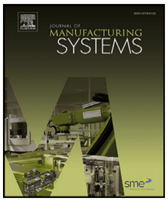
Journal of Manufacturing Systems 71 (2023) 681–694

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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



Technical paper

# Coaxiality prediction for aeroengines precision assembly based on geometric distribution error model and point cloud deep learning

Shang Ke [a,1], Wu Tianyi [b,1], Jin Xin [a,*], Zhang Zhijing [a], Li Chaojiang [a], Liu Rui [b], Wang Min [b], Dai Wei [b], Liu Jun [b,*]

[a] *Department of Mechanical Engineering, Beijing Institute of Technology, Beijing, 100081, China*
[b] *Department of Mechanical Engineering, City University of Hong Kong, 999077, Hong Kong, China*

ARTICLE INFO

*Keywords:*
Precision assembly
Aeroengine
Coaxiality prediction
Geometric distribution error
Point cloud deep learning

ABSTRACT

Assembly accuracy of aeroengines influences operation performance and service life. The coaxiality of the aeroengine is the main index of assembly accuracy and is also a core index to represent assembly quality. However, direct measurement of coaxiality is a difficult technical problem due to the sealed structure of the aeroengine casing system. A coaxiality prediction method is proposed to obtain coaxiality and assist assembly by geometric distribution error modeling and point cloud deep learning. The prediction process consists of three steps. In the beginning, the geometric distribution error model is established to construct the accurate dense point cloud of aeroengine part surfaces by the non-uniform rational B-splines (NURBS) method based on the coordinate measuring machine collecting information. Then, the mapping between the dense point cloud and coaxiality is established to obtain an assembly dataset by the virtual assembly. Finally, the dataset is fed to a new point cloud deep learning backbone, Self-channel cross attention point network, and realizes end-to-end coaxiality prediction based on the aeroengine surface point cloud. The geometric distribution error model is tested on the aeroengine simulated parts with 0.001 mm accuracy. The prediction method is verified on the aeroengine simulated parts and compared with other point cloud deep learning baselines. The method proposed in this paper realizes 93.17% prediction accuracy with 0.01 mm coaxiality precision which is a high performance and meets the requirements of industrial measurement. This paper provides an effective coaxiality prediction model for the aeroengine casing system, to improve the accuracy and efficiency of the aeroengine assembly.

## 1. Introduction

As a core geometric accuracy of aeroengine, coaxiality is closely related to vibration, dynamic imbalance, failure, etc. which directly affect operational reliability, product quality, and product life [1,2]. However, the coaxiality of multiple-bearing mounting holes is difficult to directly measure due to the complex and closed structure of the casing system. Therefore, the coaxiality of most aeroengines is obtained by the accuracy prediction method, as shown in Fig. 1. The difficult problem of high-accuracy coaxiality prediction seriously restricts product quality and assembly efficiency [3,4]. The core of the coaxiality prediction method is to establish a model that maps between measurable geometric distribution error of parts and unmeasurable coaxiality error of casing system. In recent years, many scholars have studied assembly accuracy prediction methods from two main aspects: error modeling and virtual assembly.

Geometric errors including form, orientation, location, and run-out were also considered in aeroengine assembly accuracy prediction methods. However, the geometric errors defined by ISO standards cannot comprehensively represent the non-uniform distributed microscopic error on surfaces Therefore, geometric distributed error (GDE) is proposed to represent the non-uniform distributed microscopic error in accurate digital twin modeling of precision microstructures [5]. The GDE modeling method is considered a basic problem of the aeroengine assembly accuracy prediction method in this paper. Zuo et al. [6] considered the influence of plane form error in the contact state of mating surfaces. Many scholars [7,8] established models of manufacturing error and deformation by the least square method and the random sample consensus algorithm. Schleich et al. [9] proposed the skin model to represent GDE and established models of manufacturing and assembly errors through point clouds and surface meshes. Sun

[*] Corresponding author.
*E-mail addresses:* goldking@bit.edu.cn (X. Jin), Jun.Liu@cityu.edu.hk (J. Liu).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.jmsy.2023.10.017
Received 28 August 2023; Received in revised form 10 October 2023; Accepted 30 October 2023
Available online 6 November 2023

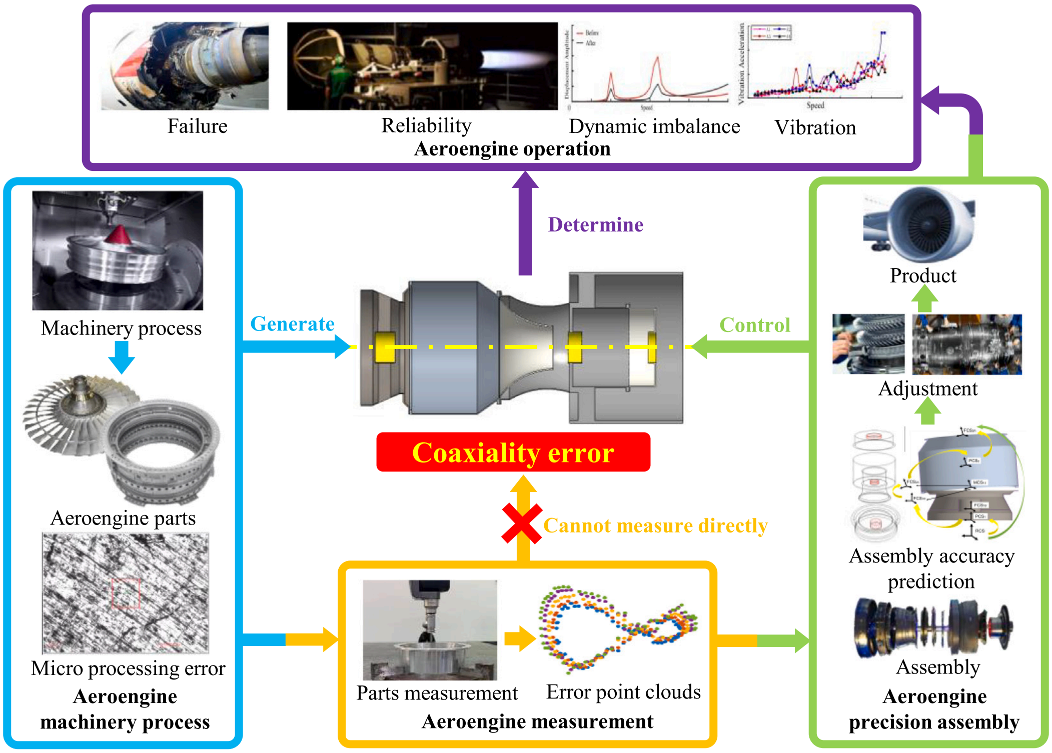


**Fig. 1.** The coaxiality error in aeroengines manufacture.

et al. [10] established the mathematical model of surface error by wavelet transform. Li et al. [11–15] established geometric and physical modeling of surface discontinuity features by high-definition metrology. Other scholars [16–19] established the outline of turbine blades based on non-uniform rational B-spline (NURBS) curves/surfaces.

However, there are many defects in the above studies. The meshed skin model which is proposed to estimate surface shape changes during the design, cannot comprehensively represent the error information between adjacent points, and the smoothness of the skin model is poor. Fitting methods such as wavelet transform and discrete cosine transform cannot guarantee that the surface model passes through every measurement point, so there is an irresolvable error between measurement points and the fitted surface. Meanwhile, the local GDE of the fitted surface is poorer than the interpolated surface. The NURBS curves/surfaces are used to establish blade ideal profiles during the design and tool trajectory planning. The influence of micro GDE on assembly accuracy is ignored.

On the basis of error modeling, scholars have studied virtual assembly methods to predict assembly accuracy. Dimensional chain calculation which replaces geometric errors with deviations to transform three-dimensional problems into one-dimensional problems, is one of the most widely used traditional methods in production [20,21]. The deviation propagation method and error propagation method based on homogeneous transformation matrix (HTM) and Jacobian-Torsor are often used in virtual assembly [7,22–24]. As an important part of the error propagation, the contact points calculation method directly affects the assembly accuracy prediction. Scholars [10,25,26] used difference surface method to obtain contact points and assembly position and posture. Other scholars [27–29] used the iterative nearest point (ICP) algorithm to obtain contact points and assembly position and posture.

However, there are many defects in the above studies. Dimensional chain calculation and deviation propagation without considering GDE cannot accurately predict coaxiality. For error propagation models based on HTM, Jacobian-Torsor, difference surface method, and ICP algorithm, the complex assembly structure contains a large number of specific model parameters so that it is difficult to model and tune, and it consumes time modeling and labor cost.

To guarantee coaxiality prediction accuracy, when assembling each pair of different surfaces, the engineer needs to adjust the model parameters according to the specific contact states. Each virtual assembly prediction is a new modeling process, which requires a lot of labor costs and computing power. Therefore, for the assembly accuracy prediction system, a new method that can realize the direct mapping between the point cloud and coaxiality is needed to eliminate the cumbersome modeling process and to improve the generalization of the assembly model.

Deep learning, a deep neural network with millions of parameters, is good at constructing relationships between mass discrete data with a single label [30]. The end-to-end prediction mode can effectively replace the manual modeling process of the contact calculation method. With the flourishing of point cloud acquisition equipment and computing power, point cloud deep learning is used in various industrial scenes, including robotics control [31], automation driving [32], and remote sensing [33]. Benefiting from discovering intricate structures in complex high-dimensional data [34], point cloud deep learning succeeds in constructing the end-to-end relationship between point cloud data and its physical meaning. The release of several public datasets [35–38] further boosts the development of deep learning on the point cloud. It derives the number of downstream tasks, including 3D shape classification [32], 3D point cloud segmentation [39], and 3D object detection [40]. In this paper, assembly workpiece surface point clouds are divided into groups according to the simulation calculation results, and we define the coaxiality prediction problem as a point cloud classification task.

Qi et al. [39] proposed PointNet, the geometric point-based deep learning method paradigm, and unstructured data processing pioneer, in 2017. Unlike the previous methods of converting point clouds into images [41–44], PointNet directly feeds point clouds into the deep neural network. It will efficiently decrease the computational costs caused by data processing and the calculation error caused by point cloud sparsity. To feed the unordered data format, PointNet proposed a symmetry

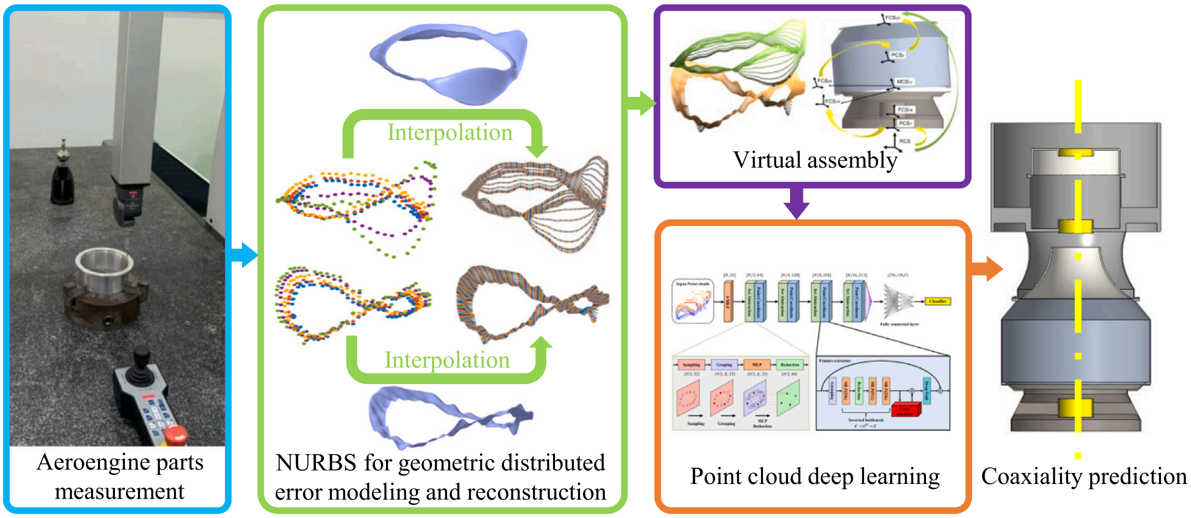

Interpolation
Interpolation
Aeroengine parts measurement
NURBS for geometric distributed error modeling and reconstruction
Virtual assembly
Point cloud deep learning
Coaxiality prediction

**Fig. 2.** The precision assembly coaxiality prediction method.

function to aggregate local and global information. Specifically, multilayer perceptrons (MLPs) are used to sum up all point cloud poses and achieve point cloud permutation invariance. Although PointNet realizes the independent point cloud feeding, the ignorance of neighbor points' local structure feature weakens the prediction accuracy. Therefore, Qi et al. [45] added a hierarchical architecture to PointNet and proposed PointNet++. The improvement can efficiently capture local geometric structures and achieve feature extraction efficiently and robustly. Based on these two backbones, DGCNN [46] considered each point as a vertex of a graph and aggregates point adjacent edge features via a channel-wise symmetric function. Inspired by the success of transformers in natural language processing and computer vision, Zhao et, al. [47] introduced the self-attention operator, the core of transformers, to the point cloud neural network and proposed Point Transformer. This architecture is remarkably effective in 3D point cloud processing and outperforms PointNet ++ by a large margin in a variety of downstream tasks. Nevertheless, PointNext [48] revisits the PointNet architecture and conducts improved model training and scaling strategies. The new architecture becomes the state-of-the-art (SOTA) method over Point Transformer and illustrates that the simple MLP is still excellent compared to other sophisticated modules.

Based on these works, a new point cloud deep learning network is proposed, named Self-channel cross attention point network, for effective point feature learning. It uses PointNet++ as a backbone and leverages an inverted bottleneck to perfect training results. To reinforce the information interaction between different feature maps, a new attention mechanism, the self-channel cross attention module, is proposed in this paper. In this module, self-attention and channel-wise attention are mixed to guarantee the extraction of local feature and global features.

In summary, there are still two problems with assembly accuracy prediction. Firstly, the GDE is not accurately and comprehensively considered in aeroengine assembly. Secondly, the current virtual assembly process which considers the GDE of complex mechanical products requires specific model tuning and calculation of different samples which increases time and labor costs and causes poor generalization. To end this, This paper proposes an end-to-end accurate assembly coaxiality prediction method based on GDE models and point cloud deep learning, as illustrated in Fig. 2. Firstly, the NURBS surface is used to reconstruct dense point clouds for comprehensive GDE of surfaces based on measured data; Secondly, the virtual assembly which maps between GDE and coaxiality is used to establish an assembly dataset; Thirdly, construct a point cloud deep learning network, establish a mapping relationship between point cloud data and assembly accuracy after surface discretization, and establish an assembly accuracy prediction model suitable for two surface assemblies with GDE; Finally, the effectiveness of the accuracy prediction model was verified by simulated aeroengine flange components as an example.

The rest of this paper is organized as follows. Section 2 introduces the GDE model and the process of constructing the relationship between the 3D point cloud and coaxiality. Section 3 illustrates the details of the Self-channel cross attention point network. Section 4 presents the experiment object and compares our method with other deep learning baselines. Section 5 draws the conclusions.

## 2. Modeling of aeroengine casing system based on GDE

In this section, a GDE model is established to map between the measurement point cloud and the reconstructed dense point cloud by the NURBS method. This model accurately and comprehensively describes the GDE of aeroengine surfaces. On this basis, a virtual assembly model is established to map between the reconstructed dense point cloud and the coaxiality of the aeroengine casing system by HTM and difference surface method. The details of the models will be introduced in this section.

### *2.1. Measurement for GDE of aeroengine casings*

Currently, the ISO 1101:2017 standard is used to calculate and evaluate geometric errors in the manufacturing industry. However, there is a problem that sometimes parts with the same geometric error evaluation value result in different assembly accuracy. And examples are shown in Fig. 3. During precision assembly, it is necessary to consider the impact of the micro GDE on the assembly accuracy.

To obtain the GDE data of aeroengines, the coordinate measuring machine (CMM) is selected as the primary measuring instrument in manufacturing. The initial point cloud is measured by CMM in this paper. However, the measurement efficiency of CMM is low, the measurement point cloud is sparse, and the error information of the measurement point cloud is not comprehensive. So, the dense point cloud which describes comprehensive geometrically distributed error is reconstructed in Section 2.2 by the NURBS method.

### *2.2. GDE modeling based on NURBS surface model*

The NURBS method is a three-dimensional interpolation model in computer graphics. The most prominent advantage of NURBS surfaces is they can accurately and uniquely represent a free-form surface

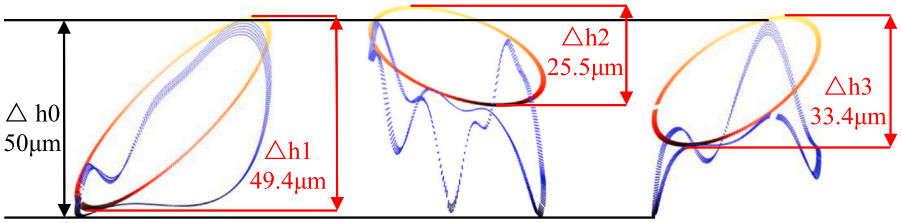
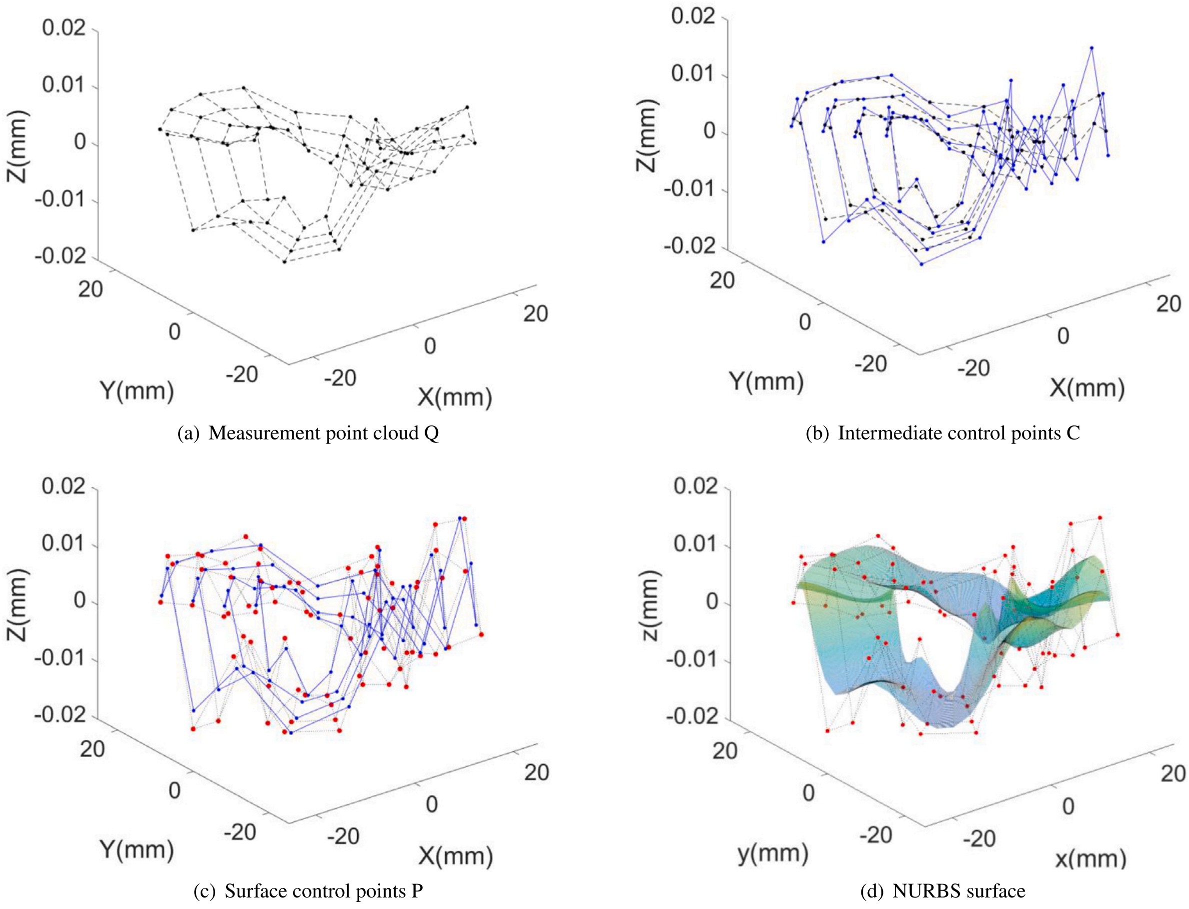


**Fig. 3.** Different assembly errors with the same tolerance zone.



(a) Measurement point cloud Q

(b) Intermediate control points C

(c) Surface control points P

(d) NURBS surface

**Fig. 4.** NURBS surface interpolation.

based on a unified mathematical form. The bicubic NURBS surface also has good locality and C2 continuity, which can accurately and comprehensively describe the micro GDE.

The NURBS method is used to predict GDE and upsample the measurement point cloud to obtain the dense point cloud. The dense point cloud can describe the comprehensive geometrically distributed error and provide a reliable data foundation for the assembly accuracy prediction model.

A NURBS surface with p degree in the u direction and q degree in the v direction is defined as Eq. (1).

$$S(u,v)=\frac{\sum_{i=0}^{m}\sum_{j=0}^{n}N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{m}\sum_{j=0}^{n}N_{i,p}(u)N_{j,q}(v)w_{i,j}} = \sum_{i=0}^{m}\sum_{j=0}^{n}N_{i,p}(u)N_{j,q}(v)P_{i,j}^{w} \quad ,0\leqslant u,v\leqslant 1 \tag{1}$$

Where, $u$ and $v$ represent knot parameters, $S(u,v)=[x(u,v),y(u,v),z(u,v)]$ represents the point on NURBS surface, $N_{i,p}(u)$ and $N_{j,q}(v)$ defined by the De Boor-Cox recursion formula [49] represent basis functions which determined by the knot vectors $\boldsymbol{U}$ and $\boldsymbol{V}$. $w_{i,j}$ represents the control point weight factor. $P_{i,j}$ represents the surface control point, $P_{i,j}^{w}$ represents the surface weighted control point.

For NURBS surface interpolation based on the measurement point cloud, the interpolation equation is assumed as Eq. (2).

$$Q_{k,l}=S(\bar{u}_k,\bar{v}_l)=\sum_{i=0}^{m}\sum_{j=0}^{n}R_{i,p}(\bar{u}_k)R_{j,q}(\bar{v}_l)P_{i,j} \tag{2}$$

Where, $Q_{k,l} = [x(\bar{u}_k,\bar{v}_l),y(\bar{u}_k,\bar{v}_l),z(\bar{u}_k,\bar{v}_l)],(k = 0,1, \dots,m;l = 0,1,\dots,n)$ represents the known $(m+1)\times(n+1)$ topological rectangular mesh measurement point cloud. $P_{i,j}$ represent unknown topological rectangular mesh control point cloud. As shown in Eq. (3), the cumulative chord length parameterization is used to calculate $\bar{u}_k$ and $\bar{v}_l$ to assign a pair of knot parameter values to each measurement point.

$$\begin{cases}\bar{u}_k=\frac{1}{(n+1)}\sum_{l=0}^{n}\bar{u}_k^{l} & k=0,1,\dots,m\\ \bar{v}_l=\frac{1}{(m+1)}\sum_{k=0}^{m}\bar{v}_l^{k} & l=0,1,\dots,n\end{cases} \tag{3}$$

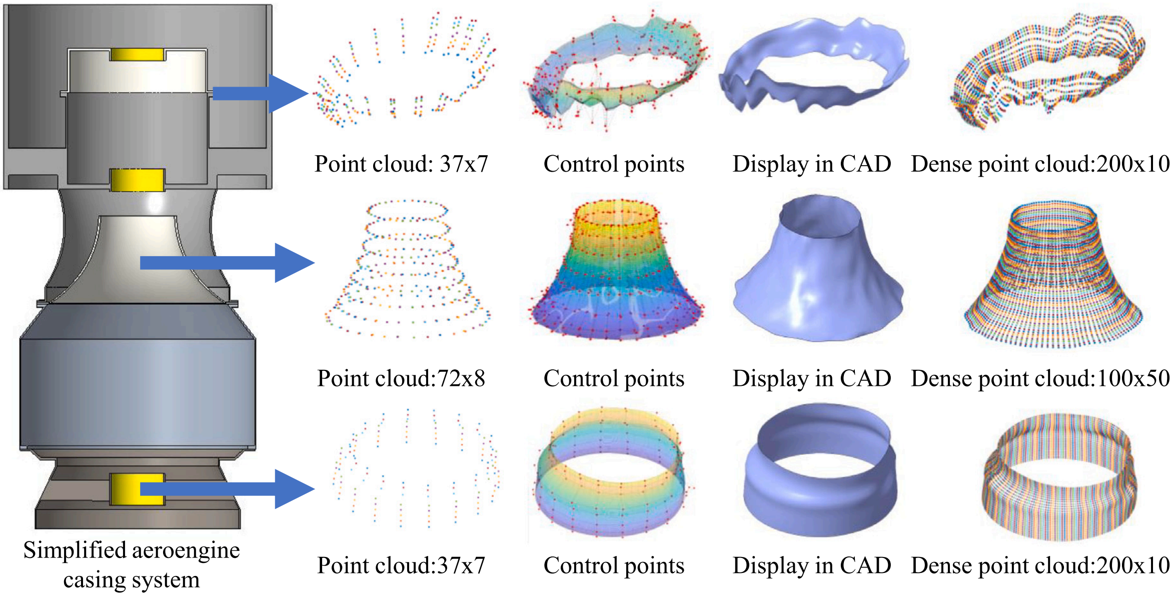


**Fig. 5.** NURBS modeling and GDE reconstruction.

As shown in Eq. (4), knot vector $\boldsymbol{U}$ is calculated by average technique [49] based on $\bar{u}_k$ . The calculation method for $\boldsymbol{V}$ is the same as for $\boldsymbol{U}$.

$$\begin{cases} u_0 = \cdots = u_p = 0 \\ u_{m-p} = \cdots = u_m = 1 \\ u_{j+p} = \frac{1}{p}\sum_{i=j}^{j+p-1} \bar{u}_i \quad , i = 1, 2, \ldots, m-p \\ \boldsymbol{U} = (\underbrace{0, 0, \ldots, 0}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{1, 1, \ldots, 1,}_{p+1}) \end{cases} \tag{4}$$

NURBS surface is a tensor product surface. So, Eq. (2) can be rewritten Eq. (5).

$$\begin{aligned} Q_{k,l} &= \sum_{i=0}^{m}\left[\sum_{j=0}^{n} P_{i,j} R_{j,q}(\bar{v}_l)\right] R_{i,p}(\bar{u}_k) \\ &= \sum_{i=0}^{m} C_{i,l}(\bar{v}_l) R_{i,p}(\bar{u}_k) \end{aligned} \tag{5}$$

Where, $C_{i,l}$ represents control points of the isoparametric curve on NURBS surface at fixed $v = \bar{v}_l$. All $C_{i,l}$ form intermediate control points of the NURBS surface. $C_{i,l}(\bar{v}_l) = \sum_{j=0}^{n} P_{i,j} R_{j,q}(\bar{v}_l)$ is a series of equations for interpolation with intermediate control points at fixed $i$ and changing $l$.

Therefore, the calculation of surface control points $P_{i,j}$ can be described as the following:

(1) In the u direction, do n+1 curve interpolations through $\{Q_{0,l}, Q_{1,l}, \ldots, Q_{m,l}\}(for \quad l = 0, 1, \ldots, n)$ based on $\boldsymbol{U}$ and $\bar{u}_k$. And intermediate control points $C_{i,l}$ are calculated, as shown in Fig. 4(b);

(2) In the v direction, do m+1 curve interpolations through $\{C_{i,0}, C_{i,1}, \ldots, C_{i,n}\}(for \quad i = 0, 1, \ldots, m)$ based on $\boldsymbol{V}$ and $\bar{v}_l$. And control points of surface $P_{i,j}$ are calculated, as shown in Fig. 4(c);

The GDE can be represented as a NURBS surface model, as shown in Fig. 4(d).

GDE models of three surfaces on a simplified aeroengine casing system are established, as shown in Fig. 5. And the GDE is amplified and displayed. Then, dense point clouds are calculated with $\bar{u}$, $\bar{v}$ based on the NURBS surface.

In summary, the GDE model is established based on the NURBS method. The micro GDE is comprehensively and accurately reconstructed as the dense point cloud. And, the accuracy of the GDE model is verified within 0.001 mm accuracy on the aeroengine simulated parts in Section 4.2.

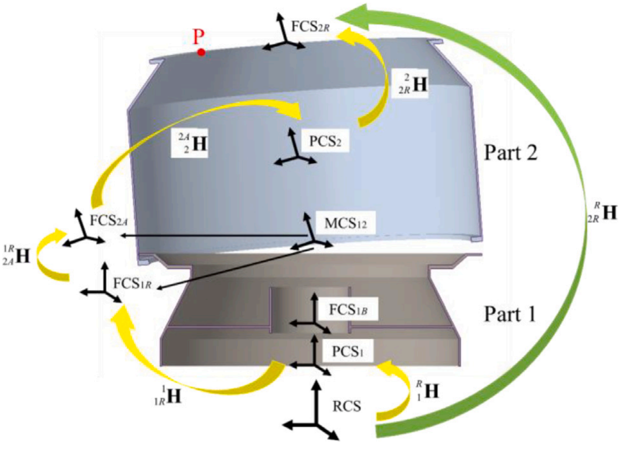


**Fig. 6.** Coordinate systems and HTM.

### 2.3. Virtual assembly based on the GDE model

Virtual assembly is the method of constraining the digital model of a product and simulating the actual assembly process. In this paper, virtual assembly refers specifically to assembly accuracy prediction modeling by unifying all aeroengine casing parts, geometrical features, and dense point clouds of GDEs in one assembly reference coordinate system (RCS) and calculating assembly accuracy.

To represent the assembly position and posture, the part coordinate system (PCS) is used to describe the aeroengine casing part, the feature coordinate system (FCS) is used to describe the surface, and the assembly motion coordinate system (MCS) is used to describe the assembly error caused by contact. As shown in Fig. 6, in RCS, the position and posture of Part 1 are described by PCS$_1$ the position and posture of the bearing mounting hole and flange surface on Part 1 are described by FCS$_{1B}$ and FCS$_{1R}$ and the assembly error caused by the contact of Part 1 and Part 2 is described by MCS$_{12}$.

$$\mathbf{H} = Trans(x,a)Trans(y,b)Trans(z,c)Rot(x,\theta x)Rot(y,\theta y)Rot(z,\theta z)$$
$$= \begin{bmatrix} cos\theta ycos\theta z & -cos\theta ysin\theta z & sin\theta y & a \\ sin\theta xsin\theta ycos\theta z + cos\theta xsin\theta z & -sin\theta xsin\theta ysin\theta z + cos\theta xcos\theta\theta z & -sin\theta xcos\theta y & b \\ -cos\theta xsin\theta ycos\theta z + sin\theta xsin\theta z & cos\theta xsin\theta ysin\theta z + sin\theta xcos\theta z & cos\theta xcos\theta y & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

**Box I.**

HTM is used to describe the spatial relationship between rigid bodies in robot kinematics [50]. In this paper, HTM is used to describe the change of position and posture in virtual assembly. The rigid body motion in virtual assembly includes parts motion, surface motion following parts motion, micro translation and rotation caused by contact, etc. For example, ${}^{1R}_{2A}\mathbf{H}$ represents the HTM from $\mathrm{FCS}_{1R}$ of surface 1R on Part 1 to $\mathrm{FCS}_{2A}$ of surface 2A on Part 2, as shown in Fig. 6. If a coordinate system first translates $a$, $b$, $c$ along $x$, $y$, $z$ axes, and then rotates $\theta x$, $\theta y$, $\theta z$ around $x$, $y$, and $z$ axes (see Box I), is represented as Eq. (6).

An error propagation model for virtual assembly is established shown in Eq. (7).

$${}^{R}_{2R}\mathbf{H} = {}^{R}_{1}\mathbf{H}{}^{1}_{1R}\mathbf{H}{}^{1R}_{2A}\mathbf{H}{}^{2A}_{2}\mathbf{H}{}^{2}_{2R}\mathbf{H} \quad (7)$$

Where, ${}^{R}_{2R}\mathbf{H}$ represents the position and posture of $\mathrm{FCS}_{2R}$ in RCS. ${}^{R}_{1}\mathbf{H}$, ${}^{1}_{1R}\mathbf{H}$, ${}^{2A}_{2}\mathbf{H}$, and ${}^{2}_{2R}\mathbf{H}$ can be obtained based on design or measured dimensions of aeroengine casings. And, ${}^{1R}_{2A}\mathbf{H}$ can be obtained based on the difference surface method.

As shown in Fig. 7, a difference surface point cloud and a zero point cloud are established based on the Surface 1R point cloud as a reference point cloud and the Surface 2A point cloud as an assembly point cloud. Then three contact points and ${}^{1R}_{2A}\mathbf{H}$ are calculated. At last, the contacted point cloud which represents the Surface 2A point cloud after assembly can be calculated based on ${}^{1R}_{2A}\mathbf{H}$ and the assembly point.

Considering the geometric structure and assembly process of aeroengine casings, there are 6 principles and constraints of contact point calculation:

(1) Rigid body assumption. No deformation of parts during virtual assembly.

(2) Non-interference principle. No interference between contact surfaces.

(3) Three contact points assumption. Based on (1) and (2), there are three contact points between two contact surfaces.

(4) The constraints degrees of freedom. The translational degrees of freedom of X and Y are 0 because aeroengine casings have rabbet or dowel pins positioning. The rotational freedom of Z is 0 because of the alignment of the through-hole of the upper flange and the threaded hole of the lower flange during assembly.

(5) Center of gravity stability principle. The triangle formed by three contact points projected on the reference plane must include the center of gravity of the part.

(6) Disturbance stability principle. Based on (5), the three line segments were obtained by connecting the three contact points with the center of gravity. When the minimum angle approaches 0°, it becomes unstable after receiving external force disturbance. When all three angles are 120°, it is most stable after being disturbed by external forces. The contact point calculation results must ensure that the minimum angle is greater than 15° in this paper.

As shown in Fig. 7, there is a point P on the Surface 2R. The known coordinate value of P in $\mathrm{FCS}_{2R}$ is ${}^{2R}_{P}v(x_{2R}, y_{2R}, z_{2R}, 1)$, and the unknown coordinate value of P in RCS is ${}^{R}_{P}v(x_R, y_R, z_R, 1)$. ${}^{R}_{P}v(x_R, y_R, z_R, 1)$ can be calculated according to Eqs. (7) and (8).

$${}^{R}_{P}v = {}^{R}_{2R}\mathbf{H}{}^{2R}_{P}v \quad (8)$$

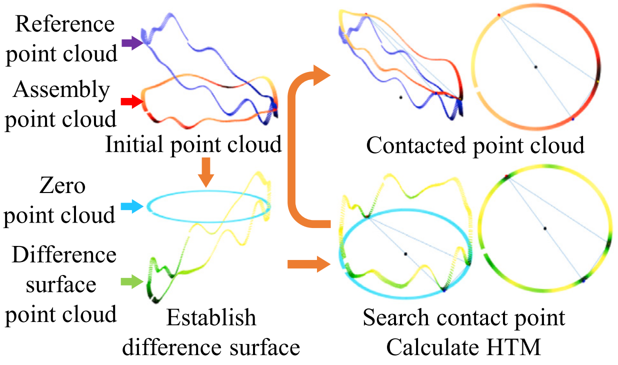


**Fig. 7.** Difference surface method.

By using the same method, the position and posture of any point, face, and part in RCS can be obtained. Any geometric accuracy such as coaxiality, runout, parallelism, perpendicularity, etc. can be obtained.

In summary, this section establishes the GDE model which maps between the measurement point cloud and the reconstructed dense point cloud by the NURBS method. The GDE model can accurately and comprehensively describe the GDE of aeroengine surfaces. This section also establishes the virtual assembly model which maps between the reconstructed dense point cloud and the coaxiality of the aeroengine casing system by HTM and difference surface method. The virtual assembly model is used to obtain the assembly dataset.

## 3. Coaxiality prediction based on point cloud deep learning

Based on part error models, the coaxiality of different workpiece assemblies can be calculated. However, it is still time-consuming and requires experienced engineers to construct virtual assembly models. To save modeling costs, point cloud deep learning is used to construct the relationship between object surface point cloud and coaxiality. The large amounts of point cloud data and the matching labels are fed to the point cloud neural network, and the end-to-end coaxiality calculation is realized. To accurately predict coaxiality, a new point cloud deep learning architecture Self-channel cross attention point network (SCCAPNet) is proposed. The details of this model are introduced in this section.

### 3.1. SCCAPNet overview

The overall architecture of SCCAPNet is illustrated in Fig. 8. The method proposed in this paper constructs a projection relationship between the assembled surface point cloud $P$ and the coaxiality $C$ of assembly results, where $P = \{p_i \in \mathbb{R}^3, i = 1, 2, \dots, N\}$ is a collection of three-dimensional arrays with $N$ elements. The $N$ presents the number of points are used for neural network training in a set of assembled surface point cloud and each element $p_i$ records the 3D coordinates $(x_i, y_i, z_i)$. Following the point cloud deep leaning diagram [39,45,48], the coordination feature size is increased from 3 to 32 by leveraging

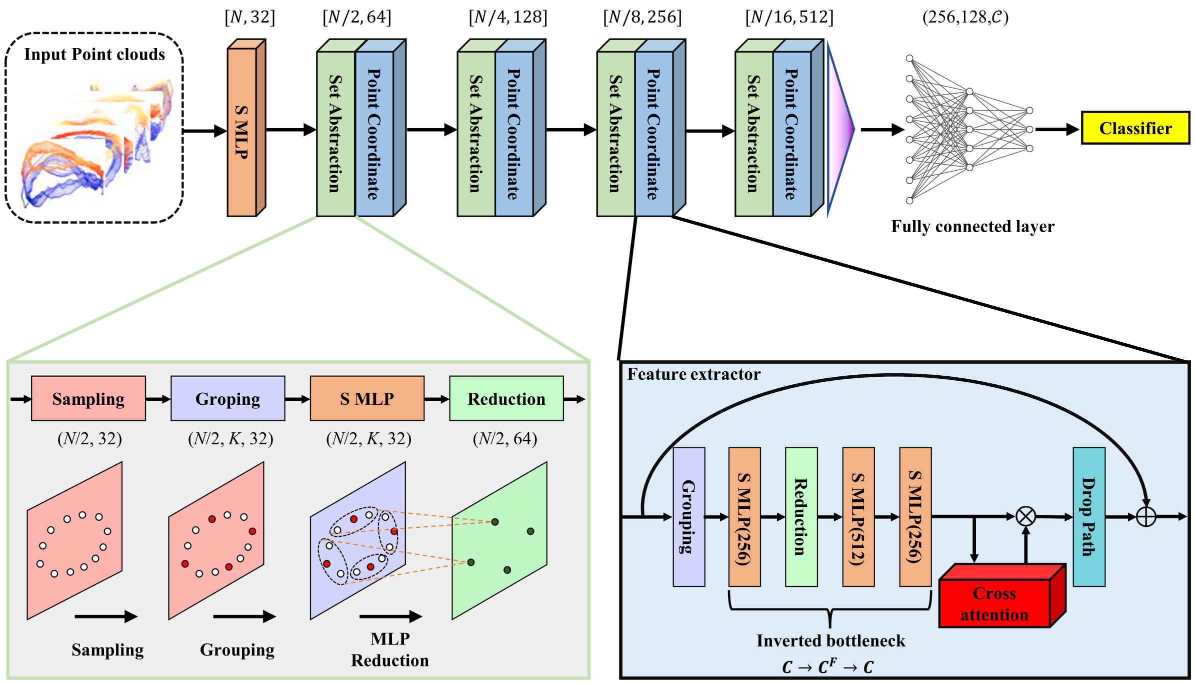


**Fig. 8.** SCCAPNet architecture.

a stem shared MLP (S MLP). After the receptive field expands, the feature map is rescaled four times, corresponding to the four stages. The number dimension $N$ is narrowed down to one in sixteen while the coordinate dimension is expanded sixteen times. For the feature extraction process, the Set Abstraction layer is designed to construct global relations between different points, and a point coordinate block is applied to learn sample features. To reinforce the feature interaction of different dimensions, a self-channel cross attention module is introduced and stacked to each point coordinator block at every training stage.

After all the feature extraction is finished, a symmetry layer is used to solve the point cloud unordered input. The SSCAPNet can get the same prediction results with different point input orders, which can reduce the dataset construction costs. The symmetry function is shown in Eq. (9).

$$e(\{p_1, p_2, \ldots, p_i\}) \approx s(SSCAP(p_1), SSCAP(p_2), \ldots, SSCAP(p_i)) \tag{9}$$

Where $e$ represents the ideal end-to-end training process, which transposes the input point cloud to the prediction label directly. *SSCAP* is the main backbone of the network architecture, and $s$ is the symmetry layer. The symmetry layer is constructed by a max pooling layer in this paper and compresses the feature map to a 512-long vector. To link with the classifier, the vector is fed to three fully connected (FC) layers defined as (256, 128, $\mathcal{C}$). The normalization layer and dropout layer are implemented between these layers to speed up the training process and prevent gradient vanishing. The classifier is equipped with a sigmoid function and cross-entropy as a loss function.

### 3.2. Set abstraction layer and point coordinate block

Empirically, the receptive field size and neural network performance present a strong positive correlation, which also is confirmed in the field of point cloud deep learning [51]. Based on this principle, the receptive field is doubled in the downsampling process via the Set Abstraction layer and point coordinate block.

The Set Abstraction layer is applied to construct the global relationship between neighbor point clouds. It contains three key processes: sampling, grouping, and feature map expansion. The sampling layer confirms several centroid points and reshapes the input features: $\mathbb{R}^{B*N*C^l} \rightarrow \mathbb{R}^{B*N'*C^l}, N' \in [1, N]$, where $C^l$ means the size of the coordinates feature in the $l_{th}$ set abstraction layer. To maintain the original shape and ensure better coverage of the entire point set [52], the Furthest Point sampling algorithm is introduced [45]. After downsampling, the grouping layer queries neighbors for each point $N'$. The grouping radius '$r$' is set to 0.8 and the number of points calculated in the radius '$K$' is set to half of the last channel in this task. The grouping layer adds a dimension to the input feature: $\mathbb{R}^{B*N*C^l} \rightarrow \mathbb{R}^{B*N'*K*C^l}$. In the end, a combination of shared MLP and reduction layer is used for feature map expansion. Each grouping point cloud set is constructed local frame relative to the sampling center point: $p_j^h = \hat{p}_j^h - p_j, h \in [1, K], j \in [1, N']$, where $p_j$ is the centroid point determined by the sampling layer and $\hat{p}_j^h$ is the original coordinate of grouping results. The shared MLP doubles the receptive field while the reduction layer aggregates the neighbor points relationship and guarantees the feature shape translating to three dimensions.

Similar to the Set Abstraction layer, the stem of the point coordinate block is an alignment of grouping, shared MLP, and reduction layer. Different from the former, the centroid point was not downsampled, due to the finite number of training points, especially in the fourth stage. Thus, the stem normalization process can be written as Eq. (10):

$$\mathcal{F}_i^{l+1} = \mathcal{R}_{(j,h):j\in N',h\in K}\left\{ s\left[\mathcal{F}_j^l; \hat{p}_j^{hl} - d(p_i^l)\right]\right\} \tag{10}$$

Where $p_i^l$ is the input point clouds used for $l_{th}$ layer. Function $d$ is the downsampling process of the sampling layer and $s$ is the shared MLP layer, which are illustrated in Section 3.1. $\hat{p}_j^{hl} - d(p_i^l)$ is relevant to the local frame construction and $\mathcal{F}_j^l$ is the output of the set abstraction layer. $\mathcal{R}$ is the reduction layer and reinforces the connection between centroid point $d(p_i^l)$ and its neighbors $\hat{p}_j^{hl}$, defined as $(j,h) : j \in N', h \in K$. Inspired by the ConvNext [53], the SOTA method of computer vision, the inverted bottleneck [54,55] is applied to shared MLP in each stage for coordinate-wise information interaction and effective model scaling. In the inverted bottleneck, the coordinates dimension is expanded and compressed for $F$ times ($C \rightarrow C^F \rightarrow C$) via shared MLP layers. $F$ is set to two in this work, and add a residual connection to prevent vanishing gradient caused by the network depth increasing.

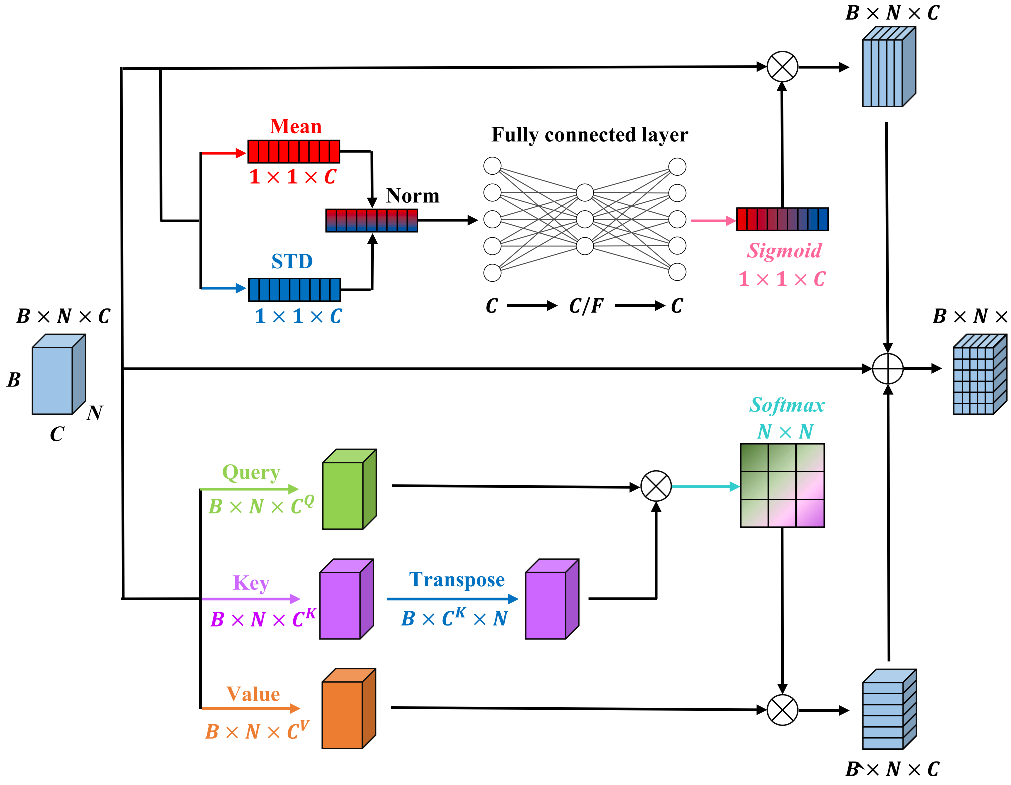


Fig. 9. Self-channel cross attention module.

To better leverage the long-range context dependencies among points, a self-channel cross attention module is introduced after an inverted bottleneck. After attention mechanisms, the feature map is fed to the drop path layer to prevent network overfitting. The dropped features are fused with the skip link and transported to the next stage for another sampling.

### 3.3. Self-channel cross attention module

The connection between neighbor point clouds sampled in radius $r$ is effectively built based on the grouping layer. However, the sampling method loses spatial dependencies among global points, containing rich geometrical information. The attention mechanism reinforces the network global information-capturing ability via expanding broader dependencies of input features, which improves training performance. Additionally, the attention mechanism is based on MLP achieved matrix multiplication, fitting the points cloud solving requirement of transformation invariance. To better correlate between different point clouds feature dimensions, this paper hybridizes disparate dimension attention and proposes a new architecture, the self-channel cross attention module.

The overview of the cross attention module is shown in Fig. 9. As for an input point cloud feature ($\mathbb{R}^{B\times N\times C}$), traditional attention mechanisms [56,57] only focus on one dimension. Different previous work, cross attention is a hybrid module mixing feature map from two feature dimensions. As for the coordinates dimension, the feature map is fed to the upper branch. In order to make full use of the global spatial information, both the mean and standard deviation (STD) of the input feature are leveraged at the same time. The two-norm is used to mix the mean feature and STD feature to make sure different features are preserved and fused. The process is shown in Eq. (11).

$$\begin{cases} \mu_d = \frac{1}{N\times B}\sum_{n=1}^{N}\sum_{b=1}^{B} f_{n\times b}^{d} \\ \sigma_d = \sqrt{\frac{1}{N\times B}\sum_{n=1}^{N}\sum_{b=1}^{B}(f_{n\times b}^{d}-\mu^{d})^2} \\ Norm_d = \sqrt{\mu_d^2+\sigma_d^2} \end{cases} \tag{11}$$

Where $f_{n\times b}^{d}$ is the $d_{th}$ feature map with shape $N\times B$ and $\mu_d, d \in [1, C]$ is the mean value of $f_{n\times b}^{d}$ realized by average pooling. $\sigma_d$ represents the STD of the same feature map. To combine these two, the two-norm ($Norm_d$) is introduced as Eq. (11). The feature size of $Norm_d$ is the same as the mean and STD value, equal to $\mathbb{R}^{1\times 1\times C}$ and is transported to a combination of two fully connected layers. In contrast to the design of an inverted bottleneck, the fully connected layers compress the vector $F$ times (from $C$ to $C/F$) and expand it to the original size. $F$ is set to two to correspond to the downsampling ratio in the main stage. Before the shortcut connection, the feature map is passed through a batch norm layer and a sigmoid function. In the upper branch, a feature map reinforced information interaction in $N \times B$ dimension is obtained.

In the lower branch, the feature map is separated into three paths. And the query, key, and value are calculated by feature transformations. The calculation process is denoted as Eq. (12).

$$\mathcal{F}^{l+1}_{Q,K,V} = \mathcal{F}^{l} \times W^{l}_{Q,K,V} \tag{12}$$

The $\mathcal{F}^{l}$ is the feature map in the $l_{th}$ layer and $W^{l}_{Q,K,V}$ is the correlated weight of three paths, where $W^{l}_{Q} \in \mathbb{R}^{C\times C_Q}$, $W^{l}_{K} \in \mathbb{R}^{C\times C_K}$, and $W^{l}_{V} \in \mathbb{R}^{C\times C_V}$. To reduce computation cost and balance the downsampling rate with the upper branch, $C_Q = C_K = C/F$ and $C_V = C$ are defined in this work. The equation of the attention result $\mathcal{F}^{l+1}_{low}$ in the lower branch is defined as Eq. (13).

$$\mathcal{F}^{l+1}_{low} = \alpha\left[\mathcal{F}^{l+1}_{Q} \times (\mathcal{F}^{l+1}_{K})^{T} / \sqrt{C/F}\right] \times \mathcal{F}^{l+1}_{V} \tag{13}$$

In the beginning, query $\mathcal{F}^{l+1}_{Q}$ performs matrix multiplication with the transpose of key $\mathcal{F}^{l+1}_{K}$. The result is normalized by the square root of the coordinates dimension $C/F$. Then the feature map is dealt with an activation function $\alpha$, and the sigmoid function is chosen. Lastly, the outputs multiply with the value $\mathcal{F}^{l+1}_{V}$, and the attention feature map of $N \times B$ dimension is obtained. After the dual attention architecture, the self-channel cross attention module combines two previous attention features with the input feature map via a shortcut. It finally mixes the information from different dimensions and reinforces the global information interaction.

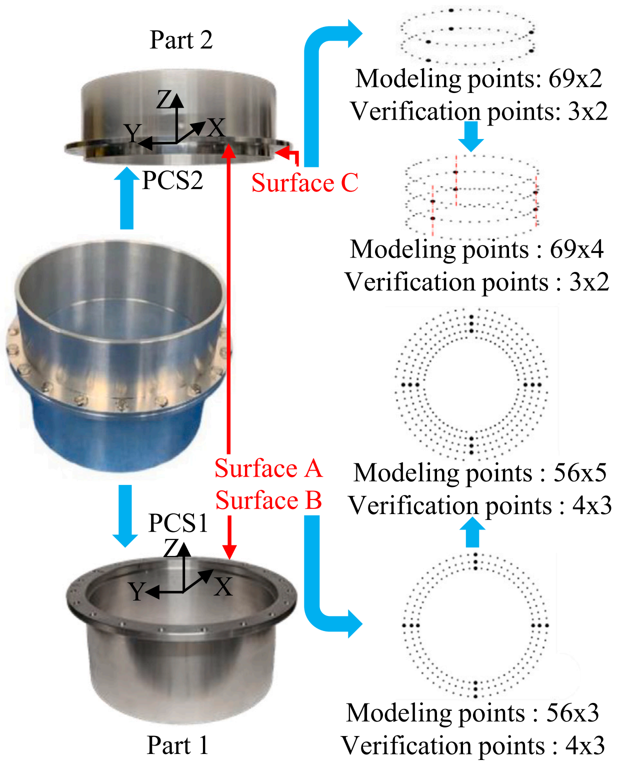


**Fig. 10.** Aeroengine simulated parts and data processing.

## 4. Experiment and analysis

In this section, a mapping dataset between aeroengine simulated workpiece point clouds and coaxiality is established through GDE modeling and the virtual assembly. The SCCAP net is tested in the dataset and compared with other SOTA deep learning methods to demonstrate the effectiveness of the approach proposed in this paper.

### *4.1. Data description*

14 aeroengine simulated parts are designed and processed, as shown in Fig. 10. Point clouds are measured with the Hexagon Leitz PMM-XI12107 CMM. The measurement uncertainty of CMM is ±(0.5+L/500) μm.

### *4.2. GDE modeling and accuracy verification*

The accuracy verification of the NURBS model based on the measurement point cloud is implemented in this section. Data processing for accuracy verification is shown in Fig. 10. The measurement point cloud is divided into two groups. Small points as modeling points are edge extended and interpolated into NURBS surfaces. Large points as verification points are not interpolated into NURBS surfaces. The model accuracy is represented as the average distance between verification points and the NURBS surface.

Dense point clouds are obtained by the NURBS method based on modeling point clouds, as shown in Fig. 11. GDE of point clouds is amplified and displayed.

Single-point distances between verification points and NURBS surfaces are calculated. The surface accuracy is represented as the average of single-point distances, as shown in Table 1, Table 2, and Table 3. The surface accuracy is 0.0009 mm, 0.0008 mm, and 0.0010 mm. The result verifies the GDE model based on the NURBS surface can accurately predict GDE.

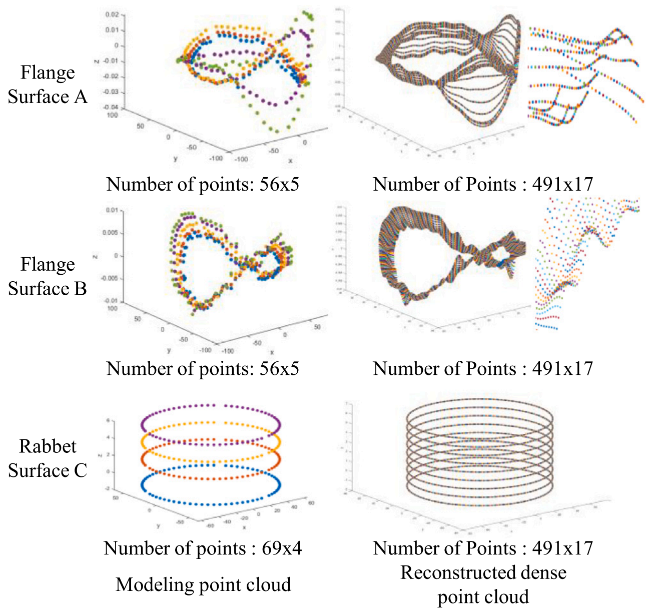


**Fig. 11.** GDE modeling and reconstruction.

**Table 1**
Accuracy of NURBS surface A (mm).

| X | Y | Z | Single point distance |
|---|---|---|---|
| 3.4544 | 65.9087 | 0.0003 | 0.0014 |
| −65.9094 | 3.4556 | −0.0089 | 0.0001 |
| −3.4540 | −65.9088 | 0.0088 | 0.0002 |
| 65.9093 | −3.4553 | −0.0171 | 0.0000 |
| 3.6639 | 69.9039 | 0.0060 | 0.0004 |
| −69.9039 | 3.6636 | −0.0058 | 0.0017 |
| −3.6634 | −69.9037 | 0.0142 | 0.0021 |
| 69.9041 | −3.6638 | −0.0126 | 0.0018 |
| 3.8731 | 73.8983 | −0.0127 | 0.0000 |
| −73.8986 | 3.8729 | −0.0003 | 0.0000 |
| −3.8727 | −73.8986 | −0.0237 | 0.0003 |
| 73.8987 | −3.8729 | 0.0040 | 0.0026 |
| Flatness | 0.0379 | Average | 0.0009 |

**Table 2**
Accuracy of NURBS Surface B (mm).

| X | Y | Z | Single point distance |
|---|---|---|---|
| 2.8790 | 65.9371 | 0.0034 | 0.0000 |
| −65.9370 | 2.8789 | −0.0053 | 0.0002 |
| −2.8787 | −65.9373 | 0.0051 | 0.0006 |
| 65.9374 | −2.8792 | −0.0049 | 0.0010 |
| 3.0534 | 69.9333 | 0.0050 | 0.0000 |
| −69.9333 | 3.0532 | −0.0048 | 0.0007 |
| −3.0532 | −69.9337 | 0.0058 | 0.0005 |
| 69.9333 | −3.0536 | −0.0051 | 0.0005 |
| 3.2279 | 73.9294 | 0.0057 | 0.0002 |
| −73.9295 | 3.2278 | −0.0037 | 0.0013 |
| −3.2277 | −73.9297 | 0.0068 | 0.0011 |
| 73.9295 | −3.2282 | −0.0045 | 0.0034 |
| Flatness | 0.0150 | Average | 0.0008 |

### *4.3. Virtual assembly and data augmentation*

The dataset of dense point clouds is augmented for the deep learning model. The methods for data augmentation include weight combination, rotation, function addition, error scaling, mirroring, noise signal addition, filtering, etc.

14 flange surface dense point clouds are augmented into 1600 point clouds. As shown in Fig. 12, dense point clouds of Surface A and Surface Band are augmented into 9 sets of point clouds.

**Table 3**
Accuracy of NURBS Surface C (mm).

| X | Y | Z | Single point distance |
|---|---|---|---|
| 45.2393 | 45.2371 | 1.9991 | 0.0009 |
| −61.8124 | 16.5631 | 2.0001 | 0.0010 |
| 16.5525 | −61.7763 | 1.9996 | 0.0013 |
| 45.2392 | 45.2381 | 4.0008 | 0.0015 |
| −61.8123 | 16.5629 | 4.0002 | 0.0003 |
| 16.5524 | −61.7769 | 4.0003 | 0.0012 |
| Cylindricity | 0.0373 | Average | 0.0010 |

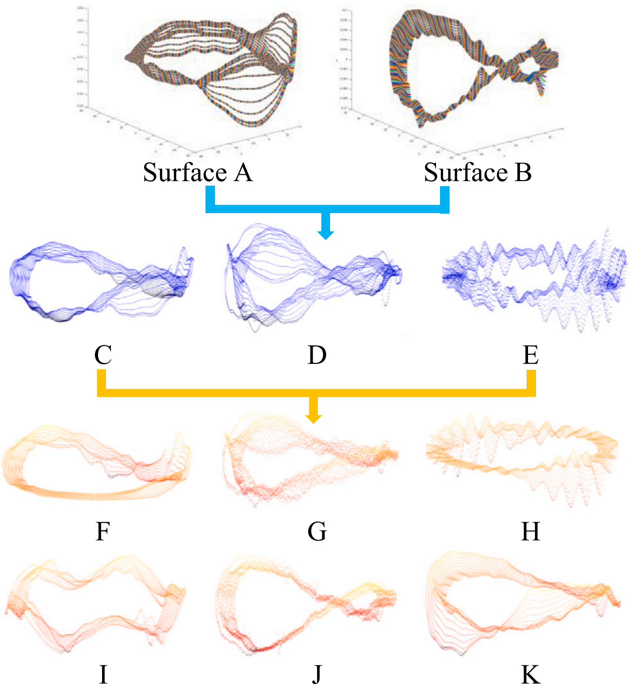


**Fig. 12.** Point clouds extension.

An example of virtual assembly is implemented in this section. The GDE model and error propagation of aeroengine simulated parts is shown in Fig. 13, and the GDE of CAD solid models is amplified and displayed. The coaxiality D$_{12}$ of Part 2, which is based on the Part 1 position reference, is defined as twice the projection distance on the XOY plane in RCS between the origin of FCS$_{1C}$ and the origin of FCS$_{2C}$. To obtain ${}^{R}_{1C}\mathbf{H}$ of FCS$_{1C}$. and ${}^{R}_{2C}\mathbf{H}$ of FCS$_{2C}$, Part 1 is selected as the assembly reference part, and the error propagation equation is defined as Eqs. (14) and (15).

$$ {}^{R}_{1C}\mathbf{H} = {}^{R}_{1}\mathbf{H}\,{}^{1}_{1C}\mathbf{H} \tag{14} $$

$$ {}^{R}_{2C}\mathbf{H} = {}^{R}_{1}\mathbf{H}\,{}^{1}_{1R}\mathbf{H}\,{}^{1R}_{2A}\mathbf{H}\,{}^{2A}_{2}\mathbf{H}\,{}^{2}_{2C}\mathbf{H} \tag{15} $$

The HTM between overlapping coordinate systems is the identity matrix **I**. Therefore, ${}^{R}_{1}\mathbf{H} = {}^{1}_{1R}\mathbf{H} = {}^{2A}_{2}\mathbf{H} = \mathbf{I}$. As shown in Fig. 13, the design dimension contains $h_1$ =60 mm and $h_2$ =40 mm, so ${}^{1}_{1C}\mathbf{H} = Trans(z,-60)$ and ${}^{2}_{2C}\mathbf{H} = Trans(z,40)$. ${}^{1R}_{2A}\mathbf{H}$ is obtained by the difference surface method in Section 2.2. Eqs. (14) and (15) can be rewritten as Eqs. (16) and (17).

$$ {}^{R}_{1C}\mathbf{H} = {}^{R}_{1}\mathbf{H}\,{}^{1}_{1C}\mathbf{H} = Trans(z,-60) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -60 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16} $$

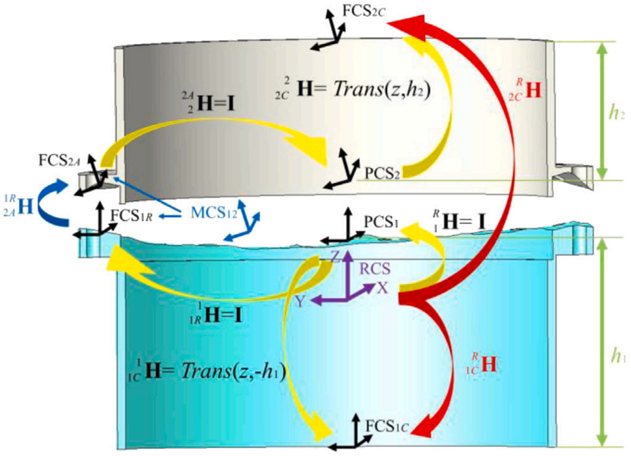


**Fig. 13.** Error propagation in virtual assembly.

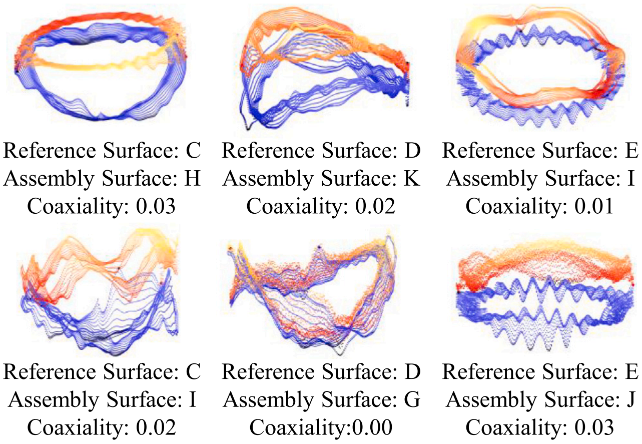


**Fig. 14.** Virtual assembly examples.

$$ \begin{aligned} {}^{R}_{2C}\mathbf{H} &= {}^{R}_{1}\mathbf{H}\,{}^{1}_{1R}\mathbf{H}\,{}^{1R}_{2A}\mathbf{H}\,{}^{2A}_{2}\mathbf{H}\,{}^{2}_{2C}\mathbf{H} \\ &= \begin{bmatrix} 1 & 0 & 0.0002 & 0 \\ 0 & 1 & 0.0001 & 0 \\ -0.0002 & -0.0001 & 1 & 0.0416 \\ 0 & 0 & 0 & 1 \end{bmatrix} Trans(z,40) \\ &= \begin{bmatrix} 1 & 0 & 0.0002 & 0.0080 \\ 0 & 1 & 0.0001 & 0.0040 \\ -0.0002 & -0.0001 & 1 & 40.0416 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{17} $$

The origin of FCS$_{1C}$ is (0, 0, −60) in RCS, and the origin of FCS$_{2C}$ is (0.008, 0.004, −40.0416) in RCS. The coaxiality D$_{12}$=0.0179 mm.

6722 virtual assembly examples are calculated based on 1600 dense point clouds. The mapping dataset between dense point clouds and coaxiality is obtained through the virtual assembly. Virtual assembly examples are shown in Fig. 14. The statistics of the dataset are shown in Table 4.

An assembly point could dataset is constructed based on the modeling process. The dataset contains coaxiality values with 0.01 mm accuracy and is labeled in ten tiers (T1 to T10). 6722 examples are separated into the training set, validation set, and test set based on the ratio 7:1:2. Additionally, Different tiers are divided into four gradings, according to assembly quality. The details of this dataset are illustrated in Table 4.

**Table 4**
Details of point cloud dataset.

| Tier | Coaxiality value (mm) | Grade | Number of examples | Train | Val | Test |
|---|---|---|---|---|---|---|
| T0 | 0 | Premium Grade | 225 | 158 | 23 | 45 |
| T1 | 0.01 | | 774 | 542 | 77 | 155 |
| T2 | 0.02 | | 928 | 650 | 93 | 186 |
| T3 | 0.03 | Qualified Grade | 713 | 499 | 71 | 143 |
| T4 | 0.04 | | 608 | 426 | 61 | 122 |
| T5 | 0.05 | | 584 | 409 | 58 | 117 |
| T6 | 0.06 | Reprocessed Grade | 543 | 380 | 54 | 109 |
| T7 | 0.07 | | 408 | 286 | 41 | 82 |
| T8 | 0.08 | | 532 | 372 | 53 | 106 |
| T9 | 0.09 | Substandard Grade | 487 | 341 | 49 | 97 |
| T10 | 0.1 | | 415 | 291 | 42 | 83 |
| T11 | >0.1 | | 505 | 354 | 51 | 101 |

**Table 5**
Training hyperparameter.

| Config | Value |
|---|---|
| Batch size | 32 |
| Epoch | 500 |
| Drop path rate | 0.2 |
| Learning rate | 0.001 |
| Optimizer | Adam, betas = (0.9,0.999) |
| StepLR | Step size = 20, gamma = 0.7 |
| Decay rate | 0.0001 |
| n_points | 1024 |
| Sampling | Furthest point sampling |

### 4.4. Details of deep learning model implementation and evaluation metric

The feature interpreter and regression part of the proposed model is programmed in Pytorch (1.9.0) with Python (3.8.11). The model is trained on a workstation with a CPU of Intel Xeon Platinum 8375C @2.90 GHz and an NVIDIA GeForce RTX 3090 GPU with 24 GB memory using the PyCharm. The batch size of 32 is implemented for the hyperparameter setting. Adam optimizer is set to 0.001 learning rate and 0.9$\beta_1$, 0.999$\beta_2$. To balance the initial convergence speed and the final convergence speed, the step learning rate and decay rate are introduced in the training strategy. As for the sampling process, we follow the training setting of the SOTA method PointNet++ and choose 1024 as the sampling number in one assembly surface pair. To get better coverage results with limited training points, the furthest point sampling algorithm is deployed to seek neighbor points relationship. The training process lasts 500 epochs with a 0.2 drop path rate, and the specific values of the hyperparameters are shown in Table 5.

### 4.5. Deep learning prediction result

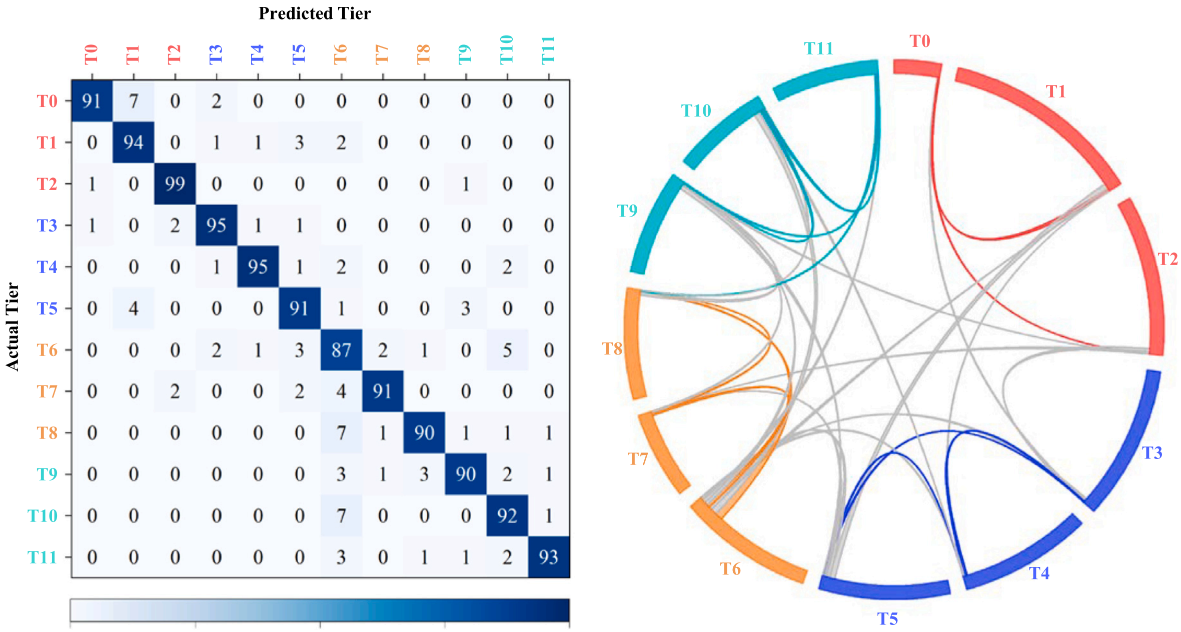
In this paper, accuracy (Acc), recall (Rec), precision (Pre), and F1-score (F1) are used to evaluate the SCCAPNet performance. To illustrate the effectiveness of the method proposed in this paper, other point cloud deep learning methods are also tested on the same dataset based on the evaluation metric. The point cloud deep learning diagram, the PointNet, is chosen as the experiment baseline. The method proposed in this paper is also compared with the MLP neural network representative PointNet++ and the graph neural network representative DGCNN. Meanwhile, PointNext, the SOTA network in Modelnet 40, is trained in the same dataset. The prediction result of these four methods and SCCAPNet is shown in Table 6. Due to the limited parameters, PointNet works poorest in our dataset, with an accuracy is less than 90%. The performance of PointNext is the best in the MLP series and reaches the top accuracy in all other methods, which is 92.06%. The method proposed in this paper, SCCAPNet, is 1.11% higher than PointNext (93.17% compared to 92.06%) and reaches the top one accuracy. Meanwhile, it is also the best in recall, precision, and F1-score. Especially, it performs well in the F1-score, 1.79% higher than the second one. The results illustrate the effectiveness and the low misdiagnosis rate of SCCAPNet, the key indicators of high precision assembly.

**Table 6**
Test result comparison with other methods.

| Model | Acc | Recall | Precision | F1 |
|---|---|---|---|---|
| PointNet | 0.8823 | 0.8915 | 0.8985 | 0.8843 |
| PointNet++ | 0.9136 | 0.9028 | 0.9051 | 0.9179 |
| DGCNN | 0.9147 | 0.9188 | 0.9036 | 0.9111 |
| PointNext | 0.9206 | 0.9175 | 0.9231 | 0.9153 |
| **Ours** | **0.9317** | **0.9228** | **0.9335** | **0.9332** |

To better illustrate the details of the method proposed in this paper, multi-dimensional data visualizations are performed as shown in Fig. 15. The confusion matrix (Fig. 15(a)) demonstrates the details of the prediction situation of each tier. Tier two gets the best prediction result, reaching 99% accuracy and there are 11 categories with more than 90% accuracy. Tier seven, with the lowest prediction rate, still retains 87% accuracy. In the real industrial situation, the misjudgments in the same grade (as shown in Table 4) are acceptable and the actual accuracy rate will be higher. The chord diagram (Fig. 15(b)) shows the misjudgment situation between different grades. The misjudgment of the same grades is linked with the color line and the gray line means the prediction result is in another grade. According to the chord diagram, it can be found that there is no misjudgment between premium grade and substandard grade, guaranteeing the method proposed in this paper will not scrap high-quality parts. It is obvious that the link of premium grade is typically less than the link of reprocessed grade, which illustrates that SCCAPNet is sensitive to high-quality parts and good at screening samples that do not require rework. If the same grade of misjudgment is regarded as accurate, the accuracy of the method proposed in this paper can improve by 1.89% and reach 95.06%.

The average precision (AP) and area under the curve (AUC) of SCCAPNet are compared to the other four methods to declare the prediction confidence, as shown in Fig. 16. As for the AP, the SCCAPNet gets the highest value 0.9356, 0.0877 higher than the last one and it also has the lowest standard deviation 0.0411. The PointNext reaches the second position, equal to 0.9126±0.0763. Compared to the second one, SCCAPNet is 0.023 higher in the AP value and 0.0352 lower in the standard deviation, while PointNet is the worst performance, with only 0.8479 AP value and the highest standard deviation of 0.1188, corresponding to the test results shown in Table 6. At the same time, the performance of all the methods on the AUC is around 0.4 higher than that of AP, and the standard deviation has shrunk to about one-half of the previous. The overall trend of the performance of the AUC evaluation is basically the same as that of the AP, except that the performance of DGCNN is better in AUC and it reaches the top during the other four methods. The PointNet is still the last one with a 0.8843 AUC value and 0.0341 standard deviation. The SCCAPNet is 0.0181 higher than the second one (0.9852 compared to 0.9671) in the AUC value, and the standard deviation is 0.0122 lower (0.0180 compared



|  | T0 | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | T11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T0 | 91 | 7 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T1 | 0 | 94 | 0 | 1 | 1 | 3 | 2 | 0 | 0 | 0 | 0 | 0 |
| T2 | 1 | 0 | 99 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| T3 | 1 | 0 | 2 | 95 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| T4 | 0 | 0 | 0 | 1 | 95 | 1 | 2 | 0 | 0 | 0 | 2 | 0 |
| T5 | 0 | 4 | 0 | 0 | 0 | 91 | 1 | 0 | 0 | 3 | 0 | 0 |
| T6 | 0 | 0 | 0 | 2 | 1 | 3 | 87 | 2 | 1 | 0 | 5 | 0 |
| T7 | 0 | 0 | 2 | 0 | 0 | 2 | 4 | 91 | 0 | 0 | 0 | 0 |
| T8 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 1 | 90 | 1 | 1 | 1 |
| T9 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 3 | 90 | 2 | 1 |
| T10 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 92 | 1 |
| T11 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 1 | 1 | 2 | 93 |

(a) Confusion matrix

(b) Chord diagram

**Fig. 15.** Results visualization.

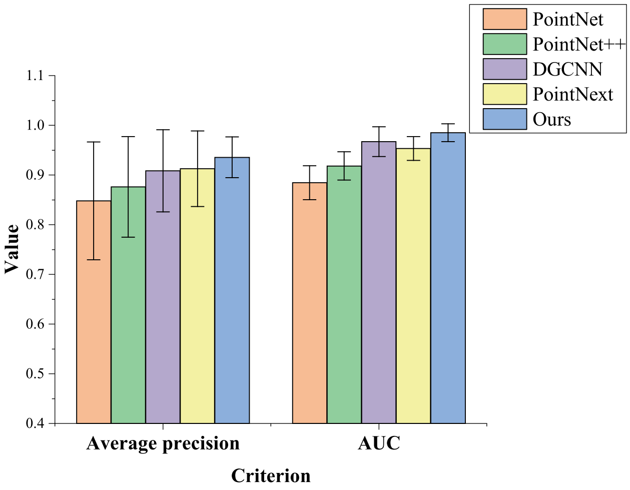


**Fig. 16.** AP and AUC results comparison.

to 0.0301). The experiment results show that SCCAPNet performs best both in the AP and AUC evaluation, which illustrates that it is not only accurate but also has high confidence and stability. SCCAPNet has highly robust performance and better adapts to complex situations in actual production.

To further illustrate the effectiveness of SCCAPNet, The point cloud data is visualized, and the t-distributed stochastic neighbor embedding (t-SNE) is plotted, as shown in Fig. 17. The original figure presents the input data distribution in two-dimensional space without neural network processing. The other five figures correspond to the prediction results of different methods. Each point in the figure presents a test example and the color means the category it belongs to. In the t-SNE image, the closer the distance between the samples, the closer the features extracted by the deep neural network. When the points of the same color are densely distributed and the points of different colors are discrete, it means that the feature extraction capability of the network is strong. The PointNet can clearly distinguish T7, T8, and T9, but it cannot distinguish other kinds. Although the results of PointNet++, DGCNN, and PointNext show a trend of separation of different types of data, there are still many kinds of confusion in the central area. It is obvious that the SCCAPNet is capable of completely isolating the features of seven species and there is only a small amount of confusion between T6, T9, and T10. The experiment result shows that most of the twelve categories have clearly separable boundaries and demonstrate the effectiveness of our backbone. This is mainly due to the multiple information extraction and fusion of the self-channel cross attention module.

### 4.6. Discussion

The method proposed in this paper is tested on aeroengine simulated parts. As the experiment results show, based on sparse point clouds measured by CMM, the geometric distribution error model can accurately reconstruct dense point clouds of contact surfaces and effectively control the error within 0.001 mm. The virtual assembly method successfully constructs the relationship between the dense point cloud and coaxiality and establishes the dataset including the assembly point clouds and corresponding coaxiality. The new deep learning architecture SCCAPNet is tested on the dataset and realizes SOTA performance compared to other baselines. The experiment result shows that the method proposed in this paper can achieve 0.01 mm precision coaxiality prediction with 93.32% accuracy. Meanwhile, the prediction process is automatic and does not require any additional modeling and parameter tuning. The method meets the needs of industrial production and contributes to the accuracy assurance of precision assembly and the reduction of assembly costs.

## 5. Conclusions

In this paper, a coaxiality prediction method for aeroengines precision assembly by combining the geometric distribution error model and point cloud deep learning is proposed. A geometric distribution error model is established to reconstruct dense point clouds to accurately and

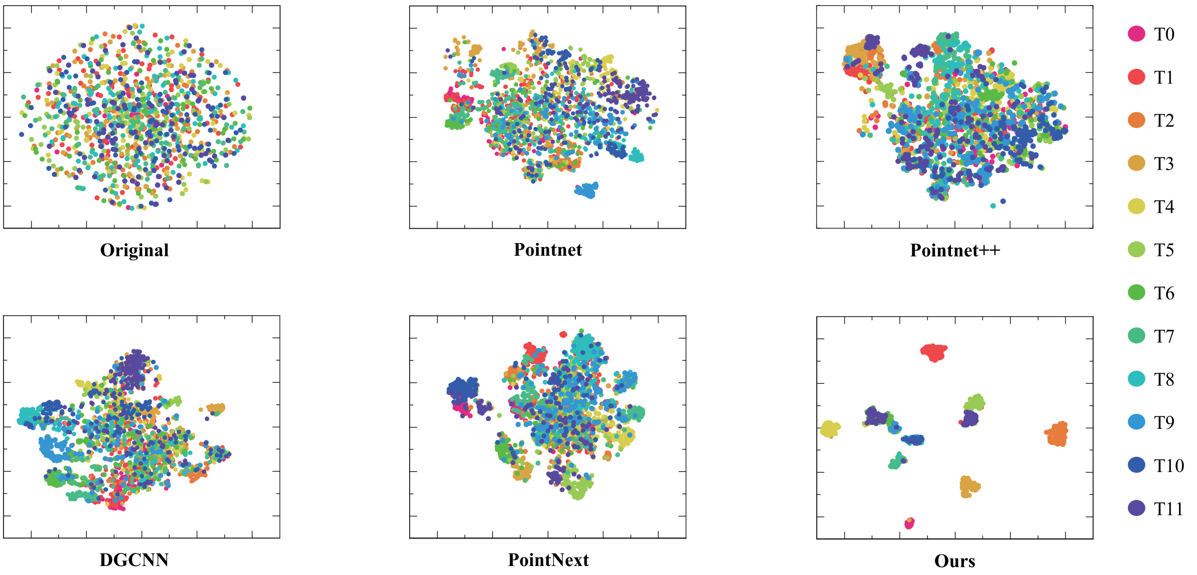


**Fig. 17.** Features visualization based on t-SNE.

comprehensively describe the machining error of aeroengine surfaces. A virtual assembly model is established to map between dense point clouds and coaxiality. A new point cloud deep learning backbone, SCCAPNet, is established to learn this mapping relationship and realize the end-to-end prediction between point clouds and coaxiality. The main contributions of this study can be summarized as follows:

(1) This is the first attempt to introduce deep learning in the area of aeroengine coaxiality calculation with 3D geometric distribution error point cloud input. The deep learning method realizes end-to-end coaxiality prediction, which saves the cost of virtual assembly model construction and parameter adjustment for specific measured objects. It also improves the generalization of the assembly accuracy prediction model.

(2) A geometric distribution error model is derived to realize the accurate surface construction by the NURBS method. The impact of micro geometric distribution errors on assembly accuracy prediction is comprehensively considered in this paper.

(3) A new point cloud backbone, the SCCAPNet, is proposed in this paper for effective point cloud feature learning. To further improve the network performance, this paper introduces a new attention module, the self-channel cross attention module, to reinforce the information interaction between different feature dimensions. This backbone is also compared with other deep learning baselines and reaches the top one accuracy in the point cloud based coaxiality prediction tasks.

(4) The precision assembly coaxiality prediction method is tested on simulated aeroengine flange components. The experiment results verified the method can realize 0.001 mm accuracy geometric distribution error and 0.01 mm precision coaxiality prediction, which can be used in real industrial situations and meets the requirement of aeroengine assembly.

The precision assembly coaxiality prediction method solved the problem of independent modeling for different contact surfaces in the coaxiality prediction. With trained neural networks, coaxiality can be obtained directly by inputting geometric distribution error point cloud information of contact surfaces, instead of modeling for this single target. The experiment results verified the effectiveness of the method. This method solved the problem of coaxiality prediction for aeroengine casing system assembly, so as to improve the assembly quality and efficiency of the aeroengine. In the future, the variety and complexity of the tested objects will be increased to improve the generalization of the model.

## 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.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (project No. U22B2088), the Research Grant Council (RGC) of Hong Kong under Grant 11217922, 11212321 and Grant ECS-21212720, and the Science and Technology Innovation Committee of Shenzhen under Grant Type-C SGDX20210823104001011.

## References

[1] Klinger Holger, Lazik Waldemar, Wunderlich Thomas. The engine 3E core engine. In: Turbo expo: Power for land, sea, and air, vol. 43116. 2008, p. 93–102.
[2] Zhu C, Chen Y. Vibration characteristics of aeroengine's rotor system during maneuvering flight. Acta Aeronaut Astron Sinica-Series A B- 2006;27(5):835.
[3] Di Pietro Enrico, Barabaschi Pietro, Kamada Yutaka, Ishida Shinichi, et al. Overview of engineering design, manufacturing and assembly of JT-60sA machine. Fusion Eng Des 2014;89(9–10):2128–35.
[4] Andolfatto Loïc, Thiébaut François, Lartigue Claire, Douilly Marc. Quality-and cost-driven assembly technique selection and geometrical tolerance allocation for mechanical structure assembly. J Manuf Syst 2014;33(1):103–15.
[5] Saren Qimuge, Zhang Zhijing, Xiong Jian, Chen Xiao, Zhu Dongsheng, Wu Wenrong, et al. An accuracy and performance-oriented accurate digital twin modeling method for precision microstructures. J Intell Manuf 2023;1–25.
[6] Zuo Fuchang, Zhang Zhijing, Jin X. Research on contact state of mating surfaces taking into account plane form error. Binggong Zidonghua (Ordnance Ind Automat) 2012;31(9):32–7.
[7] Mu Xiaokai, Wang Yunlong, Yuan Bo, Sun Wei, Liu Chong, Sun Qingchao. A new assembly precision prediction method of aeroengine high-pressure rotor system considering manufacturing error and deformation of parts. J Manuf Syst 2021;61:112–24.
[8] Huang De-Lin, Du Shi-Chang, Li Gui-Long, Wu Zhuo-Qi. A systematic approach for online minimizing volume difference of multiple chambers in machining processes based on high-definition metrology. J Manuf Sci Eng 2017;139(8):081003.
[9] Schleich Benjamin, Anwer Nabil, Mathieu Luc, Wartzack Sandro. Skin model shapes: A new paradigm shift for geometric variations modelling in mechanical engineering. Comput Aided Des 2014;50:1–15.
[10] Sun Qingchao, Zhao Binbin, Liu Xin, Mu Xiaokai, Zhang Yuanliang. Assembling deviation estimation based on the real mating status of assembly. Comput Aided Des 2019;115:244–55.
[11] Li Guilong, Du Shichang, Wang Bo, Lv Jun, Deng Yafei. High definition metrology-based quality improvement of surface texture in face milling of workpieces with discontinuous surfaces. J Manuf Sci Eng 2022;144(3):031001.

[12] Nguyen Hai Trong, Wang Hui, Hu S Jack. Characterization of cutting force induced surface shape variation in face milling using high-definition metrology. J Manuf Sci Eng 2013;135(4):041014.

[13] Shao Yiping, Wang Kun, Du Shichang, Xi Lifeng. High definition metrology enabled three dimensional discontinuous surface filtering by extended tetrolet transform. J Manuf Syst 2018;49:75–92.

[14] Karaçalı Bilge, Snyder Wesley. Reconstructing discontinuous surfaces from a given gradient field using partial integrability. Comput Vis Image Underst 2003;92(1):78–111.

[15] Li Guilong, Du Shichang, Huang Delin, Zhao Chen, Deng Yafei. Dynamics modeling-based optimization of process parameters in face milling of workpieces with discontinuous surfaces. J Manuf Sci Eng 2019;141(10):101009.

[16] Ghaly Wahid S, Mengistu Temesgen T. Optimal geometric representation of turbomachinery cascades using NURBS. Inverse Probl Sci Eng 2003;11(5):359–73.

[17] Zhu Lida, Yan Boling, Wang Yulian, Dun Yichao, Ma Jian, Li Chunlei. Inspection of blade profile and machining deviation analysis based on sample points optimization and NURBS knot insertion. Thin-Walled Struct 2021;162:107540.

[18] Zhang Zhongqing, Zhang Zhijing, Jin Xin, Zhang Qiushuang. A novel modelling method of geometric errors for precision assembly. Int J Adv Manuf Technol 2018;94:1139–60.

[19] Feng Gao, Ziyue Pan, Xutao Zhang, Yan Li, Jihao Duan. An adaptive sampling method for accurate measurement of aeroengine blades. Measurement 2021;173:108531.

[20] Ding Siyi, Zheng Xiaohu. Variation analysis considering the partial parallel connection in aero-engine rotor assembly. Energies 2022;15(12):4451.

[21] Grachev IA, Kudashov EV, Bolotov MA, Pronichev ND. Parameterized FEM for estimation of product unit assembly parameters. In: IOP conference series: materials science and engineering, vol. 1047, no. 1. IOP Publishing; 2021, 012049.

[22] Desrochers Alain, Ghie Walid, Laperriere Luc. Application of a unified Jacobian—torsor model for tolerance analysis. J Comput Inf Sci Eng 2003;3(1):2–14.

[23] Ding Siyi, Zheng Xiaohu, Bao Jinsong, Zhang Jie. An improved Jacobian-torsor model for statistical variation solution in aero-engine rotors assembly. Proc Inst Mech Eng B 2021;235(3):466–83.

[24] Zhang Maowei, Liu Yongmeng, Sun Chuanzhi, Wang Xiaoming, Tan Jiubin. Measurements error propagation and its sensitivity analysis in the aero-engine multistage rotor assembling process. Rev Sci Instrum 2019;90(11).

[25] Samper Serge, Adragna Pierre-Antoine, Favreliere Hugues, Pillet Maurice. Modeling of 2D and 3D assemblies taking into account form errors of plane surfaces. 2009.

[26] Schleich Benjamin, Wartzack Sandro. Approaches for the assembly simulation of skin model shapes. Comput Aided Des 2015;65:18–33.

[27] Schleich Benjamin, Anwer Nabil, Mathieu Luc, Wartzack Sandro. Contact and mobility simulation for mechanical assemblies based on skin model shapes. J Comput Inf Sci Eng 2015;15(2):021009.

[28] He Ci, Zhang Shuyou, Qiu Lemiao, Liu Xiaojian, Wang Zili. Assembly tolerance design based on skin model shapes considering processing feature degradation. Appl Sci 2019;9(16):3216.

[29] Qiushuang Zhang, Xin Jin, Zhongqing Zhang, Zhijing Zhang, Ke Shang. Assembly method based on constrained surface registration. J Mech Eng 2018;54(11):70–6.

[30] Guo Yulan, Wang Hanyun, Hu Qingyong, Liu Hao, Liu Li, Bennamoun Mohammed. Deep learning for 3d point clouds: A survey. IEEE Trans Pattern Anal Mach Intell 2020;43(12):4338–64.

[31] Wang Gang, Li Wenlong, Jiang Cheng, Zhu Dahu, Li Zhongwei, Xu Wei, Zhao Huan, Ding Han. Trajectory planning and optimization for robotic machining based on measured point cloud. IEEE Trans Robot 2021;38(3):1621–37.

[32] Chen Hansheng, Wang Pichao, Wang Fan, Tian Wei, Xiong Lu, Li Hao. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 2781–90.

[33] Wang Weiqi, You Xiong, Chen Lingyu, Tian Jiangpeng, Tang Fen, Zhang Lantian. A scalable and accurate de-snowing algorithm for LiDAR point clouds in winter. Remote Sens 2022;14(6):1468.

[34] LeCun Yann, Bengio Yoshua, Hinton Geoffrey. Deep learning. Nature 2015;521(7553):436–44.

[35] Wu Zhirong, Song Shuran, Khosla Aditya, Yu Fisher, Zhang Linguang, Tang Xiaoou, et al. 3D shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, p. 1912–20.

[36] Chang Angel X, Funkhouser Thomas, Guibas Leonidas, Hanrahan Pat, Huang Qixing, Li Zimo, et al. Shapenet: An information-rich 3d model repository. 2015, arXiv preprint arXiv:1512.03012.

[37] Dai Angela, Chang Angel X, Savva Manolis, Halber Maciej, Funkhouser Thomas, Nießner Matthias. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 5828–39.

[38] Hackel T, Savinov N, Ladicky L, Wegner JD, Schindler K, Pollefeys M. Semantic 3d. Net: a New Large-scale Point cloud classification Benchmark. ISPRS Anna Photogramm Remote Sens Spatial Inf Sci 2017;4:91–8.

[39] Qi Charles R, Su Hao, Mo Kaichun, Guibas Leonidas J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 652–60.

[40] Liang Ming, Yang Bin, Wang Shenlong, Urtasun Raquel. Deep continuous fusion for multi-sensor 3d object detection. In: Proceedings of the European conference on computer vision. 2018, p. 641–56.

[41] Zhang Kuangen, Hao Ming, Wang Jing, Chen Xinxing, Leng Yuquan, de Silva Clarence W, et al. Linked dynamic graph cnn: Learning through point cloud by linking hierarchical features. In: 2021 27th International conference on mechatronics and machine vision in practice. IEEE; 2021, p. 7–12.

[42] Su Hang, Maji Subhransu, Kalogerakis Evangelos, Learned-Miller Erik. Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision. 2015, p. 945–53.

[43] Yu Tan, Meng Jingjing, Yuan Junsong. Multi-view harmonized bilinear network for 3d object recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 186–94.

[44] Chen Songle, Zheng Lintao, Zhang Yan, Sun Zhixin, Xu Kai. Veram: View-enhanced recurrent attention model for 3d shape classification. IEEE Trans Visual Comput Graphics 2018;25(12):3244–57.

[45] Maturana Daniel, Scherer Sebastian. Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2015, p. 922–8.

[46] Le Truc, Duan Ye. Pointgrid: A deep network for 3d shape understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 9204–14.

[47] Riegler Gernot, Osman Ulusoy Ali, Geiger Andreas. Octnet: Learning deep 3d representations at high resolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 3577–86.

[48] Qi Charles Ruizhongtai, Yi Li, Su Hao, Guibas Leonidas J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Adv Neural Inf Process Syst 2017;30.

[49] Piegl Les, Tiller Wayne. The NURBS book. Springer Science & Business Media; 1996.

[50] Briot Sébastien, Khalil Wisama, et al. Dynamics of parallel robots. 2015, From rigid bodies to flexible elements. Springer.

[51] Engelmann Francis, Kontogianni Theodora, Leibe Bastian. Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds. In: 2020 IEEE international conference on robotics and automation. IEEE; 2020, p. 9463–9.

[52] Lin Yiqun, Yan Zizheng, Huang Haibin, Du Dong, Liu Ligang, Cui Shuguang, et al. Fpconv: Learning local flattening for point convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 4293–302.

[53] Liu Zhuang, Mao Hanzi, Wu Chao-Yuan, Feichtenhofer Christoph, Darrell Trevor, Xie Saining. A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 11976–86.

[54] Qian Guocheng, Hammoud Hasan, Li Guohao, Thabet Ali, Ghanem Bernard. Assanet: An anisotropic separable set abstraction for efficient point cloud representation learning. Adv Neural Inf Process Syst 2021;34:28119–30.

[55] Sandler Mark, Howard Andrew, Zhu Menglong, Zhmoginov Andrey, Chen Liang-Chieh. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 4510–20.

[56] Hu Jie, Shen Li, Sun Gang. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 7132–41.

[57] Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N, et al. Attention is all you need. Adv Neural Inf Process Syst 2017;30.