www.cad-journal.net
Computer-Aided Design AND APPLICATIONS

Graph Neural Network-Based Finite Element Feature Recognition from B-rep Model

Ryosuke Moriya[1], Satoshi Kanai[2], Hiroaki Date[3], Hideyoshi Takashima[4] and Tetsufumi Taichi[5]

[1]Hokkaido University, r-moriya@eis.hokudai.ac.jp
[2]Hokkaido University, kanai@ssi.ist.hokudai.ac.jp
[3]Hokkaido University, hdate@ssi.ist.hokudai.ac.jp
[4]AIS Hokkaido, Inc., hideyoshi_takashima@ais-hokkaido.co.jp
[5]AIS Hokkaido, Inc., tetsufumi_taichi@ais-hokkaido.co.jp

Corresponding author: Ryosuke Moriya, r-moriya@eis.hokudai.ac.jp

Abstract. Feature recognition using boundary representation (B-Rep) computer-aided design (CAD) models is one of the critical tools bridging CAD/CAM/CAE systems, and many recognition methods have been investigated thus far. Recently, novel learning-based recognition has been proposed to convert a CAD model into a graph and apply a graph neural network (GNN) to the model. This study advances previous methods and proposes a novel GNN-based finite element feature recognition method based on B-Rep CAD models that can handle complex freeform surfaces and curves and that is invariant to translation or rotation, by introducing unique descriptors. We tested the proposed method with our data augmentation (DA) technique on datasets containing finite element features, such as bosses and ribs. The proposed method significantly outperforms a similar state-of-the-art GNN-based method in terms of the recognition performance of rotated objects. Our DA technique could improve recognition performance against datasets slightly different from the training data.

Keywords: B-Rep., Deep Learning, Feature Recognition, Graph Neural Network, Finite Element Meshing
DOI: https://doi.org/10.14733/cadaps.2024.959-975

1 INTRODUCTION

Finite element (FE) analysis is an indispensable digital tool for modern product design and manufacturing. FE analysis requires mesh generation as a preprocess, which partitions a boundary representation (B-Rep) computer-aided design (CAD) model into FE meshes. Because mesh quality significantly affects analysis accuracy, manufacturers specify company-internal mesh generation rules for some FE features on CAD models, such as bosses and ribs, including free-form surfaces,

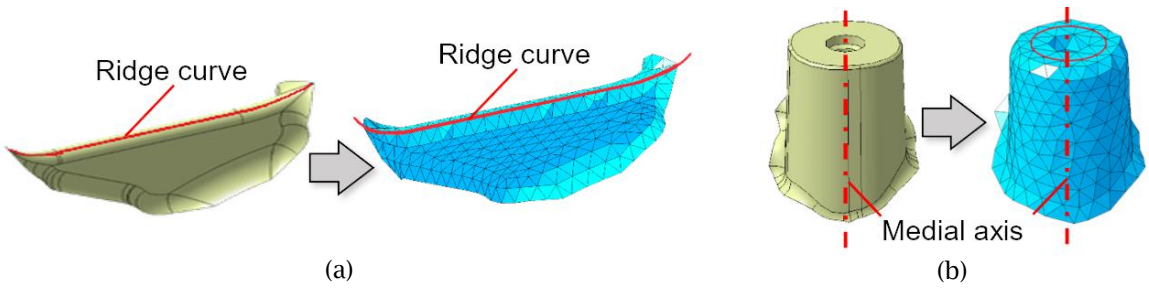


Figure 1: Typical examples of company-internal mesh generation rules for features: (a) rib case and (b) boss case. In a rib, mesh vertices should be aligned with the ridge curve. In a boss, mesh vertices should be positioned concentrically around the medial axis.

as depicted in Figure 1. However, recognizing FE features from CAD models with complex geometries heavily depends on manual intervention by expert designers, making it time-consuming and error-prone. Therefore, a reliable, versatile, and fully automatic FE feature recognition method from CAD models is urgently required for efficient, high-quality mesh generation.

Several deep neural networks (DNN)-based methods have recently been proposed for feature recognition from CAD models. Unlike classical rule-based recognition methods, DNN-based methods have the advantage of not requiring algorithm design specific to specific feature types. However, most DNN-based methods approximate CAD model geometries with voxels [29] or point clouds [26] as an input of DNNs, resulting in discretization loss in model resolution or an increase in the data size, which results in more significant memory consumption, longer training time, or poor recognition performance. Some DNN-based recognition methods have been proposed to address these issues in recent years [2, 3, 8, 9, 14], which use a "graph" as an input of the DNN and benefit from high compatibility with B-Rep CAD models. Nevertheless, these methods also have limitations. First, they are not rotation-invariant, and the recognition performance strongly depends on the model poses. Second, they were designed to extract machining features or geometric modeling procedures from CAD models and were not tested with FE features containing freeform surfaces.

This study proposes a novel FE feature recognition method from a B-Rep CAD model using a graph neural network (GNN), which has a recognition ability invariant to model rotation. The proposed method comprises a graph construction method with descriptors invariant to model rotation, a GNN capable of labeling faces in the CAD model for feature recognition, and a data augmentation (DA) technique for robust recognition. We tested our method with an original FE feature dataset, containing bosses and ribs, and compared its performance with that of an existing method [1] using two previous GNNs [8, 14], similar to ours. Although we designed the proposed method to work on FE feature recognition like [13, 16, 17, 26, 28], it can be applied to form feature recognition in other domains like product design [19], data exchange for product lifecycle management systems [6], process planning [2, 3, 5, 10, 11, 20, 24, 25, 29], and reverse engineering [14].

2 RELATED WORK

A "feature" is recognized as a set of faces in a B-Rep CAD model with higher semantics than simple geometries. Various types of features are defined based on different interests, including product design [6, 14, 19], machining [2, 3, 5, 10, 11, 20, 24, 25, 29], and FE analysis [13, 16, 17, 26, 28]. Feature recognition is a technology that searches for features from a given CAD model and can play an important role in process integration between CAD/computer-aided manufacturing/computer-aided engineering systems because they require different interpretations of model geometries depending on the processing purposes.

Feature recognition methods for three-dimensional (3D) CAD models can be categorized into two main approaches: rule-based and learning-based. In this section, we will introduce representative studies on these approaches and their limitations.

2.1 Rule-based Approach

The rule-based approach searches for features based on predefined rules defining feature extraction procedures.

One of the famous rule-based algorithms is the volumetric decomposition of input CAD models. This algorithm represents a model as a combination of simple volumes and searches for features. Kim proposed a feature recognition method that uses convex decomposition called alternating sum of volumes with partitioning (ASVP) [11]. ASVP decomposes a model into multiple boolean operations originating in convex objects. Meanwhile, Sakurai proposed a feature recognition method using volumetric decomposition based on cell decomposition, which partitions a model into small cells and searches for a combination of cells that matches features [23].

Another famous rule-based approach is a graph-based algorithm. This algorithm converts a CAD model into a graph and searches for subgraphs representing features. Graphs are suited to represent models because they have high affinity with a CAD model data structure "B-Rep.," a hierarchical structure of solids, shells, faces, edges, vertices, etc. For example, Joshi and Chang proposed using an attributed adjacency graph (AAG) to recognize machined features [10]. An AAG is a graph in which CAD faces and edges are represented as graph nodes and links, respectively, with all links attributed concave or convex based on the angle between the faces. As another example, Guo et al. proposed using the weighted attributed adjacency matrix, which is a matrix representing graph adjacency with attributes such as edge type or angle between faces [5]. Another graph-based approach has been recently proposed by Onodera et al. for the extraction of FE-oriented features from a CAD model [17].

There are some approaches not categorized into volumetric decomposition or graph-based algorithms. One example is Shape Terra proposed by Harik et al. [7], which applies a "heat retention" value to each mesh vertex. Heat retention is a value defined by the time integral of the vertex temperature until it reaches below the threshold, based on the heat diffusion simulation on the model. Feature recognition is achieved by clustering vertices with similar heat retention values. Shi et al. demonstrated that the idea of Shape Terra can be used for manufacturability analysis in additive manufacturing [25].

The rule-based approach has the advantage of not requiring a large dataset or long training time in advance and having an explicit feature derivation process. However, this approach has drawbacks in that the extraction rules need to be carefully designed by experts for each feature type, i.e., it lacks versatility. Moreover, it is challenging to realize robust recognition with this approach because real-world models do not exactly match the assumed topological structure of the feature designed by and implemented in the rules, especially when features intersect or interact.

2.2 Learning-based Approach

Learning-based approaches automatically learn the structures of feature geometry from training datasets and do not require any recognition rules designed by human experts.

As a classical study before the deep learning era, Prabhakar and Henderson proposed applying a neural network to a face adjacency matrix with values that code the geometric and topological characteristics of faces and edges to derive form features [20]. The algorithm performs pattern recognition on each row of the matrix using the network. As the neurons in the network in the study use only integers with a small number of digits as input or output, the values representing the features are also integers and do not use real numbers. Therefore, the geometry of face segments that can be encoded is restricted to planar and cylindrical surfaces whose normals are oriented in a particular orthogonal axis direction.

Following the massive success of deep learning methods, especially convolutional neural networks (CNNs) in image processing, some studies have attempted to implement 3D CAD model feature recognition using various CNN-based techniques. The first approach is to utilize images of a model rendered from multiple views. Shi et al. proposed MsvNet for feature recognition of CAD models, using multiple orthogonal sectional views [24]. Each sectional view of the model is processed with a CNN, and aggregation of multiple views is performed through a pooling layer. The second approach is to extend a CNN for images to three dimensions and apply it to a voxel space for feature recognition. As a typical example, Zhang et al. proposed FeatureNet, which uses a 3D CNN for feature recognition from 3D CAD models [29]. In FeatureNet, 64 × 64 × 64 voxels are used to classify a feature in a 10-cm cubic space.

Another 3D model representation frequently used with deep learning is a point cloud. For deep learning with point clouds, networks called PointNet [21] or PointNet++ [22] are well known. Takashima et al. proposed converting the boundary faces of a 3D CAD model into a point cloud and recognizing freeform FE features from it using PointNet++ [26].

Multiple views, point clouds, and voxels are commonly used 3D representations for deep learning because well-studied feature recognition methods based on images using CNNs can be extended straightforwardly. However, these representations also have some drawbacks. One of the significant drawbacks is that they are simply approximations of an original CAD model, and the conversion results in a loss of geometric and topological information and degradation of the resolution of the geometry. To avoid degradation, the model needs to be sampled at high resolutions, eventually leading to increased data size and inefficient computation for learning.

One solution is to use "graphs," which have elements that correspond one-to-one with the topological elements of a B-Rep CAD model, while point clouds and voxels require multiple elements to represent a single face. In this context, recent works have been published that attempt to perform feature recognition on CAD models using GNNs, which are directly capable of deep learning on graphs.

For example, Cao et al. proposed using a face adjacency graph (FAG) with node descriptors for machining feature recognition [2]. An FAG is a graph whose nodes and links represent CAD faces and edges, respectively. It uses the coefficients of the plane equation as a node descriptor. For this reason, it is only applicable to models bounded only by planes. Jones et al. weakened this restriction to planes, cylinders, cones, spheres, and tori for surfaces and lines, circles, and ellipses for edges and applied it to feature recognition [9]. They also used surface parameters, surface/curve type, and orientation for node descriptors. Nevertheless, there was a limitation on the surface/curve type. UV-net proposed by Jayaraman et al. [8] solved this problem by calculating the descriptors from rectangular grids on the two-dimensional parameter domain of faces and the one-dimensional domain of edges. However, because they directly utilize xyz coordinates and normal or tangent vectors of faces for descriptor calculation, the feature recognition results become rotation-variant even though the model shape will not change with model rotation. Hierarchical-CADNet proposed by Colligan et al. [3] also has a rotation-variant limitation. Conversely, in BRepNet proposed by Lambourne et al. [14], simple descriptors such as surface types and face area enable rotation invariance on feature recognition. Although their main proposal is a convolution kernel on the graph defined from the B-Rep structure, the method has low discriminative ability among geometries. Furthermore, the abovementioned studies have rarely shown examples of feature recognition from complex part geometries, such as FE features appearing on realworld product models, which have freeform surface geometries and multiple features interacting with each other, and most of them have performed feature classification when only a single feature exists on the part.

From the above review, while GNNs can directly perform deep learning-based feature extraction on the topological data structure of B-Rep CAD models, the descriptor design on graphs used in previous studies still suffers from the following issues:

1. Poor discrimination between feature geometries when only the surface type or face area is used as descriptors,

2. Limited applicable surface geometries due to the intrinsic geometric parameters of surfaces as descriptors, and
3. Inability to recognize features in a rotation-invariant way because of the dependence of descriptors on coordinate systems such as XYZ coordinates.

This study proposes a new FE feature recognition method that addresses the above issues by constructing rotation-invariant descriptor representations of nodes and edges for GNNs, enabling the recognition of complex FE features, including freeform surfaces, from B-Rep CAD models.

3 PROPOSED FE FEATURE RECOGNITION METHOD BASED ON GRAPH NEURAL NETWORKS

3.1 Overview

Figure 2 shows the recognition pipeline of the proposed FE feature recognition method based on GNNs.

The first step converts B-Rep CAD model data into an FAG with descriptors. The FAG is a graph in which B-Rep faces and edges are represented as nodes and links, respectively. Each node and link in the FAG has descriptors, a multidimensional vector encoding the corresponding face or edge geometry. In the second step, the FAG is input into a GNN that classifies the labels of each node. The GNN is trained in advance with CAD model datasets comprising FE features whose faces are labeled with feature labels. In the last step, the estimated face feature labels are extracted from the GNN output and reflected in the original CAD model.

Notably, the following assumptions are made in the proposed method:

- Each face in a B-Rep CAD model has exactly one feature label, and
- All CAD models used in the training and testing phases have no topological and geometric defects in product data quality (PDQ), and they are watertight solid models.

The former assumption results in the limitation that our feature extraction method cannot be applied to cases where a single face belongs to two or more FE features. However, this has not been the case in practical models that we and our collaborator's company have worked on. This problem can be solved by adopting a preprocessing method that divides B-Rep faces into patches according to their features; nevertheless, further research is required in this respect.

3.2 Descriptors on FAG

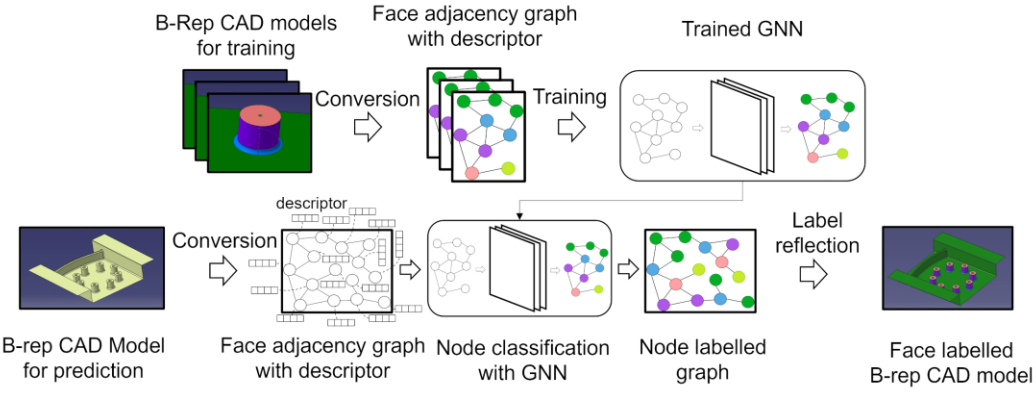


Figure 2: The proposed FE feature recognition pipeline.

The proposed method encodes the topology and geometry of a B-Rep CAD model in terms of FAG connection relations and descriptors attached to the nodes and links of the FAG, respectively. To

achieve pose-invariant feature recognition, we defined the following descriptors invariant to translation and rotation. Moreover, considering the classes of geometries to be recognized, the descriptors that can discriminate a broad range of face geometries, including freeform surfaces in B-Rep CAD models, are attached to the nodes and links of the FAG. These node and link descriptors are described below.

The node descriptor $\boldsymbol{F}_n$ encodes the geometry of a face. It comprises the following two descriptors $\boldsymbol{F}_{SI}$ and $\boldsymbol{F}_{OBB}$ calculated from sampled points on the face.

1) Shape index distribution ($\boldsymbol{F}_{SI}$):

$\boldsymbol{F}_{SI}$ encodes a statistical distribution of the shape index [12] $SI(p)$ at a sampled point p on a face. $SI(p)$ is defined by Equation (2.1), from principal curvatures κ_1, κ_2 $(\kappa_1 \geq \kappa_2)$ at p. The index ranges from -1 to 1 except for the case where p is on planes, as depicted in Figure 3.

$$SI(p) = \frac{\pi}{2}\tan^{-1}\frac{\kappa_2 + \kappa_1}{\kappa_2 - \kappa_1} \tag{2.1}$$

The descriptor $\boldsymbol{F}_{SI}$ is defined as a normalized histogram of $SI(p)$ over the face. $\boldsymbol{F}_{SI}$ of the node corresponding to a face f_i is defined by Equations (2.2) and (2.3) with l denoting the number of intervals (we used $l = 7$) and P_i^f denoting the sampled point set of the face f_i. A special bin is provided in the histogram for the points lying on the plane, since $SI(p)$ cannot be defined mathematically if p is on the plane where $\kappa_1 = \kappa_2 = 0$. The number of intervals l should be set according to the feature to be recognized, but we chose a value that allows us to assign spheres $(SI(p) = \pm 1)$ and cylinders $(SI(p) = \pm 0.5)$ to different bins in the histogram while avoiding setting the bin border to zero because the sign of the value may be unstable around zero. However, sample points with a shape index near zero typically do not appear frequently.

$$\boldsymbol{F}_{SI}(f_i) = \frac{1}{\sum_{i=0}^{l}|A_i(f_i)|}[|A_0(f_i)|, |A_1(f_i)|, \cdots, |A_l(f_i)|]^T, \tag{2.2}$$

$$A_i(f_i) = \begin{cases} \left\{p \mid -1 + \dfrac{2i}{l} \leq SI(p) \leq -1 + \dfrac{2(i+1)}{l}, p \in P_i^f\right\} (i = 0,1,\cdots,l-1) \\ \{p \mid p \text{ is on planes}, p \in P_i^f\} (i = l) \end{cases}, \tag{2.3}$$

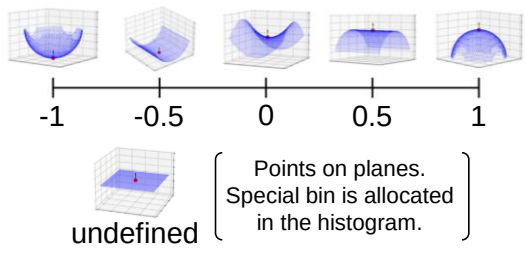


Figure 3: Typical geometries and their shape index values in the shape index distribution ($\boldsymbol{F}_{SI}$).

2) Oriented bounding box (OBB) aspect ratio ($\boldsymbol{F}_{OBB}$):

$\boldsymbol{F}_{OBB}$ encodes the proportions of three edge lengths l_1, l_2, l_3 $(l_1 \geq l_2 \geq l_3)$ of the OBB that envelops the points sampled on a face, as depicted in Figure 4 and defined by Equation (2.4):

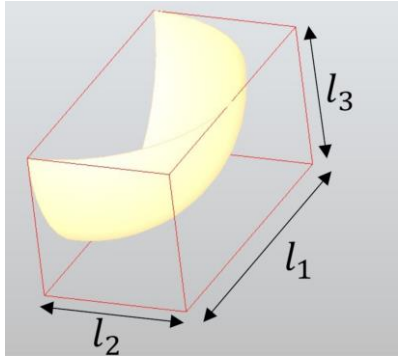


Figure 4: OBB and its edge length l_1, l_2, l_3.

$$\boldsymbol{F}_{OBB} = \frac{[l_1, l_2, l_3]^T}{\|[l_1, l_2, l_3]^T\|} \tag{2.4}$$

Conversely, the link descriptor $\boldsymbol{F}_l$ represents an angular relation between two adjacent faces connected to an edge, which is used as the indicator of the connecting condition and the relative orientation between adjacent faces while maintaining translation and rotation invariance. The following two types of angles are adopted for the link descriptor: local face angle F_{la} and global face angle F_{ga}. Similar to the node descriptor, the link descriptor is calculated from the points sampled on the edge corresponding to a link.

3) Local face angle (F_{la}):

F_{la} denotes an average of signed angles between normal vectors of the adjacent faces at the points sampled on an edge. F_{la} can indicate whether the adjacent faces connect concavely or convexly by its sign and angle value. The signed angle $\theta_{la}(q)$ at a point q on an edge is derived from Equation (2.5):

$$\theta_{la}(q) = \frac{1}{\pi}\mathrm{sgn}\left(\left(\boldsymbol{n}_i \times \boldsymbol{n}_j\right) \cdot \boldsymbol{t}_i\right)\cos^{-1}\left(\boldsymbol{n}_i \cdot \boldsymbol{n}_j\right), \tag{2.5}$$

where $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$ denote unit normal vectors at q in faces f_i and f_j adjacent to the edge respectively, and $\boldsymbol{t}_i$ denotes a unit tangent vector at q of the corresponding half edge belonging to the face f_i. Figure 5 shows the relation between $\boldsymbol{n}_i \times \boldsymbol{n}_j$ and $\boldsymbol{t}_i$. F_{la} is calculated by averaging $\theta_{la}(q)$ over the sampled point set $Q = \{q\}$ on the edge and is derived from Equation (2.6):

$$F_{la} = \frac{1}{|Q|}\sum\nolimits_{q \in Q} \theta_{la}(q). \tag{2.6}$$

Although F_{la} is a useful indicator of the adjacent face relation that focuses on the region close to their connecting edge, it becomes zero if two faces are connected smoothly, and it is insufficient to represent the face relation solely using F_{la} in such a case, which frequently occurs in FE features. Therefore, we introduced the other link descriptor F_{ga} to encode the overall orientation between the adjacent faces that does not take a zero value even in such cases.

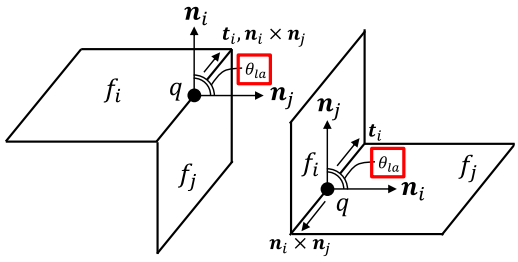


Figure 5: Local face angle (F_{la}).

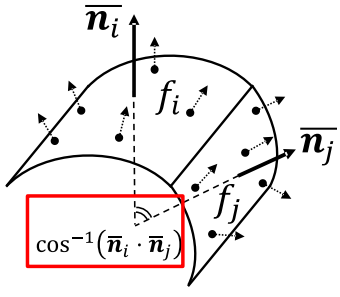


Figure 6: Global face angle (F_{ga}).

4) Global face angle (F_{ga}):

F_{ga} denotes an angle between the averaged normal vectors $\overline{\boldsymbol{n}}_i$ and $\overline{\boldsymbol{n}}_j$ on two adjacent faces f_i and f_j connected to the edge, as depicted in Figure 6, where $\overline{\boldsymbol{n}}_i$ is evaluated by taking an average of the normal vectors at the sampled points on f_i. F_{ga} is defined by Equation (2.7):

$$F_{ga} = \frac{1}{\pi}\cos^{-1}\left(\overline{\boldsymbol{n}}_i \cdot \overline{\boldsymbol{n}}_j\right). \tag{2.7}$$

F_{ga} has the disadvantage of being significantly influenced by the shape of the face because it is calculated using the averaged normal vector over each face. Overall, F_{la} and F_{ga} are meant to complement each other.

Because the descriptors are calculated based on random sampling points over a face or an edge, their values fluctuate depending on the positions randomly selected for the sampled points. To confirm that the extent of fluctuations is acceptably low, we evaluated the statistical distribution of the descriptor values of a simple-shaped model shown in Figure 7 with different sampling positions. We generated 50 sampled points in each face and 11 sampled points in each edge. Tables 1 and 2 show the results. The column F_{SI} of Table 1 shows only the fluctuation of the maximum element among the histogram elements. On the basis of these results, we concluded that the fluctuation of the descriptor values depending on the sampling position is sufficiently small and acceptable for encoding the geometric properties of faces and edges. If the extent of the fluctuation needs to be reduced, we can simply increase the sampling density.

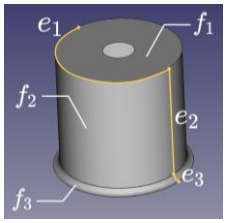


Figure 7: A model for descriptor fluctuation evaluation.

	F_{SI}		$F_{OBB,1}$		$F_{OBB,2}$		$F_{OBB,3}$	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
f_1	1.000	0.000	0.723	0.024	0.690	0.027	0.000	0.000
f_2	1.000	0.000	0.701	0.022	0.628	0.027	0.336	0.025
f_3	1.000	0.000	0.889	0.011	0.456	0.020	0.044	0.003

Table 1: Node descriptor fluctuations ($\boldsymbol{F}_{OBB} = [F_{OBB,1},\ F_{OBB,2},\ F_{OBB,3}]$).

	F_{la}		F_{ga}	
	Mean	Std.	Mean	Std.
e_1	0.500	0.000	0.500	0.000
e_2	0.000	0.000	0.952	0.037
e_3	0.000	0.000	0.361	0.034

Table 2: Link descriptor fluctuations.

3.3 Node Classification by GNN

Figure 8 shows the GNN structure used for node classification for feature recognition in this study. As depicted in Figure 8, the network comprises two phases: convolution and multilayer perceptron (MLP).

The convolution phase comprises three "MLP & Conv" layers. These layers perform three operations. First, each node/link descriptor is input into several MLP layers and the batch normalization layer sequentially. Second, node descriptors are convoluted into link descriptors defined by Equation (2.8):

$$\boldsymbol{e}'_{ij} = \sigma\left(\mathrm{BN}\left(\boldsymbol{W}(\boldsymbol{x}_i \oplus \boldsymbol{x}_j \oplus \boldsymbol{e}_{ij})\right)\right), \tag{2.8}$$

where $\boldsymbol{x}_i$ denotes the descriptor of node i before the operation, $\boldsymbol{e}_{ij}$ and $\boldsymbol{e}'_{ij}$ denote the descriptors before and after the operation of the link between node i and j respectively, $\mathrm{BN}()$ denotes the batch normalization layer, $\sigma()$ denotes the activation function, $\oplus$ denotes vector concatenation, and $\boldsymbol{W}$ denotes the learning parameters. Finally, the descriptors are convolved between adjacent nodes using graph attention networks [27], as defined by Equation (2.9):

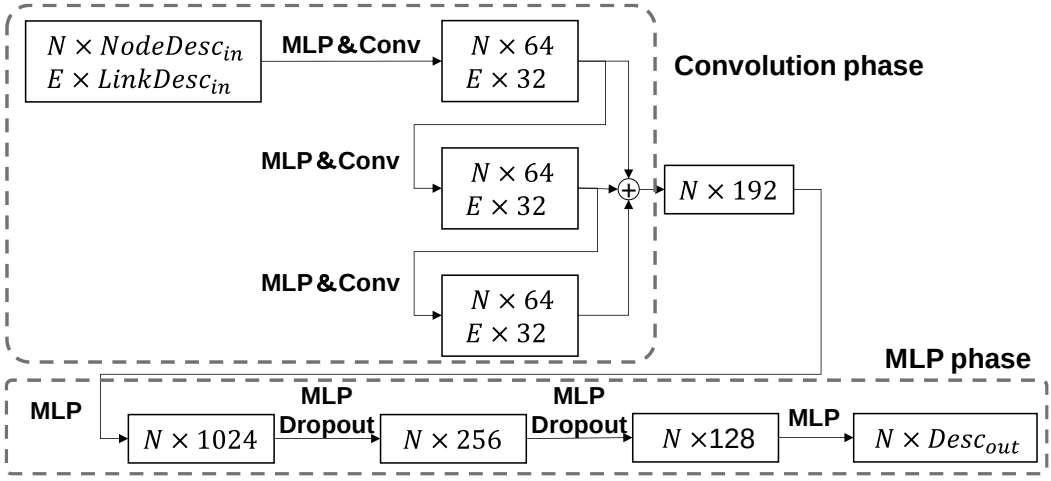


Figure 8: GNN structure used for node classification. Each square describes the descriptor size attached to nodes and links. N and E denote the numbers of nodes and links respectively, $NodeDesc_{in}$ and $LinkDesc_{in}$ denote the sizes of input descriptors of nodes and links respectively, and $Desc_{out}$ denotes the output size of node descriptors.

$$\boldsymbol{x}_i' = \sigma\left(\sum_{j\in\mathcal{N}_i} \frac{\exp\left(\sigma(\boldsymbol{W}_n x_i \oplus \boldsymbol{W}_n \boldsymbol{x}_j \oplus \boldsymbol{W}_e \boldsymbol{e}_{ij})\right)}{\sum_{k\in\mathcal{N}_i}\exp(\sigma(\boldsymbol{W}_n x_i \oplus \boldsymbol{W}_n \boldsymbol{x}_k \oplus \boldsymbol{W}_e \boldsymbol{e}_{ik}))} \boldsymbol{W}\boldsymbol{x}_j\right), \tag{2.9}$$

Meanwhile, the MLP phase consists of fully connected layers with a stochastic dropout, classifying each node based on the convolution phase output. Outputs of the MLP phase are input into a softmax function, which predicts the likelihood that each node belongs to a certain feature label. The feature label indicates what role the face corresponding to a node plays in the feature. For example, in the case of a boss feature, there are three types of feature labels: "Boss Hole," "BossTop," and "BossSide."

In the training process of the network, the cross-entropy loss was evaluated as the optimization criterion, and the Adam optimizer was used.

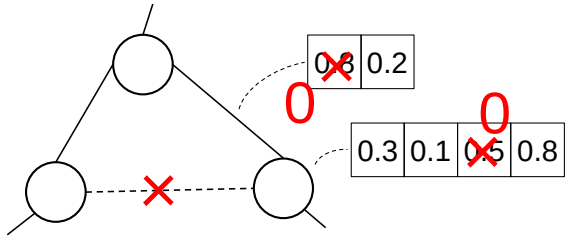


Figure 9: Example of DA on FAG.

3.4 Data Augmentation

We performed DA in each training epoch to increase recognition robustness. We intentionally introduced noise into the graph used in the training dataset. This aims to make feature recognition more stable against small differences in data and prevent the recognition mechanism from being overly reliant on a single descriptor or FAG graph. FAG data can be broadly classified into two aspects, graph structure, which represents B-Rep topology, and descriptors, which represent B-Rep geometry. Because we wanted to augment both in DA, we removed links of the input FAG and overwrote the elements of descriptors to 0 with a certain probability (0.15 in our experiment), as depicted in Figure 9. DA was applied only to the training dataset and not to the validation dataset.

Compared with modifying CAD models for training in a CAD system for augmentation, augmenting directly on graphs has the advantage of being significantly computationally simple. Although this augmentation technique could generate FAGs corresponding to topologically invalid solid models, where adjacent faces do not have a link or descriptors are not normalized, it was found to contribute to improved recognition performance. The results are discussed in section 4.3.

4 CASE STUDY

4.1 Datasets

We used two datasets to evaluate recognition performance; one is our original dataset with freeform FE features, and the other is the Fusion360 gallery segmentation dataset [14], which is publicly available.

Our original dataset consists of two types of models: "Simple Models" and "Practical Models." "Simple Models" shown in Figure 10(a) include 32,940 CAD models that comprise simple combinations of boss, rib, and fillet features and are used throughout the training, validation, and testing phases. Meanwhile, "Practical Models" shown in Figure 10(b) include only three models that comprise complex combinations of these features and are used in the testing phase to evaluate recognition methods under more realistic conditions.

Generally, preparing many CAD models and labeling all faces in them is a timeconsuming task and limits learning-based feature recognition applications. To ease the difficulty of the task, we created "Simple Models" by only varying size parameters such as radius or height of the 17 handcrafted "template" CAD models shown in Figure 10(a). By adopting this method, we only needed to build 17 template models, label their faces, set some size parameters and rules for the parameter variation, and generate valid parameters. However, because the parameter variation changes only geometries or descriptor values and does not alter any topologies of FAG structures, the dataset can be less diverse in terms of topologies. Using this type of dataset can make recognition unstable to changes in the FAG structure, which can be caused by feature interaction. To address this issue, we adopted a DA method that stochastically removes the edges of the FAG, which could increase the topological diversity of the FAG on training.

All faces in the CAD model in the original dataset were labeled with seven feature labels, as depicted in Figure 10(b): "BossTop," "BossSide," "BossHole," "RibTop," "RibSide," "Fillet," and "None." These features, such as bosses, ribs, and fillets, are typical FE features defined by the company-internal mesh generation rules described in section 1.

The sublabels "Top," "Side," and "Hole" attached to each feature label are essential in mesh generation. This is because FE analysis preprocessor software typically includes an automatic mesh generation function, and specific constraint points on the feature (e.g., the center point of a boss top face) must be passed to the mesh generation function to generate a mesh according to the constraint mesh generation rules. These sublabels are essential for identifying the constraint point from the feature and the reference point from which the mesh is generated.

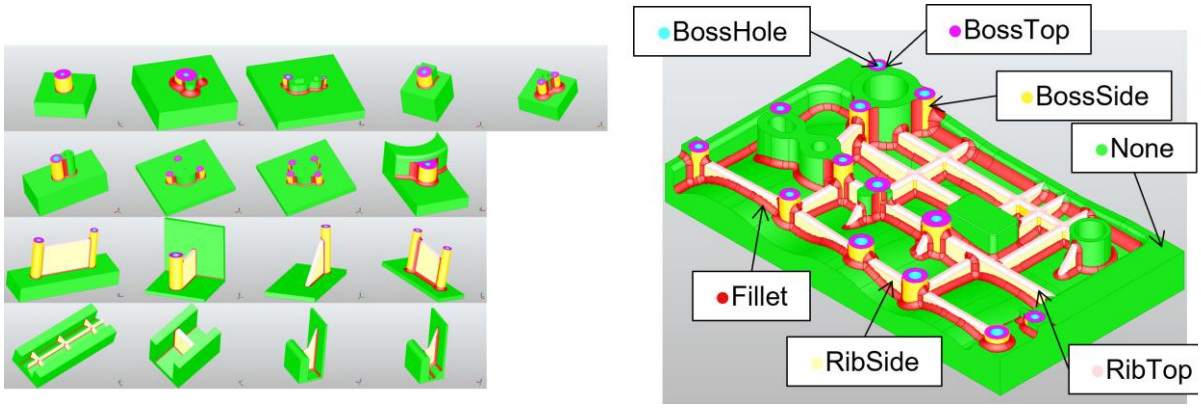


Figure 10: (a) 17 templates of "Simple Models," and (b) an example of "Practical Models" with feature labels.

4.2 Experimental Condition

We implemented the proposed GNN-based feature recognition method using PyTorch Geometric [4] and Open CASCADE Technology [18]. We performed 5-fold cross-validation and compared our results with the recognition results of the combined method of BRepNet [14] and UV-Net [8] (hereafter referred to as a "baseline method"), which are GNN-based feature recognition methods like ours. All measurements were conducted with the model of minimum validation loss. To evaluate rotation invariance, the test was conducted with both a posture similar to the training data and a randomly rotated posture. Performance was evaluated using two metrics, accuracy per face (Acc) and mean intersection over union calculated per face (mIoU). As mIoU is the average of the IoU of each feature label, mIoU increases when correctly predicting the labels of faces that contain only a few faces rather than the labels of faces that contain many faces in the dataset.

4.3 Results

4.3.1 Original dataset

Table 3 summarizes the recognition performance, and Figure 11 shows an example of the recognition results for practical models. Furthermore, Figure 12 shows the results of the feature ablation study, which indicates the change in performance when removing some descriptors.

Method	Simple Models (Original Pose)		Simple Models (Rotated)		Practical Models (Original Pose)		Practical Models (Rotated)	
	Acc [%]	mIoU [%]	Acc [%]	mIoU [%]	Acc [%]	mIoU [%]	Acc [%]	mIoU [%]
Our Method	99.43	98.83	**99.43**	**98.83**	77.02	48.21	76.99	48.07
Our Method + DA	99.43	98.83	**99.43**	**98.83**	**82.06**	**54.09**	**81.85**	**53.55**
BRepNet + UV-Net	**100.00**	**100.00**	62.70	28.33	71.92	47.46	48.09	14.23

Table 3: Recognition performance on our original dataset. Acc denotes accuracy per face, and mIoU denotes mean intersection over union per face. "DA" means the result obtained when using the training dataset with DA (section 3.4)

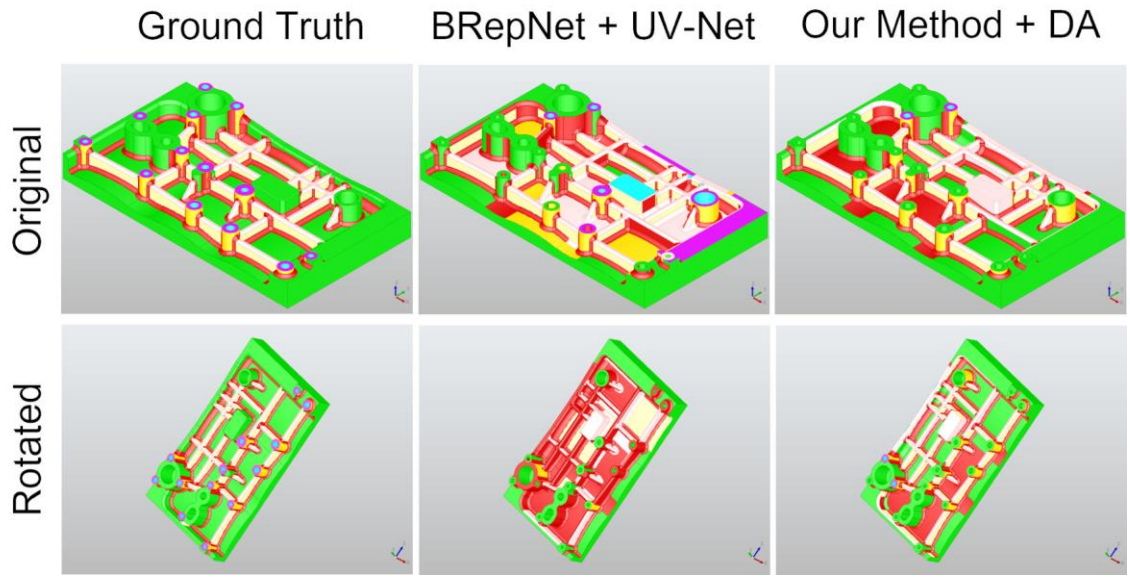
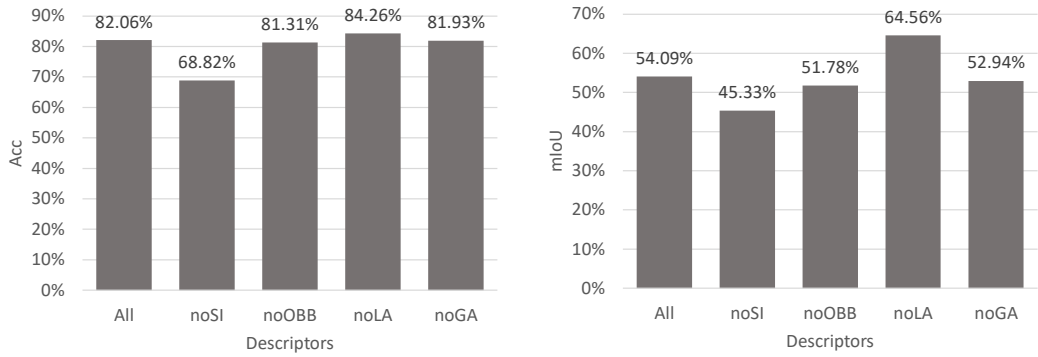


Figure 11: Example of recognition result.



(a) Accuracy per face

(b) mIoU per face

Figure 12: Results of feature ablation study. Each label on the descriptors denotes as follows: noSI: without $\boldsymbol{F}_{SI}$; noOBB : without $\boldsymbol{F}_{OBB}$; noLA: without F_{la}; noGA: without F_{ga}. DA was performed during training, and the performance was evaluated using Practical Models with the original pose.

As presented in Table 3, the proposed and baseline methods exhibited high recognition performance for the simple models with the original pose. In contrast, the performance of both methods for the practical models slightly decreased to 71%–82% in accuracy and 47%–54% in mIoU. The reason for the degradation could be that the practical models were significantly more complex and interacted more between bosses and ribs.

In the evaluation with the practical models, our method's performance was slightly higher than the baseline method in the original pose. However, when a rotation was applied to the model, the

performance of the baseline method degraded significantly. Conversely, our method did not exhibit any degradation under model rotation because it is rotation-invariant.

Furthermore, it was found that the proposed DA technique described in section 3.4 contributed to an improvement of approximately 6% in performance. This could be due to the improved topological diversity of the graphs used in the training dataset.

Finally, the feature ablation study shown in Figure 12 implied that the shape index distribution $\boldsymbol{F}_{SI}$ played an essential role in our proposed method.

In summary, for feature recognition from CAD models, such as bosses and ribs composed of complex curved surfaces, under arbitrary model rotations, our proposed method using GNNs with DA outperformed the baseline method.

4.3.2 Fusion 360 gallery segmentation dataset

Next, to evaluate the effectiveness of the proposed method on datasets containing other CAD models different from ours, a performance evaluation was performed on the Fusion 360 dataset [14]. Table 4 summarizes the feature recognition performance on the Fusion360 gallery segmentation dataset with the original pose and randomly rotated using our method and BRepNet [14] + UV-Net [8]. This segmentation dataset includes labeled CAD models with geometries similar to our FE feature dataset. The feature labels are attached according to the CAD modeling operation used to create each face. Each face had one of eight feature labels, such as "Extrude Side," "Extrude End," "Fillet," and "Chamfer." CAD models of 35,680 parts were included in the dataset. The face geometries of the models were dominated by planes, cylindrical surfaces, spheres, and tori, with few freeform surfaces included.

The result in Table 4 showed that with the original pose, BRepNet + UV-Net outperformed the proposed method. This suggests that our rotation-invariant descriptors have a lower ability to discriminate geometries than UV-Net, or the convolution kernel of BRepNet, which is based on the graph of B-Rep structure, is designed and optimized to the Fusuion 360 dataset and outperforms our simple GNN.

Nevertheless, rotation did not influence the performance of our method, but it significantly degraded the performance of BRepNet + UV-Net, which was worse than ours. Furthermore, our DA technique was less effective with this dataset than with our original dataset. This might be because this dataset has sufficient diversity in the training dataset or more similarity between the training and test datasets than our original dataset.

5 CONCLUSIONS

We proposed a novel GNN-based rotation-invariant FE feature recognition method from a B-Rep CAD model. The CAD model data were converted into an FAG with descriptors. Four rotation-invariant descriptors were proposed as the descriptors for the nodes and links of the FAG, and these descriptors could code the complex geometries of freeform faces and edges. From the input of the FAG, the GNN could directly classify the labels of each node that indicated feature labels of faces of

Method	Original Pose		Rotated	
	Acc [%]	mIoU [%]	Acc [%]	mIoU [%]
Our Method	76.64	47.16	**76.80**	**44.90**
Our Method + DA	75.00	44.23	75.21	42.33
BRepNet + UV-Net	**93.81**	**73.27**	75.02	42.86

Table 4: Recognition performance on Fusion360 gallery segmentation dataset [14].

the CAD model. DA techniques that directly apply to the FAG were also proposed for robust recognition.

The performance of our method and a similar state-of-the-art GNN-based method (BRepNet + UV-Net) was compared on CAD model datasets, containing FE features. The results showed that the proposed method significantly outperformed the GNN-based method in terms of the recognition performance of rotated objects. The proposed DA technique could improve the performance against datasets slightly different from the training data.

However, the following issues remain unaddressed and should be addressed in future work:

- Improvement in recognition performance against complex models. This can be achieved by changing the dataset generation method or using transfer learning.
- Alleviation of the strong dependency of the FAG and descriptor values on the B-Rep structure. This can lead to the issue that two identical features can be recognized in a completely different manner because of their different B-Rep structures. One possible solution to this problem is to employ some normalization procedures of B-Rep models such as "maximal faces" [15].
- Survey on the influence of PDQ defects on recognition performance.
- Enabling the recognition of models, where a single face belongs to two or more FE features, as mentioned in section 3.1.

Ryosuke Moriya, https://orcid.org/0009-0004-3616-6392
Satoshi Kanai, https://orcid.org/0000-0003-3570-1782
Hiroaki Date, https://orcid.org/0000-0002-6189-2044
Hideyoshi Takashima, https://orcid.org/0000-0002-7158-6059
Tetsufumi Taichi, https://orcid.org/0009-0006-0748-8303

REFERENCES

[1] Autodesk AI Lab: Fusion 360 gallery dataset, https://github.com/AutodeskAILab/Fusion360GalleryDataset

[2] Cao, W.; Robinson, T.; Hua, Y.; Boussuge, F.; Colligan, A. R.; Pan, W.: Graph representation of 3D CAD models for machining feature recognition with deep learning, Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 11A, 2020, V11AT11A003. https://doi.org/10.1115/DETC2020-22355

[3] Colligan, A. R.; Robinson, T. T.; Nolan, D. C.; Hua, Y.; Cao, W.: Hierarchical CADNet: learning from B-Reps for machining feature recognition, Computer-Aided Design, 147, 2022, 103226. https://doi.org/10.1016/j.cad.2022.103226

[4] Fey, M.; Lenssen, J. E.: Fast graph representation learning with PyTorch geometric, arXiv, 2019. https://doi.org/10.48550/arXiv.1903.02428

[5] Guo, L.; Zhou M.; Lu Y.; Yang, T.; Yang, F.: A hybrid 3D feature recognition method based on rule and graph, International Journal of Computer Integrated Manufacturing, 34(3), 2021, 257-281. https://doi.org/10.1080/0951192X.2020.1858507

[6] Gupta, R.K.; Gurumoorthy, B.: Automatic extraction of free-form surface features (FFSFs), Computer-Aided Design, 44(2), 2012, 99-112. https://doi.org/10.1016/j.cad.2011.09.012

[7] Harik, R.; Shi, Y.; Baek, S.: Shape Terra: mechanical feature recognition based on a persistent heat signature, Computer-Aided Design and Applications, 14(2), 2017, 206-218. https://doi.org/10.1080/16864360.2016.1223433

[8] Jayaraman, P. K.; Sanghi, A.; Lambourne, J. G.; Willis, K. D. D.; Davies, T.; Shayani, H.; Morris, N.: UV-net: Learning from boundary representations, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, 11698-11707. https://doi.org/10.1109/CVPR46437.2021.01153

[9] Jones, B. T.; Hu, M.; Kim, V. G.; Schulz, A.: Self-supervised representation learning for CAD, arxiv, 2022. https://doi.org/10.48550/arXiv.2210.10807
[10] Joshi, S.; Chang, T. C.: Graph-based heuristics for recognition of machined features from a 3D solid model, Comptuer-Aided Design, 20(2), 1988, 58-66. https://doi.org/10.1016/0010-4485(88)90050-4
[11] Kim, Y. S.: Recognition of form features using convex decomposition, Computer-Aided Design, 24(9), 1992, 461-476. https://doi.org/10.1016/0010-4485(92)90027-8
[12] Koenderink, J. J.; van Doorn, A. J.: Surface shape and curvature scales, Image and Vision Computing, 10(8), 1992, 557-564. https://doi.org/10.1016/0262-8856(92)90076-F
[13] Lai, J. Y.; Wang, M. H.; Song, P. P.; Hsu, C. H.; Tsai, Y. C.: Recognition and decomposition of rib features in thin-shell plastic parts for finite element analysis, Computer-Aided Design and Applications, 15(2), 2018, 264-279. https://doi.org/10.1080/16864360.2017.1375678
[14] Lambourne, J. G.; Willis, K. D. D.; Jayaraman, P. K.; Sanghi, A.; Meltzer, P.; Shayani, H.: BRepNet: A topological message passing system for solid models, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, 12768-12777. https://doi.org/10.1109/CVPR46437.2021.01258
[15] Li, K.; Foucault, G.; Léon, J. C.; Trlin M.: Fast global and partial reflective symmetry analyses using boundary surfaces of mechanical components, Computer-Aided Design, 53, 2014, 70-89. https://doi.org/10.1016/j.cad.2014.03.005
[16] Lu, Y.; Gadh, R.; Tautges, T.J.: Feature based hex meshing methodology: feature recognition and volume decomposition, Computer-Aided Design, 33(3), 2001, 221-232. https://doi.org/10.1016/S0010-4485(00)00122-6
[17] Onodera, M.; Hariya, M.; Kongo, C.; Shintani, M.; Ka, K.; Watanuki, K.: Shell mesh generation technique reusing proven models by similar sub-part search, Journal of Advanced Mechanical Design, Systems, and Manufacturing, 13(4), 2019, JAMDSM0071. https://doi.org/10.1299/jamdsm.2019jamdsm0071
[18] Open Cascade Technology, https://dev.opencascade.org/doc/overview/html/index.html.
[19] Pernot, J. P.; Falcidieno, B.; Giannini, F.; Léon, J. C.: Incorporating free-form features in aesthetic and engineering product design: state-of-the-art report, Computers in Industry, 59(6), 2008, 626-637. https://doi.org/10.1016/j.compind.2008.03.004
[20] Prabhakar, S.; Henderson, M. R.: Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models, Computer-Aided Design, 24(7), 1992, 381-393. https://doi.org/10.1016/0010-4485(92)90064-H
[21] Qi, C. R.; Su, H.; Mo, K.; Guibas, L. J.: PointNet: deep learning on point sets for 3D classification and segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, 77-85. https://doi.org/10.1109/CVPR.2017.16
[22] Qi, C. R.; Yi, L.; Su, H.; Guibas, L. J.: PointNet++: deep hierarchical feature learning on point sets in a metric space, Advances in Neural Information Processing Systems, 30, 2017. https://doi.org/10.48550/arXiv.1706.02413
[23] Sakurai, H.: Volume decomposition and feature recognition: part 1—polyhedral objects, Computer-Aided Design, 27(11), 1995, 833-843. https://doi.org/10.1016/0010-4485(95)00007-0
[24] Shi, P.; Qi, Q.; Qin, Y.; Scott, P. J.; Jiang, X.: A novel learning-based feature recognition method using multiple sectional view representation, Journal of Intelligent Manufacturing, 31, 2020, 1291-1309. https://doi.org/10.1007/s10845-020-01533-w
[25] Shi, Y.; Zhang, Y.; Baek, S.; Backer, W. D.; Harik, R.: Manufacturability analysis for additive manufacturing using a novel feature recognition technique, Computer-Aided Design and Applications, 15(6), 2018, 941-952. https://doi.org/10.1080/16864360.2018.1462574
[26] Takashima, H.; Kanai, S.: Recognition of free-form features for finite element meshing using deep learning, Computer-Aided Design and Application, 19(4), 2022, 677-693. https://doi.org/10.14733/cadaps.2022.677-693
[27] Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y.: Graph attention networks., arXiv, 2017. https://doi.org/10.48550/arXiv.1710.10903

[28] Wu, H.; Gao, S.: Automatic swept volume decomposition based on sweep directions extraction for hexahedral meshing, Procedia Engineering, 82, 2014, 136-148. https://doi.org/10.1016/j.proeng.2014.10.379
[29] Zhang, Z.; Jaiswal, P.; Rai, R.: FeatureNet: machining feature recognition based on 3D convolution neural network, Computer-Aided Design, 101, 2018, 12-22. https://doi.org/10.1016/j.cad.2018.03.006