - v_{min})} \tag{1}$$

where v refers to either the spatial, i.e. x-coordinate and r-coordinate or flow variables, e.g. axial velocity, radial velocity or pressure.

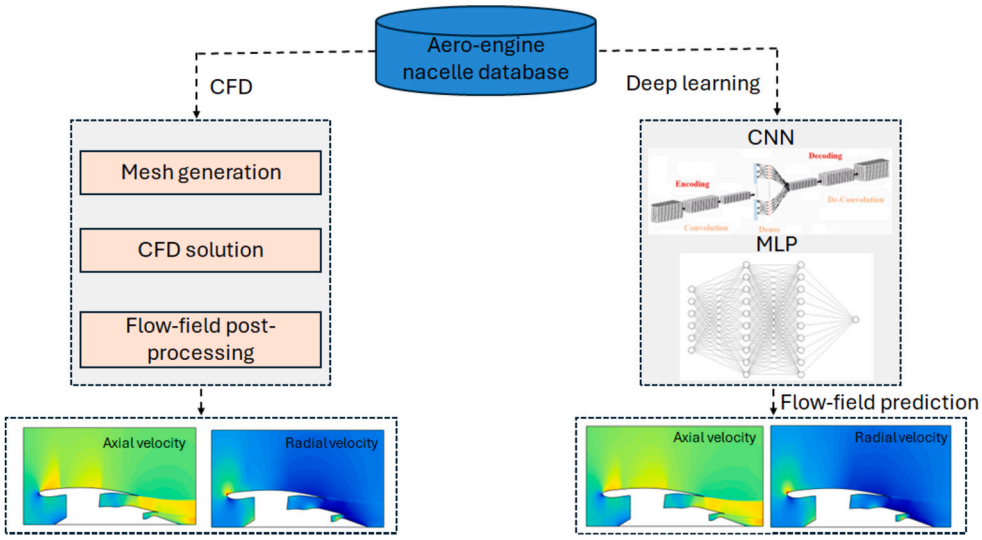


Fig. 2. Workflow to obtain the flow-field around an aero-engine nacelle using CFD or deep-learning methods. Based on [33].

2.3. Deep-learning methods

Over the last years, flow-field prediction methods based on CNNs have been demonstrated. They have been tested for a range of application such as airfoils [13] or aero-engine nacelles [23]. One limitation of these methods is the reduced spatial resolution that is inherent to the CNN, and therefore, the boundary layer cannot be typically predicted. The PointNet method solves this issue by using as input a point cloud with the same resolution as the CFD database [15]. PointNet architectures were introduced by Qui et al. [16] as an enabling method to handle point cloud data for object classification or part segmentation. Subsequently, Kashefi et al. [15] used this method for flow-field prediction for a wide range of shapes, e.g. circles or squares, in laminar and incompressible flow regimes. The PointNet was based on a series of multilayer perceptrons (MLPs) followed by a max-pooling layer to produce an array of global features. Features from an intermediate MLP were combined with the global features of the max-pooling layer through a new series of MLPs to produce the final PointNet output. This provides the objective function of interest for each input of the point cloud, which represents the cells of a traditional CFD computational domain. However, this approach is computationally expensive because the max-pooling operation requires the computation of a $N \times N$ tensor, where N is the number of cells, and each grid point is an array of d where d represents the dimensionality of the problem. Within Kashefi's work [15], the grid consisted of $N = 1024$ cells which was feasible to train. However, complex engineering flows typically need meshes with a significantly greater number of cells.

The proposed novel deep learning method, named as Point-Enhanced Convolutional Neural Network (PCNN), is an adaptation of the PointNet architecture. Both PCNN and PointNet use point clouds as input, and therefore have the same spatial resolution as the CFD training data. PointNet uses a max-pooling layer in a $N \times N$ tensor. This is computationally expensive as the size of the point cloud increases. For this work with a modest cell count of approximately 30,000 the memory requirements were prohibitive, and it was not feasible to train the model. To enable better scalability, the deep learning PCNN method is proposed, in which the max-pooling is performed in a $m \times N$ tensor, where $m << N$. Whilst this approach reduces the memory requirements in the training process, and therefore the PCNN scales better with the number of cells, a drawback is that permutation invariance is lost [15]. As the PCNN method does not preserve global features, Convolutional Neural Networks (CNN) are used to retain spatial correlations in the domain. Therefore, the Point-Enhanced Convolutional Neural Network (PCNN) approach is a combination of existing methodologies to bring together the advantages of each one. PointNet and CNNs are combined to retain the same resolution as the underlying data (PointNet feature) whilst maintaining spatial correlation and reduced computational requirements for training the model (CNN feature). A detailed description of the Point-Enhanced Convolutional Neural Network (PCNN) model is described below, in which the blocking structure is also provided (Fig. 4) to showcase a working approach that can be used for transonic aerodynamic applications. However, it is important to note that other datasets may require changes in the hyperparameters to maximise the predictive accuracy [35]. The PCNN is composed by two main sections. These are a shared MLP and a CNN, referred to in Fig. 4 as Section-I and Section-II, respectively. Each block with a multilayer perceptron contains input, hidden and output layers (Fig. 3a). The input layer can be defined as $x \in \mathbb{R}^N$, where N is the dimensionality of the input features, e.g. number of points of the computational mesh for the first MLP input. For the hidden and output layers, the linear response (v_k) of a k-th neuron can be expressed as in Eq. (2):

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \tag{2}$$

where x_j is the input signal from the j-th neuron of the previous layer, w_{kj} is the weighting coefficient that connect the k-j neurons and b_k is the bias factor.

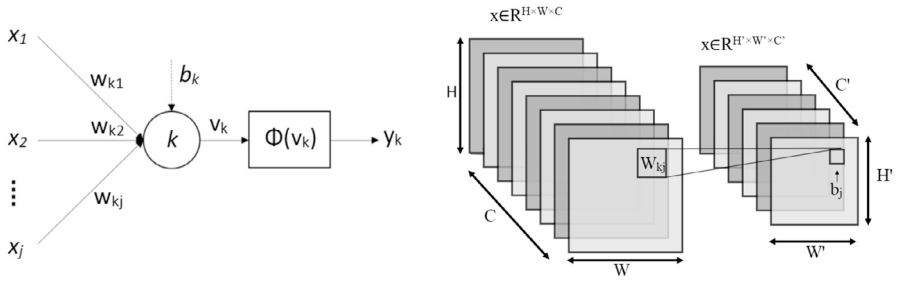
The linear response of a neuron (v_k) is then passed to an active function (ϕ) to produce the output of the neuron (y_k). As part of this work, two different ones are considered: Rectified Linear Unit (ReLU) due to its simplicity and effectiveness for non-linear data [36] (Eq. (3)), and the sigmoid activation function (Eq. (4))

$$\phi(v_k) = \max(0, v_k) \tag{3}$$

$$\phi(v_k) = \frac{1}{1 + e^{-v_k}} \tag{4}$$

After the series of MLPs, the PCNN method follows with a convolutional neural network. The CNN input is a multi-dimensional array, i.e. tensor, that can be defined as $x \in \mathbb{R}^{H \times W \times C}$, where H is the height, W is the width and C is the depth. The information is passed through convolution layers by applying a set of filters, i.e. kernels, to identify and generate feature maps (Fig. 3b). For example, the output of the k-th layer (c_k) can be expressed as in Eq. (5):

$$c_k = \sum_{j=1}^{m} W_{kj} * x_j + b_k \tag{5}$$



(a) MLP (b) CNN

Fig. 3. Sketches of the methods used in the Point-Enhanced Convolutional Neural Network.

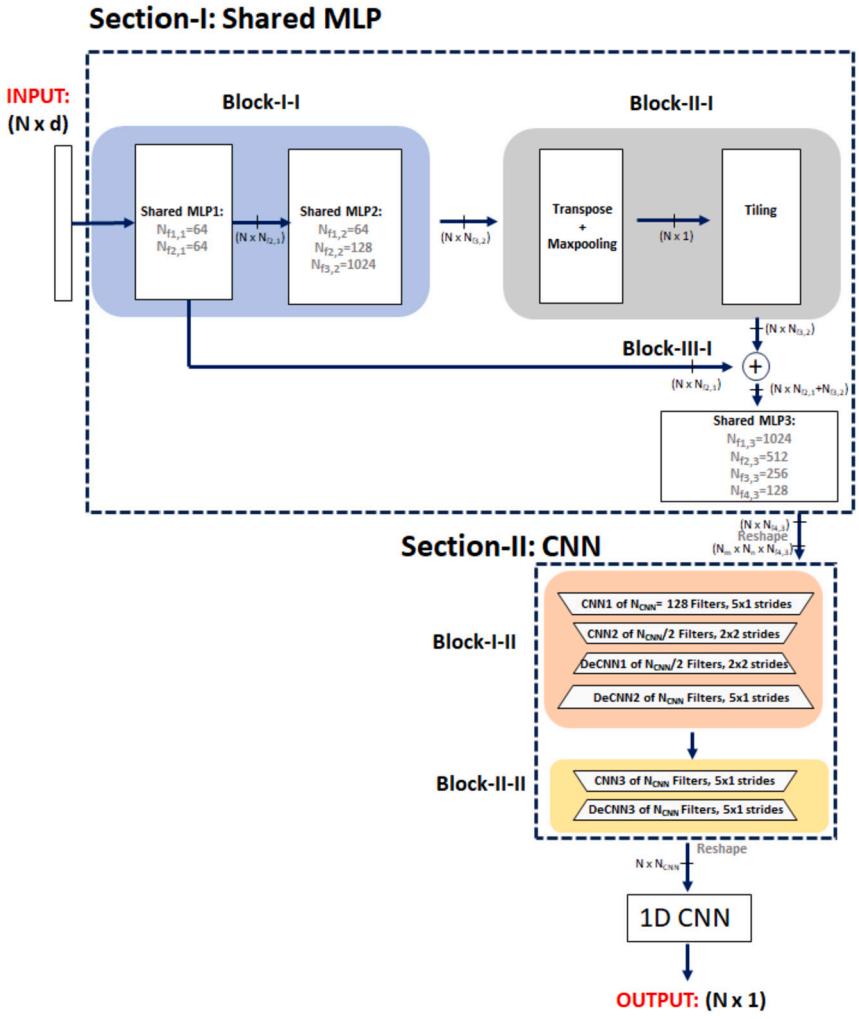


Fig. 4. Developed PCNN method for transonic flow-field predictions.

where W_{kj} is the set of filters for the j-th layer, for which a convolution operator is used with the input signal (x_j). As for the MLP, b_k is the bias factor.

The activation function of ReLu and sigmoid are also used for the CNN (Eq. (6) and (7)).

$$\phi(c_k) = \max(0, c_k) \tag{6}$$

$$\phi(c_k) = \frac{1}{1 + e^{-c_k}} \tag{7}$$

As part of this work, the input of the PCNN model consists of an $N \times d$ array, where N refers to the mesh points and d is the dimensionality of the problem, i.e. $d = 2$ for this axisymmetric nacelle example. Within Section-I, the model uses three MLP blocks (Block-I-I, Block-II-I and Block-III-I as presented in Fig. 4) that follow a PointNet architecture. The first block (Block-I-I) starts with two shared MLP layers (MLP1 in Fig. 4) with 64 filters (N_f) each. Then a second set of shared MLP layers (MLP2) of increasing number of filters $N_f = 64$, 128 and 1024 is used. The dimension in the output of the Block-I-I is $N \times 1024$. Since the number of grid points (N) is different to 1024, the PCNN cannot directly use a max-pooling layer. As previously noted, the calculation of a $N \times N$ tensor can be computationally expensive for large meshes. As such, the inherent permutation invariance of the original PointNet [15] is not preserved in the proposed PCNN. However, this is not required for this work because the underlying data was generated with a fully

structured grid. Nevertheless, it is important to highlight that this is a limitation of the PCNN model for CFD data compiled with unstructured grid strategies. As permutation invariance is not needed, the output from Block-I-I (Fig. 4), which is an $N \times 1024$ array, is transposed. This data is then used as input in the Block-II-I (Fig. 4). A max-pooling is applied along each of the rows to identify the dominant features in each of the N rows (noted as "Maxpooling" in Fig. 4). This results in an output of $Nx1$, which is followed by a layer where the output is expanded to $N \times 1024$ by tiling the original tensor (noted as "Tiling" in Fig. 4). It is important to note that the output of Block-II-I is not the extracted global feature because the max-pooling operator is not performed in a $N \times N$ tensor. Block-II-I is used to retain local features of the point cloud. This is then concatenated with the output from the MLP1 to preserve the geometrical information of the different shapes. This is known as an intermediate skip connection, and its objective is to retain information of the geometry from the point cloud as the deep learning architecture progresses. The proposed PCNN is a deep network, and therefore, the gradients used to update the weights during the training process can become very small. Without this skip connection, the network could stop learning the associated flow-fields from the different geometric shapes, i.e. input point clouds. This concatenation is used as input for Block-III-I (Fig. 4). This block uses 4 shared MLP layers (MLP3), in which the depth of the layers was progressively decreased from $N_f = 1024$ to $N_f = 512$, 256 and 128. The output from the last layer in MLP3 is modified into $N_m \times N_n \times 128$ tensors. The dimensions these tensors, defined by N_m and N_n, require that the product is equal to the original number of points handled by the PCNN architecture (Eq. (8)). As different values of N_m and N_n may fulfil Eq. (8), they are selected so that N_m/N_n is closer to 1 and $N_m > N_n$. In the case that N is a prime number, padding can be added to the original mesh points to enable the creation of the tensor defined with N_m and N_n.

$$N = N_m x N_n \tag{8}$$

The Section-II of the PCNN uses as input the output generated in Section-I (Fig. 4), and comprises two blocks (Block-I-II and Block-II-II). The Block-I-II consists of two encoding (CNN1, CNN2) and two decoding layers (DeCNN1, DeCNN2). For this architecture, CNN1 uses 128 filters, which are reduced to 64 in CNN2. The process is followed by the two decoding layers of DeCNN1 and DeCCN2 which increase the size to $N_m \times N_n \times 128$. The Block-II-II comprises of a single encoder and decoder (CNN3 and DeCNN3), and its structure is the same as the CNN1 and DeCNN2 layers. The output of Block-II-II is a tensor of size $N_m \times N_n \times 128$ which is modified to a 2D tensor with a column of size N and depth 128. This tensor is passed through a 1D CNN to change the size from $N \times 128$ to $N \times 1$. This output ($N \times 1$) is the predicted field-data of the point cloud provided as input to the PCNN model. Throughout the PCNN architecture, the ReLu activation function is used except for the final 1D CNN output layer where the sigmoid function was utilized.

An important aspect when proposing a novel deep learning method is to benchmark the predictive accuracy with previous existing approaches. For this purpose, the accuracy for off-surface prediction of three different methods is compared: the PCNN model, convolutional neural networks [23] as well as a modified PointNet architecture. Sureshbabu et al. [23] provided a description of the CNN method for aero-engine nacelle applications. In this study, the same encoder and decoder architecture is used, and as a result, the technical details are not included here. The modified PointNet (m-PointNet) is inspired by the original PointNet model proposed by Kashefi et al. [15]. This m-PointNet follows a similar structure as PointNet but the maxpooling layer is not performed in a $N \times N$ tensor, and T-Nets are not used. Instead, the max-pooling operator is performed in a $m \times N$ tensor, where m = 1024. The output of this section was resampled to an tensor of $N \times 1$ to produce the final flow-field output. The m-PointNet method has the same structure as Section-I of the PCNN architecture (Fig. 4). It is worth highlighting that none of the three methods, i.e. CNN, m-PointNet and PCNN, preserve permutation invariance. The spatial correlation in the domain is only retained in the CNN and PCNN approaches due to the use of convolutional kernels.

2.4. Data analysis

This work investigates the flow-field prediction capabilities of low order models. This is performed for off-surface predictions with special emphasis in the boundary layer, in which attached and separated flows are considered. Although a more detailed characterisation of the boundary layer could be acquired with $y^+ = 1$, this work uses $y^+ = 50$ with well-established wall-functions as it is sufficient to investigate whether deep-learning approaches can predict the boundary layer characteristics and separated flows.

Data-driven approaches for flow prediction are typically agnostic of the underlying governing equations [37]. As such, the predicted primitive variables may not be self-consistent as they are in physics-based methods [38,39]. Within this work, independent PCNN models were trained for each flow property and the incompressible formulation [40] was used to determine the integral characteristics of the boundary layer, i.e. incompressible displacement thickness (δ_i^*, Eq. (9)) and momentum thickness (θ_i^*, Eq. (10)) and boundary layer shape factor ($H_i = \delta_i^*/\theta_i^*$). For the calculation of δ_i^* and θ_i^*, the streamwise velocity component (u_s) was integrated along the wall normal direction (y_{wall}) through a trapezoidal numerical integration method. The compressible effects on the characteristics of the boundary layer are typically negligible for a Mach number below 5 [41]. This incompressible definition has been successfully used in the past to investigate the effect of Reynolds number on a normal shock wave-transitional boundary later interaction over a curved surface [42]. The outer edge of the boundary layer (δ and U_e) was determined based on a threshold of vorticity magnitude (ϵ, Eq. (11)). A second-order central differencing scheme was used to compute the velocity gradients and the vorticity field ($\vec{\omega}$). At the boundary elements a forward/backward difference scheme was applied instead. Typically with a change in ϵ from 10^{-3} to 10^{-4}, the boundary layer characteristics were not significantly affected and therefore $\epsilon = 10^{-3}$ was defined as the threshold to identify the outer edge of the boundary layer. Although the outer flow field is not strictly irrotational immediately downstream of a shock [43], this method was proved to be robust in the characterisation of the boundary layer for similar transonic flows [43]. As such, it is considered a useful approach within the scope of this work.

$$\delta_i^* = \int_0^{\delta} \left(1 - \frac{U}{U_e}\right) dy_{wall} \tag{9}$$

$$\theta_i^* = \int_0^{\delta} \frac{U}{U_e}\left(1 - \frac{U}{U_e}\right) dy_{wall} \tag{10}$$

$$U = U_e \Leftrightarrow \frac{|\vec{\omega}_e|}{|\vec{\omega}_{y_{wall}=0}|} \leq \epsilon \tag{11}$$

3. Results and analysis

This investigation is based on a compact aero-engine nacelle with a fixed $L_{nac}/r_{hi} = 3.1$ and $r_{te}/r_{hi} = 0.91$, where the other four intuitive variables of r_{max}, f_{max}, r_{if} and β_{nac} (Fig. 1) are varied to generate a wide range of shapes. The work focuses on civil nacelles for long-range applications with a flight Mach number of M = 0.85, a massflow capture ratio of MFCR = 0.7 and a Reynolds number based on the highlight radius of $Re_{r_{hi}} \approx 12 \times 10^6$ [32]. As this work is based on 2D axisymmetric aero-engine nacelles, the angle of attack is zero. Therefore, the input to the low order models only depends on the geometry and the operating conditions are fixed. This is already challenging due to the non-linearity of transonic aerodynamics.

This is a challenging aerodynamic design space to evaluate the capabilities of the proposed PCNN deep learning method for off-surface

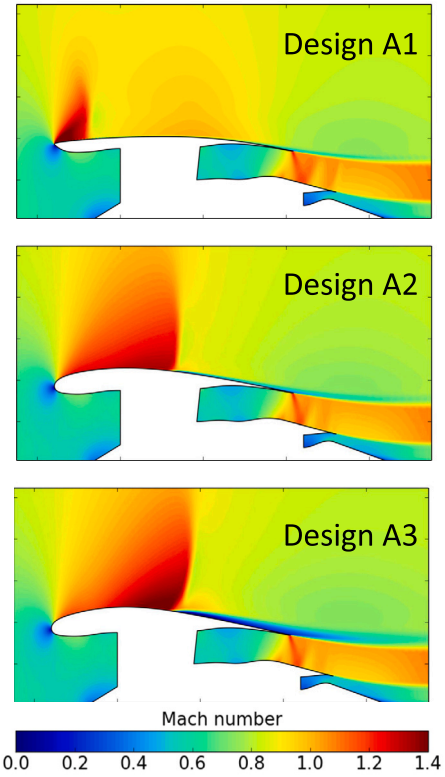


Fig. 5. Example of flow-field across the design space considered for compact aero-engine nacelles (M = 0.85).

flow-field prediction. In this respect, different flow features along the aero-engine nacelle are encountered. For example, Fig. 5 shows three different nacelles (Design A1, A2 and A3) within the investigated design space. Design A1 has relatively low values of the initial radius of curvature at the nacelle highlight (r_{if}), which results in a large initial acceleration along the forebody with a peak isentropic Mach number of approximately $M_{is} = 1.88$. Design A2 has a larger value of r_{if} and r_{max}, which induces a continuous acceleration along the nacelle forebody to terminate on a well defined shock-wave after the nacelle crest with a pre-shock Mach number of around 1.34. Both configurations, A1 and A2, have a boundary layer that is fully attached. However, across the design space there are combination of nacelle design variables (r_{max}, f_{max}, r_{if} and β_{nac} as defined in Fig. 1a) that result in shapes like Design A3 with clear undesirable aerodynamic flow characteristics. For this configuration, there is also a continuous acceleration along the forebody that terminates in a shock after the nacelle crest with a pre-shock Mach number of 1.45. For this shape (A3), the strength of the generated shock induces flow separation that extends until the trailing edge (Fig. 5).

A database of compact aero-engine nacelles was compiled with the computational method described in Section 2. For this purpose, a design space exploration (DSE) based on a latin hypercube sampling [44] of approximately 800 designs was performed. The data was randomly split into training, validation and testing sets. As part of this work, all the flow-field comparisons that are presented are based on the independent testing database which was not used as part of the training process of the deep learning models. The overall size of the dataset was approximately 24 million input points (800 meshes × 30,000 cells/mesh) for each flow variable, e.g. axial velocity, radial velocity or pressure. An independent model was generated for each flow variable. The training of the neural networks used the gradient descent ADAM optimiser [36]. It is a first-order gradient-based algorithm that minimizes the output of a loss function on a training dataset. For this work, the loss function was the root mean square error of the flow variable of interest. The learning rate during the training was fixed at 0.0005, with the exponential decay

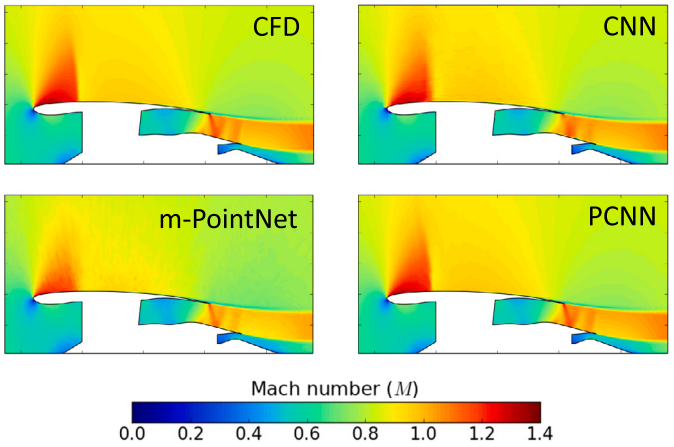


Fig. 6. Off-surface flow-field comparison of the three deep learning methods considered.

rates for the first and second moment estimates of 0.9 and 0.999, respectively. The batch size was 16 which, despite being relatively small, helped to learn the different shapes and was necessary due to the large changes in the transonic flow physics of the database. A subset of 640 samples, i.e. 80% of the DSE, was used for the training and validation process where the split was 576 and 64, respectively. The remaining 160 designs, i.e. 20% of the full DSE, were used as an independent dataset to test the model's accuracy. The PCNN model was trained for 2000 epochs until the loss function converged. This also ensured that the model was not overfitted, which was verified with the validation set.

$$\sigma_f = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(f^{CFD} - f^{ANN}\right)^2} \tag{12}$$

where f refers to the flow variable that is being trained with the PCNN.

It is well-known that there are no established rules to define the hyperparameters of a deep learning model because they usually depend on aspects such as the non-linearity of data, the amount of training samples or the dimensionality of the problem [7]. A trial and error based method was followed to identify a combination of hyperparameters that had a low MSE across the design space [35]. The details of the final PCNN used in this work were provided in Section 2.3.

3.1. Comparison of deep learning methods

To provide an initial overview of the off-surface predictive capabilities of the three deep learning approaches (CNN, m-PointNet and PCNN), the methods were trained to predict Mach number. Fig. 6 shows a comparison of the Mach number distribution for a configuration within the testing dataset. Relative to the CFD results, all three approaches qualitatively capture the principal flow characteristics associated with this particular nacelle shape. This includes an initial flow acceleration along the nacelle leading edge that results in the formation of a shock wave on the forebody. Although this work is focused on the nacelle aerodynamics, it is worth highlighting that the models also predict the complex aerodynamics of the exhaust system with similar features in the expansion and compression waves of the jet (Fig. 6). However, there are quantitative differences among the predictions. For instance, m-PointNet underpredicts the peak isentropic Mach number along the nacelle with a difference of approximately $\Delta M_{is} = -0.1$ compared with the numerical simulation (Table 2). The Point-Enhanced Convolutional Neural Network (PCNN) and the CNN methods both are within $\Delta M_{is} = 0.04$ of the CFD data (Fig. 6). All three deep learning methods predict the location of the shock wave within $\Delta X/L_{nac} \approx 0.005$. For this work, the axial location (x) where the shock wave occurs is based on identifying the x-location (x_s) where the first derivative of isentropic Mach

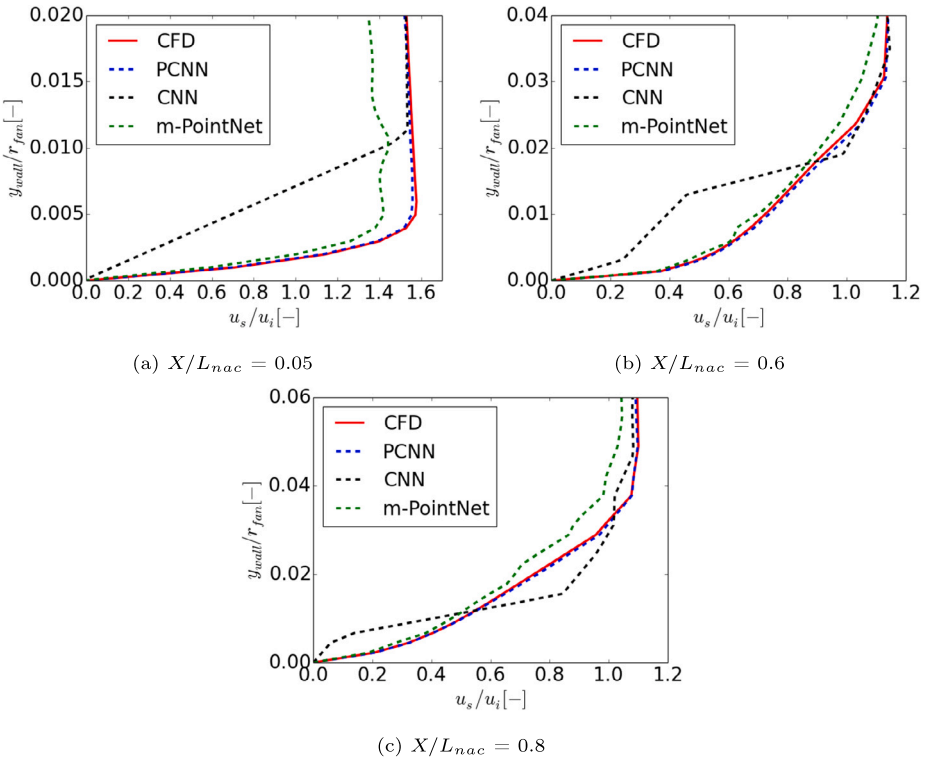


Fig. 7. Boundary layer comparison of the three deep learning methods considered.

Table 2
Difference between CFD and considered deep-learning methods in peak and pre-shock isentropic Mach number and shock location.

Method	peak M_{is}	pre-shock M_{is}	shock location (X/L_{nac})
CNN	+0.04	+0.001	+0.005
m-PointNet	-0.10	-0.09	+0.003
PCNN	+0.02	+0.03	+0.002

number with respect to x is minimised (Eq. (13)). The pre-shock isentropic Mach number is calculated by interrogating the Mach number at $x/L_{nac} = x_s/L_{nac} - 0.003$

$$x_s = \arg\min_x \left(\frac{dM_{is}(x)}{dx} \right) \tag{13}$$

The overall predictive accuracy of the three deep learning techniques (i.e. CNN, m-PointNet and PCNN) across the full design space is based on 160 aero-engine nacelles that compose the testing dataset. The root mean square error (Eq. (12)) is calculated by comparing the predicted Mach number distributions with the CFD data across the full off-surface domain. For the considered design space that contains a wide range of different transonic flow aerodynamics (Fig. 5), the PCNN model outperforms the CNN and m-PointNet. For the training dataset the predictive accuracy of CNN, m-PointNet and PCNN is $\sigma_M \approx 1.8 \times 10^{-5}$, 4.5×10^{-5} and 3.2×10^{-6}, respectively. For the training set the Point-Enhanced Convolutional Neural Network (PCNN) has a $\sigma_M \approx 1.1 \times 10^{-5}$, which is also considerable better that the accuracy of CNN and m-PointNet with $\sigma_M \approx 2.8 \times 10^{-5}$ and 1.3×10^{-4}, respectively. It is important to emphasize that the m-PointNet method is a simplification of the original PointNet proposed by Kashefi [15]. The main difference is that the m-PointNet does not use T-Nets [15] and the max-pooling operator is with a $m\ x\ N$ tensor, where $m << N$. Whilst this results in losing the spatial correlation in domain and permutation invariance, it enables the training of the model as the computational requirements of the max-pooling operator are reduced. Consequently, it is anticipated that this deviation may lead to reduced prediction capabilities compared to the original PointNet. Nevertheless, this study presents results using the m-PointNet method as an approach which can handle the necessary larger grids. For this reason, the PCNN method is proposed, which extends the m-PointNet approach by integrating convolutional neural networks, and thereby preserving spatial correlations in the domain. The predictive capabilities within the boundary layer for the three methods, i.e. CNN, m-PointNet and PCNN, are presented in Fig. 7. The comparisons are at three axial locations with $X/L_{nac} = 0.05, 0.5, 0.9$. As expected, the CNN method is not able to predict the boundary layer characteristics due to the low spatial resolution near the wall [13]. The m-PointNet, which has the same resolution as the CFD data, gives a better prediction of the boundary layer compared with the CNN approach but there are also notable local differences in the profile shape and magnitude. For example, for the axial location $X/L_{nac} = 0.05$, the streamwise velocity is underpredicted by approximately 14% at $y_{wall}/r_{fan} = 0.015$. Table 3 compares the boundary layer characteristics in terms of incompressible displacement thickness (δ_i^*, Eq. (9)), momentum thickness (θ_i^*, Eq. (10)) and boundary layer shape factor ($H_i = \delta_i^*/\theta_i^*$) between CFD and the deep-learning methods. Overall, the PCNN approach has the best agreement across various axial locations. A detailed analysis of the

Table 3
Relative difference between CFD and considered deep-learning methods in boundary layer characteristics.

X/L_{nac}	Method	δ_i^*[%]	θ_i^*[%]	H_i[%]
0.05	CNN	-214.4	37.6	-404.3
	m-PointNet	-10.8	-29.3	14.1
	PCNN	+1.9	+8.6	-7.7
0.6	CNN	-5.7	7.4	-6.9
	m-PointNet	-5.9	-8.1	+2.0
	PCNN	+3.2	+3.0	+0.2
0.8	CNN	+2.3	+30.1	-39.8
	m-PointNet	-9.9	-13.3	+3.0
	PCNN	+2.0	2.1	-0.1

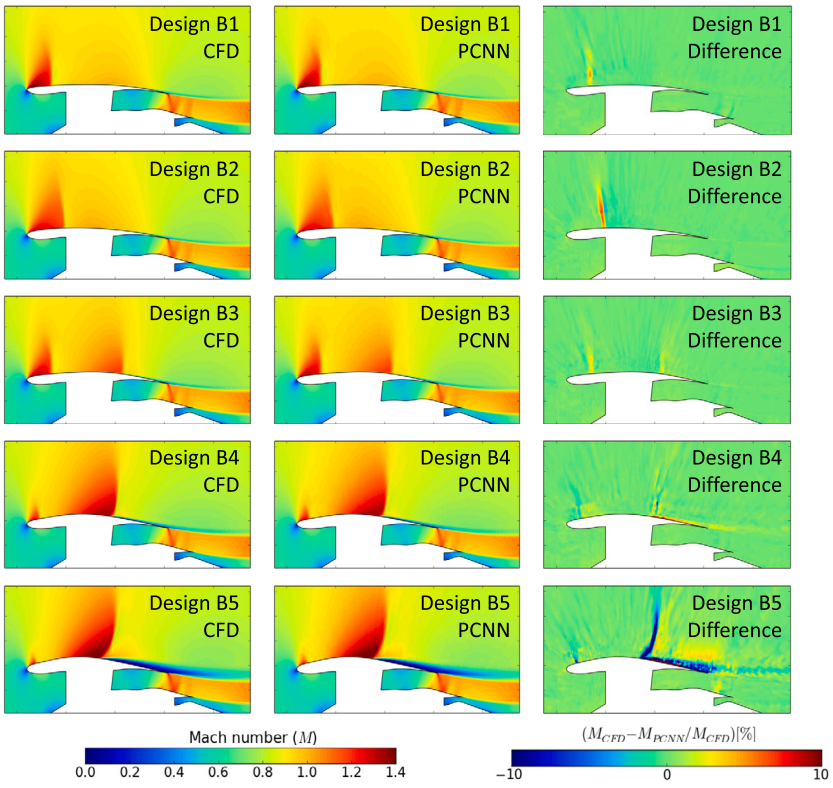


Fig. 8. Off-surface Mach number prediction and difference with CFD of the PCNN deep learning methods for a range of aero-engine nacelles.

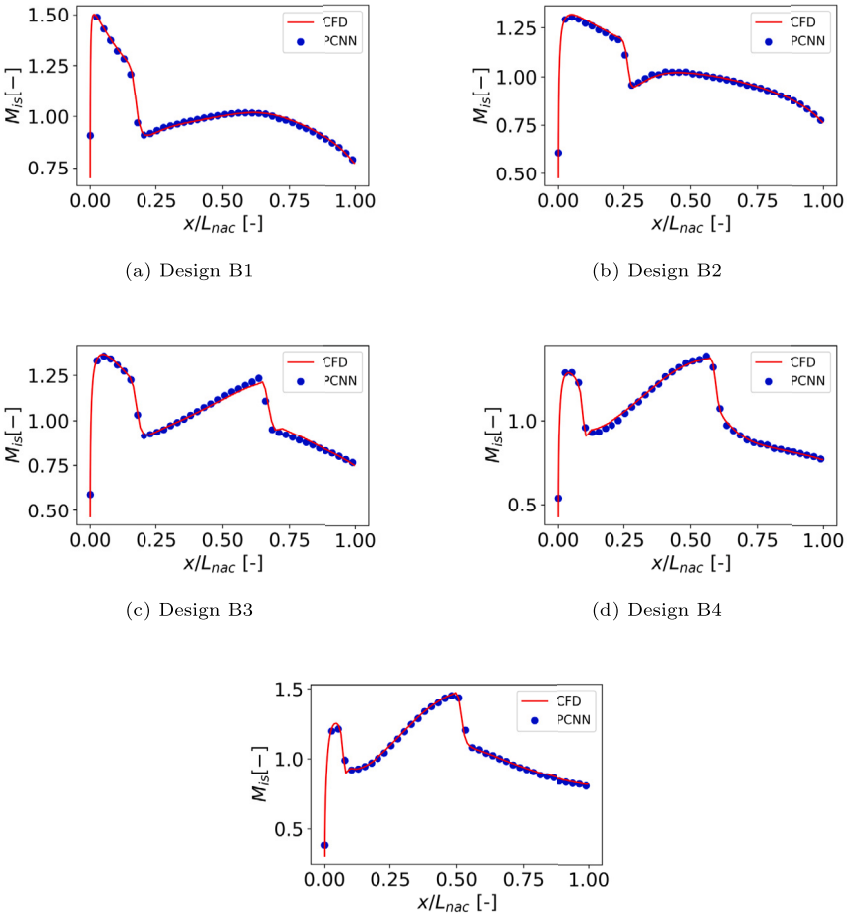
prediction capabilities of the PCNN is presented below for different aero-engine nacelles.

The CFD database generation and the training of the various models, i.e. CNN, m-PointNet, and PCNN, were conducted on an AMD EPYC 7543 32-core CPU with an NVIDIA A100 GPU. To benchmark the training duration, the total time (t_{CFD}) required to generate the CFD database of 800 nacelles was used as reference. The training times for the CNN, m-PointNet, and PCNN models were approximately 0.5%, 18%, and 20% of t_{CFD}, respectively. The m-PointNet and PCNN model required the longest training time due to its significantly more complex architecture and larger number of hyperparameters. Whilst this is a disadvantage, it is important to note that the training process is only performed once. As such, it is concluded that the proposed PCNN method is a feasible low order model for flow-field prediction in transonic applications.

3.2. PCNN for flow-field prediction

Having established confidence on the accuracy of the PCNN model across the design space (Section 3.1), and to test the predictive capabilities for off-surface flow-field prediction, Mach number distributions for a selected range of cases are presented in Fig. 8. By changing the nacelle design variables of r_{max}, f_{max}, r_{if} and β_{nac} (Fig. 1), a wide range of flow aerodynamics arises. The PCNN model captures these flow characteristics for this complex transonic aerodynamic conditions with M = 0.85. Design B1 is an example of a forward loaded configuration with a large flow acceleration with a maximum peak Mach number of approximately 1.5, in which the PCNN underpredicts it by 0.01. Across the domain, the low order model predicts the Mach number with a maximum error of $\Delta M = 0.05$. This difference is encountered at the shock location with a difference in its prediction of $\Delta X/L_{nac} \approx 0.002$. The root mean square error on the Mach number prediction across the domain is $\sigma_M = 0.004$. Relative to configuration B1, the nacelle B2 has a lower acceleration around the nacelle lip with a maximum peak Mach number of M = 1.32, which is successfully captured by the deep learning model within $\Delta M = 0.01$. For this configuration the maximum difference in the predicted Mach number is also located at the shock location, and has a root mean square error (σ_M) of 0.003 across the full domain. Finally, designs B3, B4 and B5 have the same double shock structure in which the first shock appears at the forebody and the flow reaccelerates and terminates with a second shock after the nacelle crest. Relative to B3, the shape B4 has a reduction in the normalised axial location of the nacelle crest (f_{max}) and an increment of the maximum radius (r_{max}), whereas the configuration B5 has a further reduction of f_{max} and increment of r_{max}. These changes of both design variables results in a weaker first shock and a stronger second shock as the designs are modified from B3 to B5. The second shock-wave for B5 has a pre-shock Mach number of about 1.45 that induces flow separation, and extends to the nacelle trailing edge. The nacelle B4 has shock-induced separation and reattachment. A thorough analysis of the boundary layer characteristics of these configurations is presented in Section 3.3. For these three configurations B3, B4 and B5, the PCNN deep learning approach captures the changes on Mach number distribution with σ_M of 0.002, 0.004 and 0.02, respectively. It is important to note that the largest prediction errors are at the shock positions for the considered B1-B5 designs (Fig. 8). This is caused by differences in the prediction of the shock location. For the most challenging design of B5, which has the strongest pre-shock Mach number, the error is of approximately 10%. For transonic aerodynamics with shock waves, it is acknowledged that there will be always differences between two methods if the shock location is predicted differently. This error is considered acceptable as the PCNN method predicts the shock position within $\Delta X/L_{nac} = 0.5\%$, and at worst 0.8%. This is typically in the range of validation studies for transonic applications [45].

These results show that the PCNN method can predict the off surface Mach number distributions for a wide range of nacelle designs. The method can also be used for other flow variables. For example, the same technique was used to train a model for static pressure. Similar levels of accuracy were obtained, and therefore they are not presented in this work. It is important to note that the input of the PCNN is an array of approximately 30,000 points for the x-coordinate and r-coordinate



(a) Design B1

(b) Design B2

(c) Design B3

(d) Design B4

(e) Design B5

Fig. 9. On-surface isentropic Mach number prediction of the PCNN deep learning method for a range of aero-engine nacelles.

Table 4
Difference between CFD and PCNN on-surface predictions for selected aero-engine configurations.

Design	peak M_{is}	1^{st} pre-shock M_{is} (ΔM_{is})	1^{st} shock location ($\Delta X/L_{nac}$)	2^{nd} pre-shock M_{is} (ΔM_{is})	2^{nd} shock location ($\Delta X/L_{nac}$)
B1	+0.02	+0.001	+0.004	-	-
B2	+0.01	+0.005	+0.003	-	-
B3	+0.005	0.0	+0.002	-0.02	+0.007
B4	-0.01	-0.04	-0.002	-0.002	-0.002
B5	+0.02	+0.02	-0.003	+0.002	-0.008

of the domain and the output is the flow property of interest in each point. Therefore, the grid-points that define the nacelle surface can be identified and the associated pressure extracted. This process allows for on-surface prediction with the off-surface data generated with the PCNN model. Previous studies focused on on-surface data for aero-engine nacelles. For example, Tejero et al. [29] used artificial neural networks for the prediction of isentropic Mach number along 2D axisymmetric aero-engine nacelles. It was concluded that fully-connected neural networks can identify the main flow characteristics in terms of peak Mach number, pre-shock Mach number and shock location.

Having a PCNN model for absolute static pressure (P_s), the isentropic Mach number (M_{is}) along the cowl is calculated with Eq. (14):

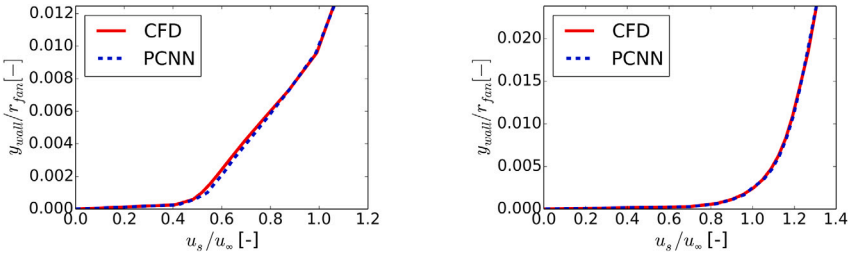
$$M_{is} = \sqrt{((\frac{P_t}{P_s})^{(\frac{\gamma-1}{\gamma})} - 1)(\frac{2}{\gamma - 1})} \tag{14}$$

where P_t refers to the freestream total pressure and γ is the heat capacity ratio.

Fig. 9 shows the M_{is} distributions along the cowl for the selected B1, B2, B3, B4 and B5 configurations. The PCNN method predicts a wide range of different flow characteristics for this complex non-linear aerodynamic problem. The main features of peak isentropic peak Mach number (M_{is}), pre-shock isentropic Mach number and shock location (X/L_{nac}) are captured within $\Delta M_{is} = 0.02$, $\Delta M_{is} = 0.04$, $\delta X/L_{nac} = 0.007$, respectively (Table 4). These comparisons demonstrate that the technique can predict surface quantities in addition to off-surface data.

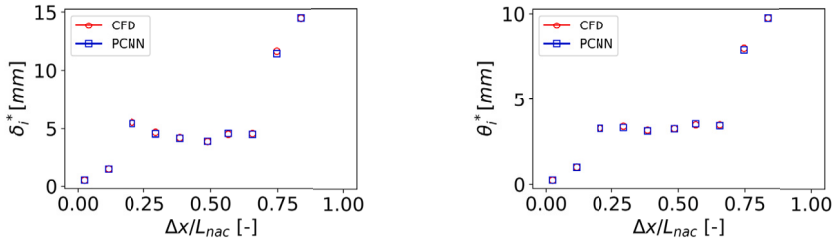
3.3. PCNN predictions in the boundary layer

Compared to more traditional CNN-based deep learning methods, the key novel aspect of the PCNN is that it can predict the flow in the region close to the wall. Thus, it is important to quantify the agreement between the CFD and the PCNN prediction within the boundary layer. For this purpose, two independent PCNN models were trained to predict the axial and radial velocities from which the boundary layer characteristics could be determined (Section 2.4). A subset of the nacelle designs

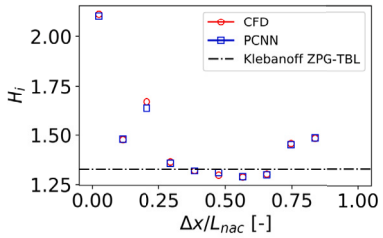


(a) $\Delta x/L_{nac} \approx 0.2$ (b) $\Delta x/L_{nac} \approx 0.75$

Fig. 10. Non-dimensional boundary layer profiles for Design B3.



(a) Incompressible displacement thickness (b) Incompressible momentum thickness



(c) Incompressible shape factor

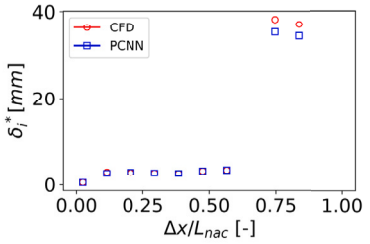
Fig. 11. Boundary layer characteristics for nacelle B3. The black dashed line in (c) marks the incompressible shape factor for a turbulent boundary layer with zero pressure gradient [46].

presented previously was used to determine the boundary layer characteristics along the nacelle cowl. The case studies included an attached boundary layer (nacelle B3), as well as two cases with boundary layer separation (nacelles B4 and B5).

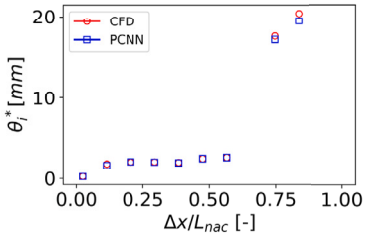
For the attached case (Design B3), the boundary layers for the CFD data and PCNN predictions are very similar (Fig. 10). For example, downstream of the first shock wave at $\Delta x/L_{nac} \approx 0.2$ (Fig. 10a) the boundary layer characteristics are comparable with a difference in the incompressible displacement thickness, momentum thickness and shape factor relative to CFD of 2.1%, 0.19% and 1.89%, respectively (Fig. 11). The boundary layer thickness increases significantly after the second shock-wave at $\Delta x/L_{nac} \approx 0.75$ (Fig. 10b). The PCNN method predicts the properties of δ_i^*, θ_i^* and H_i with a difference relative to CFD of 1.7%, 1.3% and 0.4%, respectively. Across, the different axial locations of the aero-engine nacelle the maximum difference between the PCNN and CFD methods is 2.1% in δ_i^* (Fig. 11a), 2.4% in θ_i^* (Fig. 11b) and 1.9% in H_i (Fig. 11c). At axial locations between both shocks, the shape factor was approximately 1.3 for both methods. This is similar to an incompressible turbulent boundary layer over a flat plate with zero pressure gradient [46].

Other designs with shock induced separation and reattachment (B4) and shock induced separation that extends until the trailing edge (B5) were used to demonstrate the capabilities of the new PCNN method to achieve near wall resolution. For the nacelle B4, the PCNN predicts at axial locations of $\Delta x/L_{nac} < 0.65$ an incompressible shape factor, momentum thickness and shape factor that is typically within 2.6%, 2.9% and 0.9% of the CFD results, respectively (Fig. 12). These are upstream of the boundary layer separation that is induced by the second shock (Fig. 9d). For axial locations downstream of the flow reattachment the maximum difference between both methods is of 7.6% in δ_i^* (Fig. 12a), 2.6% in θ_i^* (Fig. 12b) and 5.3% in H_i (Fig. 12c). The largest discrepancy in the integral parameters between PCNN and CFD is at $X/L_{nac} \approx 0.75$ (Fig. 12). However, the boundary layer profile is very similar between both methods (Fig. 13), which indicated that the PCNN method broadly capture the recovery of the boundary layer after the initial separation. For axial locations upstream of the flow separation, e.g. $X/L_{nac} \approx 0.48$ in Fig. 13, the PCNN prediction is excellent as expected from the agreement in the integral parameters of δ_i^*, θ_i^* and H_i (Fig. 12).

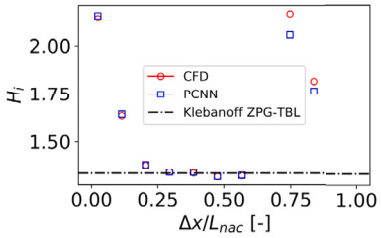
For the configuration B5, larger differences were typically identified in the incompressible integral values relative to the nacelles B3 and B4. Within the region of attached flow upstream of the second shock-wave, the maximum difference in δ_i^*, θ_i^* and H_i was 5.3%, 5.2% and 0.25%, respectively (Fig. 14). Overall, for attached boundary layers the new deep learning method (PCNN) successfully captures the incompressible integral characteristics. As such, it is a viable rapid and fast method for transonic wall-bounded flows. For this nacelle, the streamwise and wall-normal extent of the separation were slightly underestimated by the PCNN compared to the CFD (Fig. 15). For example, at $\Delta x/L_{nac} \approx 0.57$ the separation extent normal to the nacelle was approximately $0.005r_{fan}$ for the PCNN, whereas it was about $0.012r_{fan}$ for the CFD. Similarly, at $\Delta x/L_{nac} \approx 0.84$, the thickness of the separation bubble was smaller for the PCNN relative to the CFD.



(a) Incompressible displacement thickness

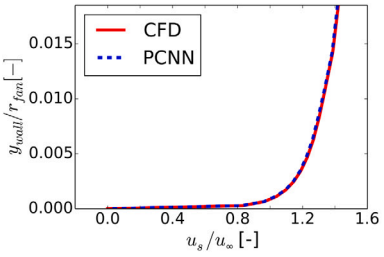


(b) Incompressible momentum thickness

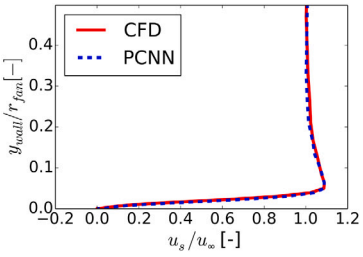


(c) Incompressible shape factor

Fig. 12. Boundary layer characteristics for nacelle B4. The black dashed line in (c) marks the incompressible shape factor for a turbulent boundary layer with zero pressure gradient [46].

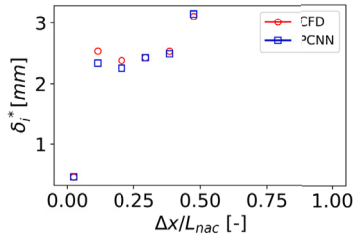


(a) $X/L_{nac} \approx 0.48$

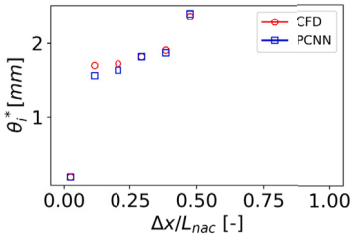


(b) $X/L_{nac} \approx 0.75$

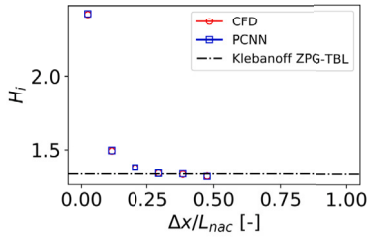
Fig. 13. Non-dimensional streamwise velocity profiles at different axial positions (X/L_{nac}) for the design B4.



(a) Incompressible displacement thickness

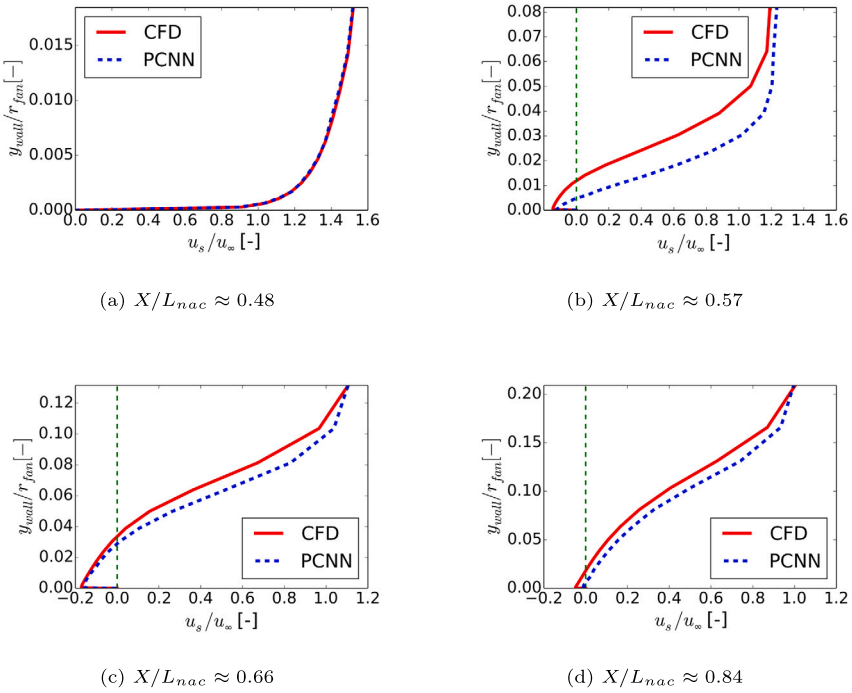


(b) Incompressible momentum thickness



(c) Incompressible shape factor

Fig. 14. Boundary layer characteristics for nacelle B5. The black dashed line in (c) marks the incompressible shape factor for a turbulent boundary layer with zero pressure gradient [46].



(a) $X/L_{nac} \approx 0.48$ (b) $X/L_{nac} \approx 0.57$

(c) $X/L_{nac} \approx 0.66$ (d) $X/L_{nac} \approx 0.84$

Fig. 15. Non-dimensional streamwise velocity profiles at different axial positions (X/L_{nac}) for the design B5. Green dotted-dashed line marks negative (separated) and positive (attached) regions of streamwise velocity. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

The results presented above demonstrate that the PCNN method accurately predicts a range of flow-field characteristics such as shock waves or shock-induced separation. Relative to previous CNN methods tested for transonic applications, the PCNN approach enables the prediction of the boundary layer. In addition, relative to the PointNet method, the scalability to large computational domains has been improved. This is a clear step forward to develop low order models with the same spatial resolution as the training data. On the other hand, it is important to acknowledge some limitations. The architecture does not preserve permutation invariance, making it suitable only for point cloud data generated using a fully structured grid approach. Currently, the method can be trained for one flow quantity, and in this study, independent PCNN models were trained for each primitive variable. However, this process does not adhere to a physics-based approach, and the underlying Navier-Stokes equations may not be satisfied by the PCNN predictions. For this purpose, more advanced loss functions based on physics-informed neural networks should be implemented.

4. Conclusions

This article proposes a deep learning method for transonic wall-bounded flows. The approach, named Point-Enhanced Convolutional Neural Network (PCNN), is based on state-of-the-art convolutional neural networks and PointNets, and reduces the computational memory requirements in the training process while maintaining spatial correlation in the domain and the same resolution in the predictions as the underlying data.

The novelty of this work is in the development of a low order model for flow-field prediction that preserves the spatial resolution of the training data. This enables the prediction of the boundary layer from which integral values such as displacement or momentum thickness can be calculated. The method has been demonstrated for a transonic flow test case. This was an aero-engine configuration comprising an intake, nacelle and separate jet exhaust. For the design space considered, a wide range of different transonic flow aerodynamics is encountered. The associated flow physics of transonic nacelles were evaluated with the PCNN model, in which the peak Mach number, shock location and pre-shock Mach number were successfully predicted. This deep learning approach is demonstrated for transonic wall-bounded flows, in which the incompressible displacement thickness, momentum thickness and shape factor are typically within 5% of the CFD. While the PCNN deep learning technique is demonstrated in nacelle applications, the results of this study hold significant potential within the aerospace sector, particularly in scenarios of transonic wall-bounded flows where it is required to have high resolution and accuracy within the boundary layer.

CRediT authorship contribution statement

Fernando Tejero: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sanjeeth Sureshbabu:** Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Luca Boscagli:** Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **David MacManus:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This project has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement number 101007598. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the Union.

Data availability

The data that has been used is confidential.

References

[1] L. Martinez, T. Nguyen, Challenges and opportunities of simulation-driven industrial design, Eng. Des. J. 22 (4) (2017) 289–306.
[2] R. Thompson, S. Patel, Machine learning in industrial design: a comprehensive review, Int. J. Organ. Des. Eng. 7 (3) (2018) 210–227.
[3] J. Chen, Y. Gao, Y. Liu, Multi-fidelity data aggregation using convolutional neural networks, Comput. Methods Appl. Mech. Eng. 391 (2022) 114490.
[4] B. Vachharajani, D. Pandya, Dimension reduction techniques: current status and perspectives, Mater. Today Proc. 62 (2022) 7024–7027.
[5] Z. Han, Y. Zhang, C. Song, K. Zhang, Weighted gradient-enhanced Kriging for high-dimensional surrogate modeling and design optimization, AIAA J. 55 (12) (2017) 4330–4346.
[6] C.C.D. Ronco, R. Ponza, E. Benini, Aerodynamic shape optimization of aircraft components using an advanced multi-objective evolutionary approach, Comput. Methods Appl. Mech. Eng. 285 (2015) 255–290.
[7] J. Li, X. Du, J. Martins, Machine learning in aerodynamic shape optimization, Prog. Aerosp. Sci. 134 (2022) 100849.
[8] J. Li, M. Zhang, J.R.R.A. Martins, C. Shu, Efficient aerodynamic shape optimization with deep-learning-based geometric filtering, AIAA J. 58 (10) (2020) 4243–4259.
[9] T. Franz, R. Zimmermann, S. Gortz, N. Karcher, Interpolation-based reduced-order modelling for steady transonic flows via manifold learning, Int. J. Comput. Fluid Dyn. 28 (3–4) (2014) 106–121.
[10] G. Immordino, A.D. Ronch, M. Righi, Deep–learning framework for aircraft aerodynamics prediction, in: AIAA AVIATION 2023 Forum, AIAA Paper 2023-3846, 2023.
[11] D. Hines, P. Bekemeyer, Graph neural networks for the prediction of aircraft surface pressure distributions, Aerosp. Sci. Technol. 137 (2023) 108268.
[12] V. Sekar, Q. Jiang, C. Shu, B.C. Khoo, Fast flow field prediction over airfoils using deep learning approach, Phys. Fluids 31 (5) (2019) 057103.
[13] C. Duru, H. Alemdar, O.U. Baran, A deep learning approach for the transonic flow field predictions around airfoils, Comput. Fluids 236 (2022) 105312.
[14] H. Wu, X. Liu, W. An, H. Lyu, A generative deep learning framework for airfoil flow field prediction with sparse data, Chin. J. Aeronaut. 35 (1) (2022) 470–484.
[15] A. Kashefi, D. Rempe, L.J. Guibas, A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries, Phys. Fluids 33 (2) (2021) 027104.
[16] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: deep learning on point sets for 3D classification and segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77–85.
[17] N. Birch, 2020 vision: the prospects for large civil aircraft propulsion, Aeronaut. J. 104 (1038) (2000) 347–352.
[18] F. Tejero, D. MacManus, I. Goulos, C. Sheaf, Propulsion integration study of civil aero-engine nacelles, Aeronaut. J. (2023) 1–15.
[19] F. Tejero, D. MacManus, J. Hueso-Rebassa, F. Sanchez-Moreno, I. Goulos, C. Sheaf, Aerodynamic optimisation of future civil aero-engines by dimensionality reduction and multi-fidelity techniques, Int. J. Numer. Methods Heat Fluid Flow (2022).
[20] F. Sanchez-Moreno, D. MacManus, J. Hueso-Rebassa, F. Tejero, J. Matesanz-Garcia, C. Sheaf, Optimization of installed compact and robust nacelles using surrogate models, in: Proceedings of 33rd Congress of the International Council of the Aeronautical Sciences, 2022.
[21] C. Wang, S. Wang, L. Wang, C. Cao, G. Sun, C. Li, Y. Yang, Framework of nacelle inverse design method based on improved generative adversarial networks, Aerosp. Sci. Technol. 121 (2022) 107365.
[22] F. Tejero, D. MacManus, J. Matesanz-Garcia, L. Boscagli, J. Hueso-Rebassa, C. Sheaf, Deep-learning for flow-field prediction of 3D non-axisymmetric aero-engine nacelles, in: AIAA AVIATION 2023 Forum, AIAA Paper 2023-3307, 2023.
[23] S. Sureshbabu, F. Tejero, F. Sanchez-Moreno, D. MacManus, C. Sheaf, Deep-learning methods for non-linear transonic flow-field prediction, in: AIAA AVIATION 2023 Forum, AIAA Paper 2023-3719, 2023.
[24] G. Drakoulas, T. Gortsas, G. Bourantas, V. Burganos, D. Polyzos, FastSVD-ML–ROM: a reduced-order modeling framework based on machine learning for real-time applications, Comput. Methods Appl. Mech. Eng. 414 (2023) 116155.
[25] A. Benaouali, S. Kachel, Multidisciplinary design optimization of aircraft wing using commercial software integration, Aerosp. Sci. Technol. 92 (2019) 766–776.
[26] X. Meng, X. Lu, H. Ye, B. Yang, F. Cao, A new self-augment CNN for 3D point cloud classification and segmentation, Int. J. Mach. Learn. Cybern. 15 (2024) 807–818.
[27] Y. Miao, Y. Sun, Y. Zhang, J. Wang, X. Zhang, An efficient point cloud semantic segmentation network with multiscale super-patch transformer, Nature Sci. Rep. 14 (2024) 14581.
[28] S. Shi, X. Wang, H. Li, PointRCNN: 3D object proposal generation and detection from point cloud, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 770–779.
[29] F. Tejero, D. MacManus, F. Sanchez-Moreno, C. Sheaf, Neural network-based multipoint, multi-objective optimisation for transonic applications, Aerosp. Sci. Technol. 136 (2023) 108208.
[30] R. Christie, M. Robinson, F. Tejero, D. MacManus, The use of hybrid intuitive class shape transformation curves in aerodynamic design, Aerosp. Sci. Technol. 95 (2019) 105473.
[31] Ansys Inc., 275 Technology Drive, Canonsburg, PA 15317, ANSYS FLUENT User's Guide.
[32] F. Tejero, M. Robinson, D. MacManus, C. Sheaf, Multi-objective optimization of short nacelles for high bypass ratio engines, Aerosp. Sci. Technol. 91 (2019) 410–421.
[33] K. Zuo, S. Bu, W. Zhang, J. Hu, Z. Ye, X. Yuan, Fast sparse flow field prediction around airfoils via multi-head perceptron based deep learning architecture, Aerosp. Sci. Technol. 130 (2022) 107942.
[34] A. Heidebrecht, D. MacManus, Surrogate model of complex non-linear data for preliminary nacelle design, Aerosp. Sci. Technol. 84 (2019) 399–411.
[35] M. Bouhlel, S. He, J. Martins, Scalable gradient-enhanced artificial neural networks for airfoil shape design in the subsonic and transonic regimes, Struct. Multidiscip. Optim. 61 (2020) 1363–1376.
[36] D. Kingma, J. Ba, ADAM: a method for stochastic optimization, in: 3rd International Conference on Learning Representation, 2015.
[37] L. Queiroz, F. Santos, J. Oliveira, M. Souza, Physics-informed deep learning to predict flow fields in cyclone separators, Digit. Chem. Eng. 1 (2021) 100002.
[38] F.V. Senhora, H. Chi, Y. Zhang, L. Mirabella, T.L.E. Tang, G.H. Paulino, Machine learning for topology optimization: physics-based learning through an independent training strategy, Comput. Methods Appl. Mech. Eng. 398 (2022) 115116.
[39] J. Fu, D. Xiao, R. Fu, C. Li, C. Zhu, R. Arcucci, I.M. Navon, Physics-data combined machine learning for parametric reduced-order modelling of nonlinear dynamical systems in small-data regimes, Comput. Methods Appl. Mech. Eng. 404 (2023) 115771.
[40] H. Schlichting, K. Gersten, Boundary-Layer Theory, Springer Science and Business Media, 2003.
[41] M.V. Morkovin, Effects of compressibility on turbulent flows, in: Mecanique de la Turbulence, vol. 367, 1962, p. 26.
[42] A. Coschignano, N. Atkins, H. Babinsky, J. Serna, Effect of Reynolds number on a normal shock wave-transitional boundary-layer interaction over a curved surface, Exp. Fluids 60 (2019) 1–12.
[43] L. Boscagli, D. MacManus, F. Tejero, K. Sabnis, H. Babinsky, C. Sheaf, Characteristics of shock-induced boundary layer separation on nacelles under windmilling diversion condition, AIAA J. (2023), https://doi.org/10.2514/1.J063209.
[44] J.C. Helton, F.J. Davis, Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems, Reliab. Eng. Syst. Saf. 81 (2003) 23–69.
[45] J. Chen, Y. Zhang, H. Zhao, G. Zhou, Numerical investigations of the NASA common research model with aeroelastic twist, J. Aircr. 55 (4) (2018) 1469–1481.
[46] P. Klebanoff, Characteristics of Turbulence in a Boundary Layer with Zero Pressure Gradient, Tech. Rep., NACA-TR-1247, National Bureau of Standards Gaithersburg Md, 1955.