Non-parametric Surrogate Model Method based on Machine Learning

Jiajun Cao[a], Qingbiao Li[b], Liping Xu[a], Rui Yang[c], Yuejin Dai[c]

[a] *Whittle Laboratory,Department of Engineering,University of Cambridge ,Cambridge, United Kingdom CB3 0DY*
[b] *Computer Lab,Department of Computer Science and Technology,University of Cambridge,Cambridge, United Kingdom CB3 0FD*
[c] *Shanghai Turbine Works Co., Ltd. Shanghai, China 200240*

Abstract

In this paper, a novel "non-parametric" surrogate model method is introduced. The method extracts geometric information from surface mesh of fluid domain using Graph Neural Network (GNN) and predicts the two-dimensional distributions of flow variables (in forms of contours) using Convolutional Neural Network (CNN). This method can extract relevant geometric information from surface mesh automatically, while existing data-driven surrogate model methods need manual parameterisation, which may generates a lot of uncertainties. Existing methods can only process geometries defined by their own parameterisation methods because the inputs of existing surrogate models are human-defined geometric parameters, while new methods can process any geometries with the same topology because its input is the surface mesh, which could potentially access more designs from different sources to create a larger database. In addition, this novel surrogate model is able to predict the contour of variables, not only several performance metrics. Predicting contours can also prevent over-fitting problem by balancing the data size of input and output. In this paper, this novel surrogate model method will be demonstrated with an example: low pressure steam turbine exhaust system. The new surrogate model uses 10 surface meshes

Email addresses: jc980@cam.ac.uk (Jiajun Cao), ql295@cam.ac.uk (Qingbiao Li), lpx1@cam.ac.uk (Liping Xu), yangrui@shanghai-electric.com (Rui Yang), daiyj3@shanghai-electric.com (Yuejin Dai)

of the system as input and predicts the power flux contour at the outlet of the last rotor. To build the surrogate model, altogether 582 numerical simulations have been created, which contains two types of geometries defined by different methods. Among them, 550 cases are used for training, and 32 cases are used for testing. The power output of the last two stages predicted by the surrogate model has 0.86 % difference compared with those of numerical simulations. The similarity score that measures the differences between the simulated and predicted contours is 0.9594 (1.0 being identical).

Keywords: surrogate model method, optimisation, graph neural network, parameterisation

1. Introduction

The motivation of this study is to develop a surrogate model method for multidisciplinary optimisation, especially for simulation domains with complicated geometries. Surrogate model can accelerate the multidisciplinary optimisation by making use of the result of previous simulation cases to evaluate performance of new designs. However, the accuracy and flexibility of existing surrogate model methods sometimes do not meet the requirement of industry, especially in some complicated application scenes. The authors of this paper think the bottleneck that prevents further improvement of surrogate model method is parameterisation. A lot of geometric information is lost during manual parameterisation, which are not compensable. Therefore, this study is trying to find a new way to bypass manual parameterisation with the help of new tools from machine learning.

Fig 1 highlights the differences between existing surrogate model methods and this novel surrogate method. Unlike existing surrogate model methods, the new method does not need parameterisation. It processes surface mesh directly and picks relevant geometric information according to their importance to the result. It also has the ability of predicting contours like numerical simulation. These features are achieved by the application of GNN, CNN and conditional

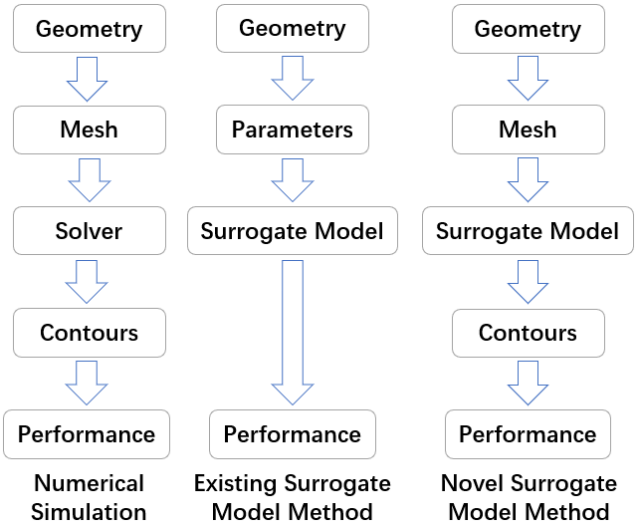

Figure 1: Comparison among numerical simulation, existing surrogate model method and novel surrogate method

variational autoencoder, which will be introduced in details in the fourth chapter.

Low Pressure Steam Turbine Exhaust System (LPES) is used as an example for demonstration in this paper. Before further discussion of the surrogate model structure, it is better to introduce the application scenarios and illustrate the necessity of the new method.

LPES is designed to recover the kinetic energy leaving the low pressure turbine and convert to static pressure for condenser. It usually consists of three parts: an axial-to- radial diffuser, an asymmetric collector and an extension. If building surrogate model for LPES with existing method, the first thing to do is to parameterise the geometry. Fig 2 shows a parameterisation method to define the geometry of the system, namely, the diffuser length ratio ($\frac{L_1}{L_0}$),diffuser area ratio ($\frac{A_1}{A_0}$),flow guide height ratio ($\frac{H_1}{L_0}$) ,diffuser turning angle($\Delta\theta$),tip kink angle(Θ_{tip}), hub kink angle(Θ_{hub}), hood height ratio ($\frac{H_1}{L_0}$) and hood width ratio ($\frac{W_1}{L_0}$). Though using so many geometric parameters to describe the geometry of LPES, there are still a lot of geometric features missing, for example, the

curvature of the diffuser, the height change of the collector, the width change of the extension and many more details. But if using more parameters, it will cause over-fitting problem in regression model.

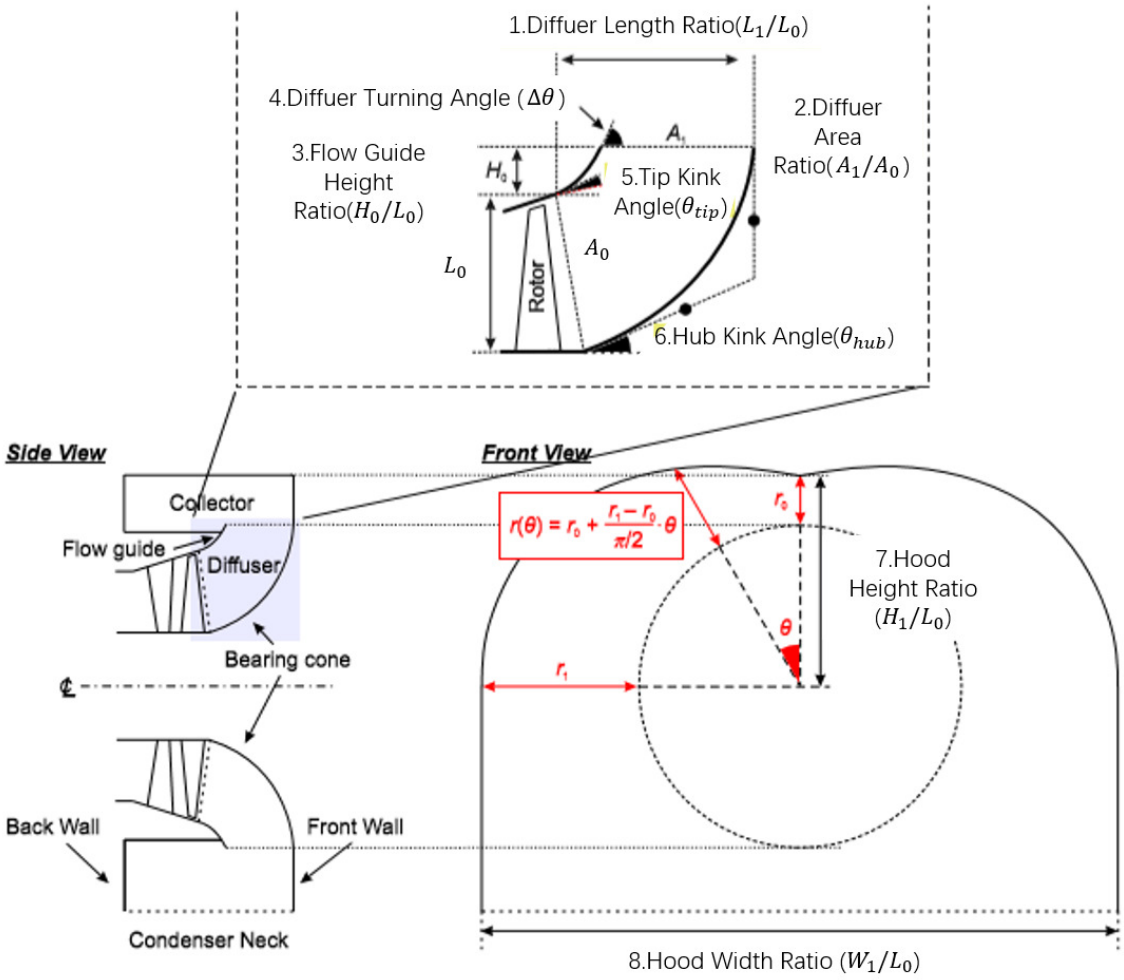

Figure 2: A typical down-flow type low pressure exhaust system for large steam turbine. Figure adapted from [1]

From this example, manual parameterisation is shown to be an inappropriate way to describe complicated geometry because it loses too much geometric details, which introduces too many uncertainties into the surrogate model.

Though **Surrogate model method** has been widely-used in designing and multidisciplinary optimisation, the studies of surrogate model still focus on regression model, not how to reduce the parameterisation uncertainties. There are a wide range of surrogate model methods appeared in the recent decade, which include:(1) Polynomial Response Surface Method [2]; (2) Kriging Model [3]; (3) Radial Basis Function and Extended Radial Basis Function [4]; (4) Artificial Neural Network [5]; (5) Support Vector Machine [6]. These studies, though have some achievements in regression model, are limited by the parameterisation. It not only prevents further improvement of prediction accuracy, but also limits the application flexibility. The surrogate model trained by geometric parameters can only process design defined by the same parameterisation method. But the input of new method is surface mesh, which means it can process any geometry as long as they have same topology structure.

Compared with existing surrogate model method, a new function of new method is predicting contours. This is achieved by **Convolutional Neural Network**. It is a type of deep neural networks, which is designed to process imagery. With the combination of convolutional layers, it can extract information from figures, and recognises the convolutional result with multi-layer perceptron [7]. In this study, CNN is used to predict the contour based on the latent distribution. Another reason why using CNN is to balance the data size of input and output. If directly predicting the performance of design base on geometric information extracted from surface mesh, the output is just a number, which is too small compared with data size of surface mesh. This will cause over-fitting problem.

This study uses **Graph Neural Network** to process surface mesh. It is motivated by CNN [8]. CNN can only process regular Euclidean data like figures, while GNN can process non-Euclidean domain by defining the connectivity of mesh points. Because of the ability of GNN to process non-Euclidean data, the input of the present surrogate model can be both unstructural mesh and structural mesh. With the connectivity matrix of mesh points, GNN can extract geometric information more comprehensive. According to the original GNN pa-

per [8], GNN is categorised into three types: Recurrent Graph Neural Network, Spatial Graph Neural Network and Spectral Graph Neural Network. In this study, the surrogate model is built based on Spectral Graph Neural Network because it is more suitable for large mesh size and extracting features[9]. Spectral Graph Neural Network is built on signal processing theory. The convolutional operation is done by Chebyshev polynomial approximation [10].

In conclusion, though surrogate model methods are widely used in optimisation, accuracy, flexibility and then generality are still limited by the parameterisation. The example shown in Fig 2 has very complicated geometry, which is difficult to parameterise. Inappropriate parameterisation will lose information, which makes surrogate model unreliable. And too many parameters will also generate many uncertainties. But with the new machine learning tools appeared in the recent years, surrogate model method is free from parameterisation. The GNN can extract geometric information from the surface mesh of the designs automatically. And the CNN can predict the 2D distributions rather than 1D variations, which can prevent over-fitting. The new method can also make use of different types of designs because the input of model is the surface mesh not user-defined parameters.

2. Methodology

The present surrogate model consists of two parts: optimisation part and machine learning part. Optimisation part is to build up the database for the surrogate model. The designs are generated by genetic algorithm (GA) because GA can explore the design space more comprehensive compared with gradient-based optimisation method.

To test the model's ability of processing geometries defined by different methods, two types of geometries defined by different levels of details are used. One is defined by 95 geometric parameters, the other by 66 geometric parameters.

The machine learning part is to train the surrogate model. It consists of two parts: mesh encoder and conditional variational contour decoder. The

mathematical theory of mesh encoder is from a mesh autoencoder [11], which is to compress the mesh to a latent vector and reconstruct the mesh with less similarity loss. In this study, the surface mesh of the design is defined by the coordinates of n vertices and edges, M=(V,E). V is the n vertices in the Euclidean space, which is an n×3 vector. The edges,E, are represented by the sparse adjacency matrix A. Its size is n×n, where $A_{ij} = 1$ denotes a connection between vertex i and vertex j. Otherwise, $A_{ij} = 0$.

The most important layers used in the mesh encoder is the fast spectral convolution layer. The mesh convolution operator $*$ is defined as a Hadamard product in Fourier space:

$$x * y = U((U^T x) \odot (U^T y)) \tag{1}$$

To reduce the computational cost, convolution is conducted by a kernel g_θ with Chebyshev polynomial of order K.

$$g_\theta (L) = \sum_{k=0}^{K-1} \theta_k T_k(\widetilde{L}) \tag{2}$$

where $\widetilde{L} = 2L/\lambda_{max} - I_n$ is graph Laplacian matrix. It is defined as L=D-A, where diagonal matrix $D_{ii} = \sum_j A_{ij}$. And θ_k are the coefficients of the Chebyshev polynomials. T_k can be expressed as:

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \tag{3}$$

with the initial condition $T_0 = 1$ and $T_1 = x$. This represents a Chebyshev polynomial of order k.

With the mesh filter shown above,the fast spectral convolution layer can be expressed as the following equation with n×F_{in} input and n×F_{out} input

$$y_j = \sum_{i=1}^{F_{in}} g_{\theta_{ij}}(L) x_i \tag{4}$$

where y_j means the j^{th} feature.

Another important layer used in the mesh encoder is the mesh sampling layer, which includes down-sampling layer and up-sampling layer in autoencoder [11]. Mesh sampling layer is to represent mesh in multi-scales so that

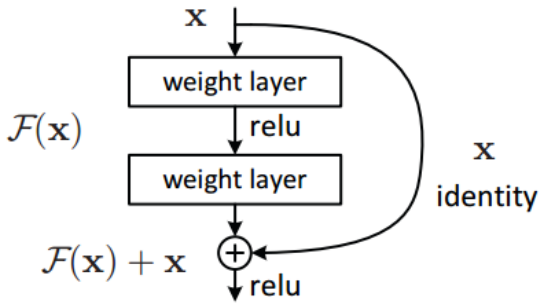

Figure 3: A basic block of Residual neural network. Figure adapted from [13]

convolution layer can capture the local and global geometric information. In this study, only down-sampling layer is used in the surrogate model. The down-sampling operation is conducted by a transform matrix $Q_{down} \in \{0,1\}^{n\times m}$, where m is the number of vertices in the original mesh and n is the number of vertices in the down-sampled mesh. $Q_{down}(p,q) = 1$ means the q-th vertex is kept during the down-sampling, while $Q_{down}(p,q) = 0$ means the vertex is discarded. The transform matrix is iteratively optimised to minimise the surface error by quadric matrices [12].

Contour decoder is built by CNN. Its structure is a Residual neural network (ResNet), a classical structure of artificial neural network, which is inspired by pyramidal cells in the cerebral cortex. ResNet simulates this by building shortcuts to skip some layer, rather than passing information layer by layer. Fig 3 shows a basic block of ResNet. F(x) is to fit the residual between x and target mapping H(x) rather than directly fitting H(x). It is easier for optimiser to minimise the residual to zero [13]. In this study, more hidden layers are added to fit the highly non-linear relationship between input and output, but the performance decreases rapidly with more layers. To solve degradation problem, ResNet is adopted because it can pass information from front layers to latent layers, which reduces the loss of information in more hidden layer.

The conditional variational decoder is an upgrade version of variational autoencoder(VAE). And VAE is an upgrade version of autoencoder(AE). AE uses

CNN to compress graphical data to a latent vector and then reconstruct the graph with the latent vector. The neural network is trained to reconstruct the graphs with less loss, compared to the input graphs. VAE uses variational inference to estimate the latent vector rather than directly encoding from input graph[14]. The latent vector z can be estimated by observation vector x using the following equation:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{5}$$

However, $p(z|x)$ is usually very difficult to compute directly. Therefore, another distribution $q(z|x)$ is used to approximate the $p(z|x)$. The Kullback-Leibler divergence is used to measure the difference between two probability distributions, which will be minimised during the training process. Conditional variational autoencoder adds conditions into the latent distribution so that different classes of input data are categorised into different groups. In this study, conditions (blade passages index) are added into the latent distribution twice to label the input data.

In the optimisation scenarios, surrogate model will be used as a performance evaluator in the framework of optimisation, which will replace numerical simulation and predict the performance of the design. But if the design is quite different from the existing designs in the database, this optimisation iteration still needs the result of additional numerical simulation, and then adds it to the surrogate model. Therefore, ideally, the optimisation will run in hybrid mode shown in Fig 4.

3. Optimisation Setup

In this study, optimisation is to build up the database for the surrogate model. The framework is a typical optimisation process based on genetic algorithm. In this section, the mesh generation process and numerical simulation setup are introduced in details.

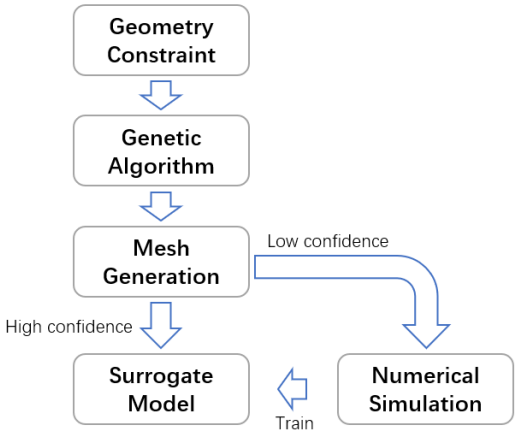

Figure 4: Framework of optimisation process with surrogate model

3.1. Mesh generation

Mesh generation starts from the coordinates of control points given by genetic algorithm. The control points will be used to generate non-uniform rational B-spline (NURBS) surfaces, and then evaluate the NURBS surfaces to generate the surface mesh, which is the input of surrogate model. Volume mesh will be generated by solving Laplace equation:

$$\begin{aligned} \frac{\partial^2 x}{\partial i^2} + \frac{\partial^2 x}{\partial j^2} + \frac{\partial^2 x}{\partial k^2} &= 0 \\ \frac{\partial^2 y}{\partial i^2} + \frac{\partial^2 y}{\partial j^2} + \frac{\partial^2 y}{\partial k^2} &= 0 \\ \frac{\partial^2 z}{\partial i^2} + \frac{\partial^2 z}{\partial j^2} + \frac{\partial^2 z}{\partial k^2} &= 0 \end{aligned} \tag{6}$$

where x,y,z are the coordinates of mesh vertices and i,j,k are indices of mesh vertices. The boundary condition is defined by the coordinates of surface mesh. Since Laplace equation represents a potential field, equipotential lines do not intersect and are orthogonal at vertices. The volume mesh can be generated by solve x,y, and z coordinates potential field respectively.The mesh generation method used in this study is able to generate mesh for the fluid domain with same topology, regardless the change of geometry.

There are two types of geometries. One is defined by 95 parameters, the other

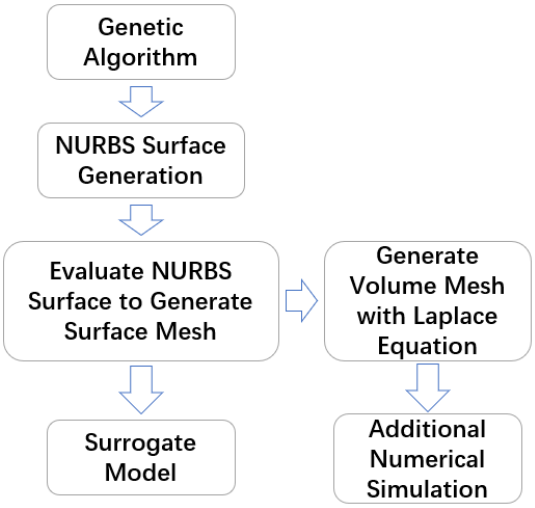

Figure 5: Framework of mesh generation process

one is defined by 66 parameters. The first one firstly defines the cross section of diffuser and collector, and then revolves it with ellipse equation to generate circumferential distribution. It also has asymmetric features in extension. The second one directly defines the cross section along the axial direction with ellipse equations. Two types of geometries can test the ability of processing geometries from different sources.

3.2. Numerical Simulation Setup

3.2.1. Simulation Domain

The fluid domain of simulation includes: two LP stages, axial-to-radial diffuser, collector and extension. Fig 6 shows the geometry of two low pressure stages, which is from a typical large steam turbine. It can generate a representative inlet profile for the exhaust hood. In the downstream, there are axial-to-radial diffuser, collector and extension, shown in Fig 7.

3.2.2. Simulation Setup

The simulation solver is Ansys CFX, which is a widely accepted commercial solver for research community and low pressure exhaust system. The simulation

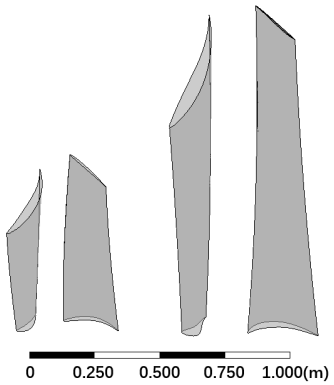

Figure 6: The geometry of the last two low pressure stages

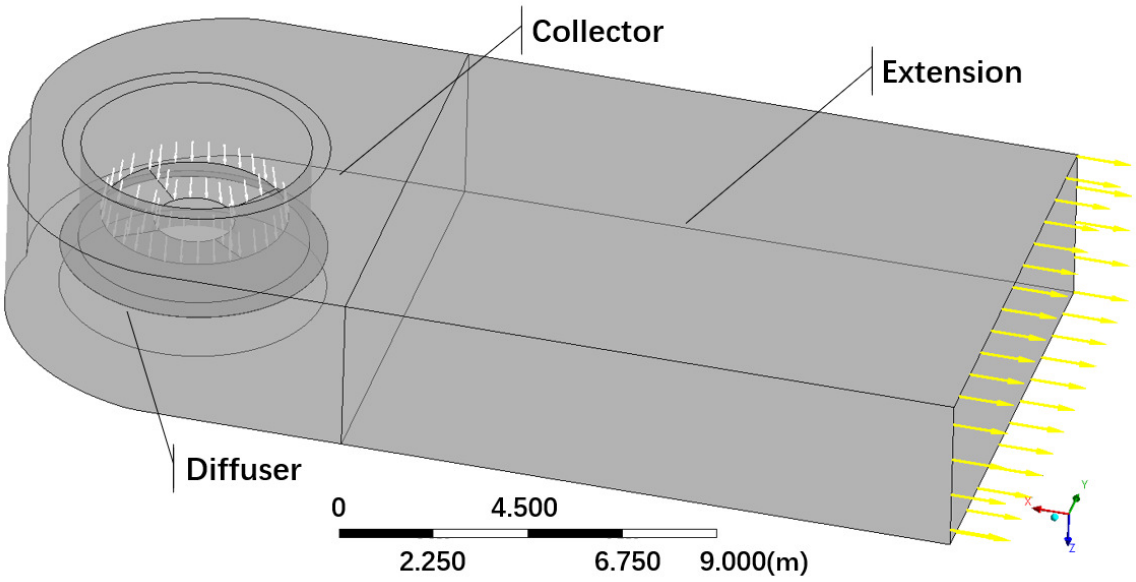

Figure 7: The geometry of exhaust hood

is a Reynolds-averaged Navier–Stokes (RANS) simulation. For the demonstrative purpose, $k-\epsilon$ turbulence model has been used [15].

The inlet flow condition is applied at the inlet of first stator, which has total pressure and total enthalpy information. And the outlet of simulation domain is at the outlet of extension, which has static pressure information. To perform the part load simulations, total pressure is reduced at the inlet to reduce the mass flow rate and work load. The outlet static pressure is 6.2kPa due to the steam property at the condenser. The fluid material in the simulation domain is steam3vl, which is from IAPWS package in the CFX and is widely used in the simulation of low pressure exhaust system.

Another setup worth mentioning is the interface treatment method between stages and the inlet of diffuser. Because the downstream of stages is asymmetric, it is necessary to model the circumferential non-uniformity. The interface treatment method used in this study is multiple mixing plane method, shown in Fig 8, which is proven to be a reliable treatment method in the literature [15]. In the simulations, only four blade passages are simulated to generate inlet flow condition for low pressure exhaust hood, which means one blade passage is responsible for a 90-degree section of the diffuser. The outlet boundary condition of blade passage is copied to cover the section. Multiple mixing plane method, though losing accuracy with 4 blade passages, reduces the computational cost considerably.

3.3. Processing of Numerical Simulation results

In this study, the performance of designs is measured by the power output of the last two stages. Therefore, the power contour at the outlet of the last rotor is extracted from numerical simulations, which will also be the output of the surrogate model. Because of the multiple mixing plane method, there are four blade passages for each simulation, and five workload conditions for each design.

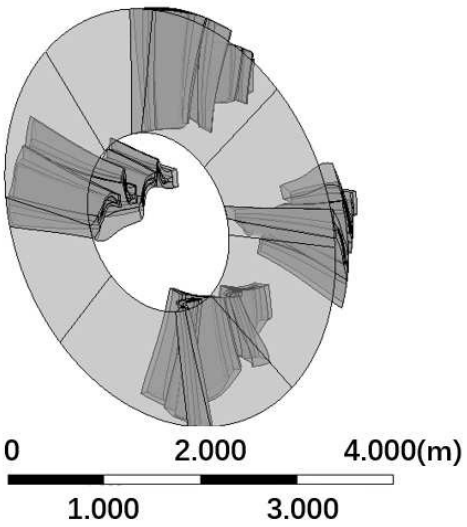

Figure 8: Demonstration of Multiple mixing plane method

4. Surrogate Model Setup

4.1. Neural Network Structure

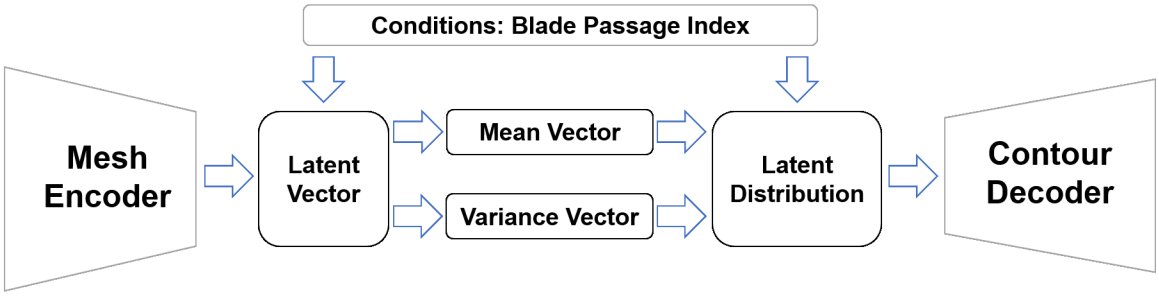

Figure 9: Neural Network Structure

The neural network is built under the framework of Pytorch. Fig 9 shows the main structure of network. Fig 10 shows the change of data size through the network. The input is 10 surface meshes, which have 195200 vertices. Therefore, the input data is the coordinates of vertices and adjacency matrix. The mesh encoder has 6 mesh encoder blocks. The size of filters in the front 4 blocks is 16, and it is 32 in the rear 2 blocks. This is to capture larger geometric features.

After the mesh encoder, the mesh is compressed into a 128×1 latent vector. Conditions (blade passage index) are added into the latent vector. Then, two fully-connected layers are used to estimate the mean vector and variance vector of latent distribution. Conditions are added into the latent distribution again. The latent distribution are reshaped to be the input of contour decoder. The first two blocks of contour decoder are ResNet blocks, and the rear two blocks are basic blocks.

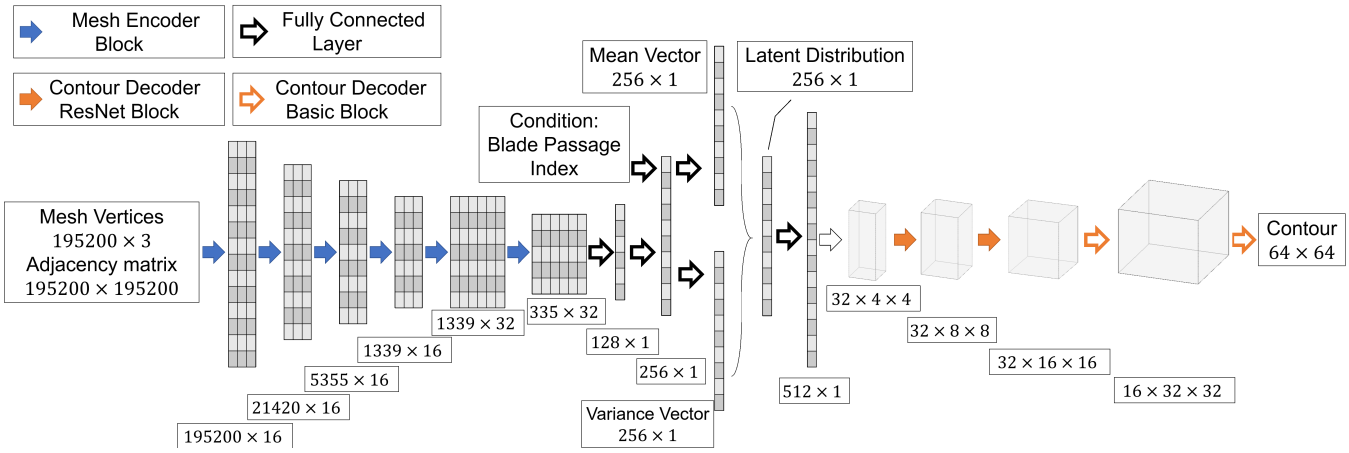

Figure 10: The Change of Data Size in the Network

4.2. Mesh Encoder Block

The structure of mesh encoder block is shown in Fig 11. One basic block is consist of a Chebyshev convolution layer, a normalisation layer (batch normalisation), an activation layer (the rectified linear unit function), a down-sampling layer and a pool layer (max pooling layer). The mesh encoder is consist of several basic blocks, which is depends on the size of mesh vertices and latent vector. More basic block means smaller latent vector, which may lose geometric information. But larger latent vector needs more training cases to prevent over-fitting. Inside the basic block, the Chebyshev convolution layer is to scan the mesh points and connectivity matrix with Chebyshev polynomial filter and convert them to a vector. The normalisation layer is to normalise the value of

vectors in the same batch. And the activation layer is to amplify the differences of values in the vector. The pool layer is to keep the most significant values and give up other values, which also reduce the dimension of the vector. After several basic blocks, relevant geometric information is picked to form the latent vector.

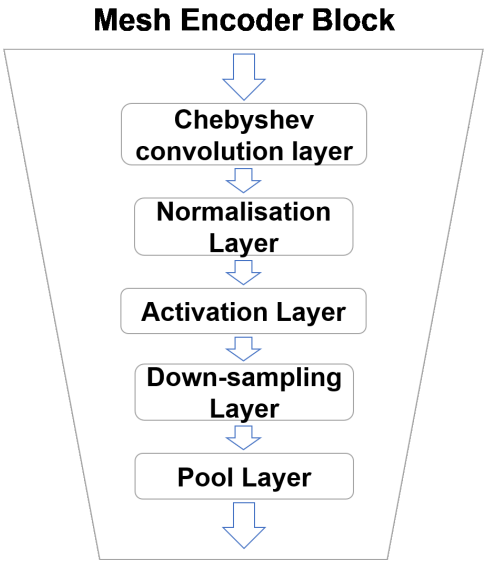

Figure 11: Mesh Encoder Basic Block

4.3. Conditional Variational Contour Decoder Blocks

Contour decoder consists of two types of blocks: ResNet block and basic block. ResNet block is shown in Fig 12, which has three layers and one shortcut. Basic block is shown in Fig 13 has four layers. The contour decoder has two ResNet blocks and two basic blocks,and the number of blocks can increase or decrease according to the size of latent distribution and the contours.

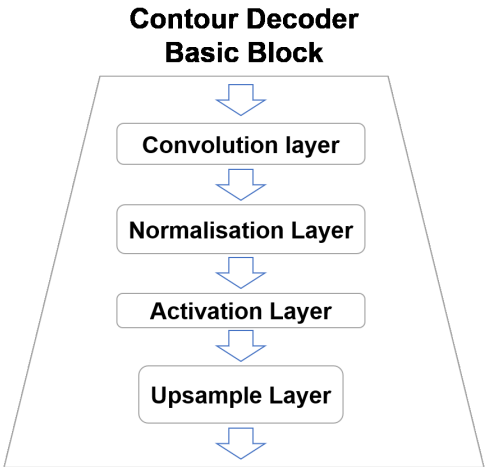

Figure 12: Contour Encoder Basic Block

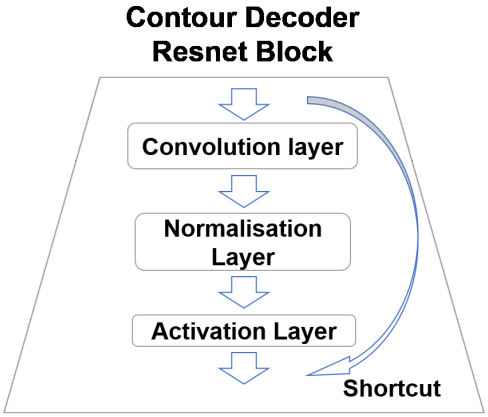

Figure 13: Contour Encoder ResNet Block

4.4. Loss Function

The loss function of the neural network is consist of three types of losses: mean squared error (MSE) loss , Kullback-Leibler divergence (KLD) loss and structural similarity loss. Mean sqaured error measures the average of the squares of the errors. It is defined by the following equation:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - X_i)^2 \tag{7}$$

KLD measures the difference between one probability distribution and the reference probability distribution. In variational autoencoder, KL loss is the sum of all the KLD between the components in latent distribution and the standard normal. With minimising the KL loss, the latent distribution is closer to the standard normal, which can improve the interpolation ability of the surrogate model. KLD is defined by the following equation [16]:

$$\begin{aligned} KLD(p,q) &= -\int p(x)logq(x)dx + \int p(x)logp(x)dx \\ &= \frac{1}{2}log(2\pi\sigma_2^2) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}(1 + log2\pi\sigma_1^2) \\ &= log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \end{aligned} \tag{8}$$

Since it is to measure the KLD between the components in latent distribution and the standard normal ($\sigma_2 = 1, \mu_2 = 0$), it can be simplified as the following equation for convenience:

$$KL_{loss} = \sum_{i=1}^{n}(\sigma_i^2 + \mu_i^2 - log(\sigma_i) - 1) \tag{9}$$

where μ is the mean vector, σ is the variance vector.

Structural similarity loss, or structural similarity index measure (SSIM), is a method to measure the similarity between two figures. It is defined as the following equation:[17]

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{10}$$

where μ_x, μ_y are the mean of x and y, σ_x^2, σ_y^2 are the variances of x and y, σ_{xy} is the covariance of x and y, c_1, c_2 are two variables to stabilise the division with weak denominator.

Finally, the loss function of the surrogate model is defined by the following equation:

$$loss = k_1 MSE + k_2 KLD + k_3(1 - SSIM) \tag{11}$$

The three coefficients k_1, k_2 and k_3 are determined by the experience of user.

5. Test and result

5.1. Test Setup

To test the surrogate model, 32 cases are randomly selected from 582 cases.Therefore, there is 128 test cases in total because each design has 4 blade passages. And during the training, there is a k-fold cross validation, which means 55 cases of the remaining 550 cases will be used for validation every 10 epochs.

The result of surrogate model are contours of power output at the outlet of the last rotor. There are 4 blade passages modelled for each simulation, and 5 work load condition for each design. This means there are 20 contours for one design.

As mentioned above,there are two types of sub-datasets. One has 333 training cases,and the other one has 249 training cases. During the training and test, two sub-datasets are mixed with each other to test the ability of processing different kinds of geometries. In the test, there are two test sub-datasets, both of which contains 64 cases.

5.2. Test Result

Fig 14 shows some typical results of the test cases, which shows some typical flow features of 5 workload conditions. The figure shows power flux distribution at the outlet of the last rotor. The performance of surrogate model is also evaluate by three performance metrics: mean sqaured error, structural similarity index measure and summed value error. Mean squared error measures the convergence of the training process. Structural similarity index measure shows the similarity of figures. A value of Structural Similarity index 1 means two figures are identical. The summed value error (SVE) measures the differences of predicted flow variables,which is defined by the following equation:

$$SVE = \frac{\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \widehat{y}_i}{\sum_{i=1}^{n} \widehat{y}_i} \tag{12}$$

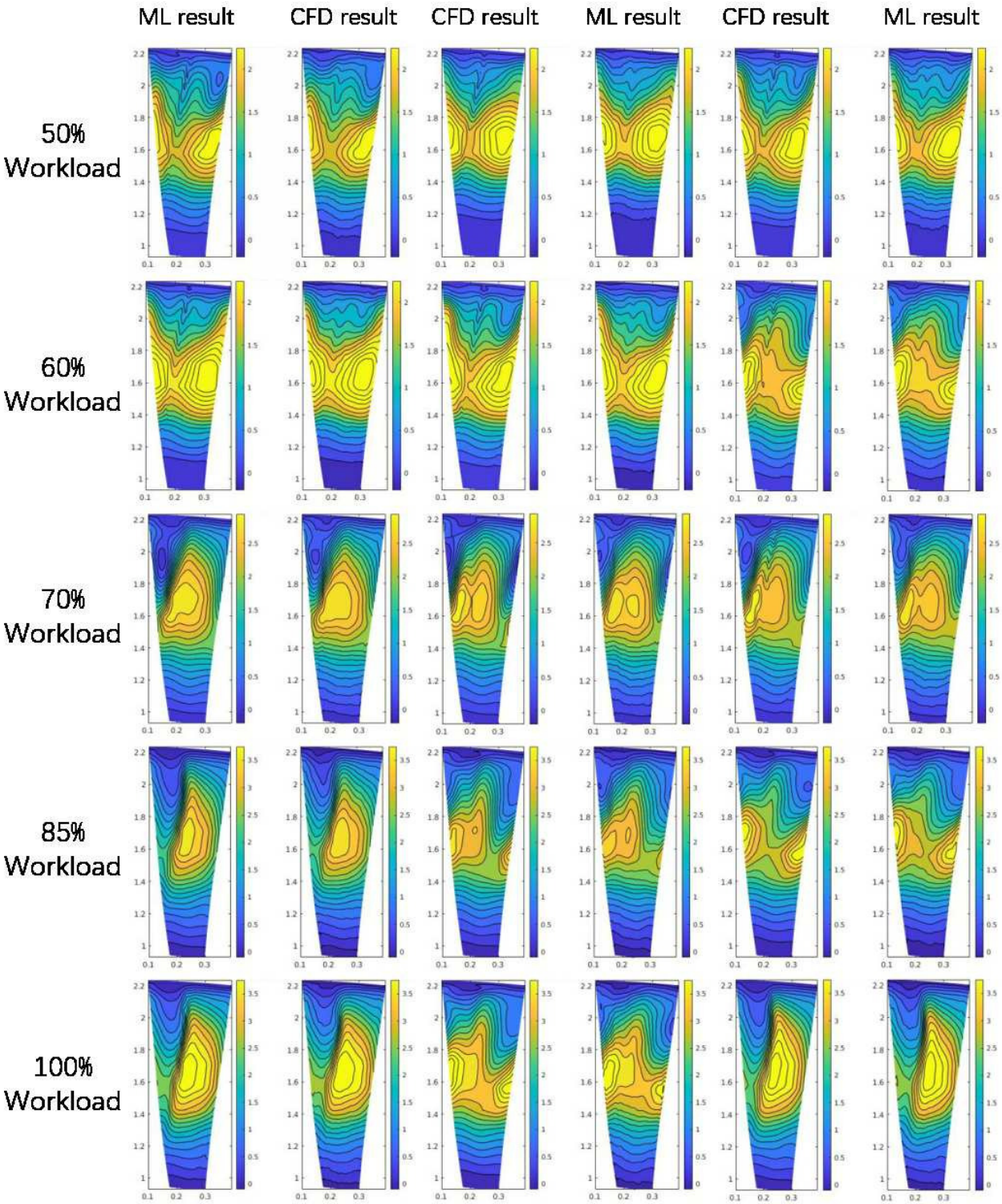

Figure 14: Contour results of computational fluid dynamics (CFD) and machine learning (ML) of some typical flow features Unit: $1 \times 10^4 kJ/s$

In the demonstration, it is the summed value of power go through the outlet of the last rotor. It is because the objective of optimisation is to increase the power output of the last two stages, which requires the summed values to be evaluated. This error also indicates the uncertainties in predicting the averaged value, like pressure, temperature, velocity, because the number of pixels are the same for these contours. Table 1 shows the result of all 128 test cases. Among them, there are two sub-datasets, both of which contains 64 cases. Summed value error shows it is able to predict performance for optimisation. Contour results, fig 14, shows it can predict most flow features of contours. Table 1 shows that the results of two test sub-datasets have some differences, but it is acceptable because they have met the requirement of optimisation.

Table 1: Summary of Result

Work Condition	Performance Metrics	Dataset	Sub-dataset1	Sub-dataset2
50%	MSE Loss	0.0047	0.0038	0.0056
	Similarity Index	0.9541	0.9520	0.9561
	Summed Value Error	0.0060	0.0068	0.0052
60%	MSE Loss	0.0068	0.0042	0.0094
	Similarity Index	0.9482	0.9517	0.9447
	Summed Value Error	0.0096	0.0122	0.0070
70%	MSE Loss	0.0017	0.0016	0.0019
	Similarity Index	0.9632	0.9602	0.9661
	Summed Value Error	0.0089	0.0095	0.0082
85%	MSE Loss	0.0016	0.0017	0.0016
	Similarity Index	0.9649	0.9618	0.9679
	Summed Value Error	0.0090	0.0104	0.0076
100%	MSE Loss	0.0019	0.0019	0.0020
	Similarity Index	0.9668	0.9641	0.9695
	Summed Value Error	0.0094	0.0108	0.0081
Mean	MSE Loss	0.0033	0.0026	0.0041
	Similarity Index	0.9594	0.9580	0.9609
	Summed Value Error	0.0086	0.0099	0.0072

6. Conclusion

This study presents a novel non-parametric surrogate model method, which is demonstrated with low pressure steam turbine exhaust system. New method directly takes surface mesh of fluid domain as input, which reduces the error generated by manual parameterisation and loss of geometric information. This has great advantage in building surrogate model for cases with complicated geometries. In the test, the average summed value error of 128 contours is 0.86 %, which has met the requirement of optimisation.

This method also shows high flexibility and compatibility. Because the input of new method is surface mesh, it can take geometries with same topology as database. This means it is compatible with geometry defined by different method. It is meaningful for further increase the size of database of surrogate model.

Compared with existing surrogate model method, this new method can also build a mapping relationship between surface mesh and 2 dimensional distribution of variables, which can be used to serve for many other purposes. In the test, the average similarity score of 128 contours is 0.9594, which can reproduce most flow features.

7. Discussion

Authors of this paper have proposed a concept: **Let the surrogate model knows what numerical simulation knows and predicts what numerical simulation can predict.** The input of numerical simulation usually contains mesh, boundary conditions and properties of material and the output of numerical simulation are usually 2 dimensional or 3 dimensional distribution of variables. Existing surrogate models are unable to process the mesh or 2D and 3D distributions so that they do not have the same amount of information as numerical simulations have, which limits the accuracy. Also, they do not have the ability of predicting 2D and 3D distributions like numerical simulations. With the new tools, this novel surrogate model is able to take mesh, boundary conditions and properties of material as input and predicts 2D and 3D distributions, which means it is a "complete" surrogate model of numerical simulation.

Though this novel surrogate model method is demonstrated with an aerodynamics optimisation example, this method is not limited in the area of aerodynamics. Since it can process unstructural mesh, it is also applicable in areas of finite element analysis and electromagnetic analysis. It has application potential in all areas that needs expensive numerical simulations.

However, there are still three important pieces of works to be done. The

first thing is to take the contour of inlet boundary condition as input. In the demonstration, the inlet boundary condition does not change too much, so this ability has not been tested. The second thing is the confidence level calculation, which can quantify the uncertainty of the surrogate model. This can be done by using classifier, but still needs more efforts. Thirdly, the differences between the results of two sub-datasets may due to the fidelity of numerical simulations, which needs further evidences.

References

[1] B. Ding, Aerodynamics of low pressure steam turbine exhaust systems, Ph.D. thesis (Nov 2018).

[2] R. H. Myers, D. C. Montgomery, C. M. Anderson-Cook, Response surface methodology: process and product optimization using designed experiments, John Wiley & Sons, 2016.

[3] S. Sakata, F. Ashida, M. Zako, Structural optimization using kriging approximation, Computer methods in applied mechanics and engineering 192 (7-8) (2003) 923–939.

[4] H.-M. Gutmann, A radial basis function method for global optimization, Journal of global optimization 19 (3) (2001) 201–227.

[5] T. Mengistu, W. Ghaly, Aerodynamic optimization of turbomachinery blades using evolutionary methods and ann-based surrogate models, Optimization and Engineering 9 (3) (2008) 239–255.

[6] A. Lal, B. Datta, Development and implementation of support vector machine regression surrogate models for predicting groundwater pumping-induced saltwater intrusion into coastal aquifers, Water Resources Management 32 (7) (2018) 2405–2419.

[7] M. Valueva, N. Nagornov, P. Lyakhov, G. Valuev, N. Chervyakov, Application of the residue number system to reduce hardware

costs of the convolutional neural network implementation, Mathematics and Computers in Simulation 177 (2020) 232 – 243. doi:https://doi.org/10.1016/j.matcom.2020.04.031.
URL http://www.sciencedirect.com/science/article/pii/S0378475420301580

[8] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks 20 (1) (2009) 61–80. doi:10.1109/TNN.2008.2005605.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, IEEE transactions on neural networks and learning systems.

[10] D. K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Applied and Computational Harmonic Analysis 30 (2) (2011) 129–150.

[11] A. Ranjan, T. Bolkart, S. Sanyal, M. J. Black, Generating 3D faces using convolutional mesh autoencoders, in: European Conference on Computer Vision (ECCV), 2018, pp. 725–741.
URL 'http://coma.is.tue.mpg.de/'

[12] M. Garland, P. S. Heckbert, Surface simplification using quadric error metrics, in: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, 1997, pp. 209–216.

[13] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition (2015). arXiv:1512.03385.

[14] D. M. Blei, A. Kucukelbir, J. D. McAuliffe, Variational inference: A review for statisticians, Journal of the American statistical Association 112 (518) (2017) 859–877.

[15] Z. Burton, G. L. Ingram, S. Hogg, A literature review of low pressure steam turbine exhaust hood and diffuser studies, Journal of Engineering for Gas Turbines and Power 135 (6) (2013) 062001.

[16] S. Kullback, R. A. Leibler, On information and sufficiency, The annals of mathematical statistics 22 (1) (1951) 79–86.

[17] Z. Wang, E. P. Simoncelli, A. C. Bovik, Multiscale structural similarity for image quality assessment, in: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, Vol. 2, Ieee, 2003, pp. 1398–1402.