

Article

Hierarchical Graph Neural Network for Manufacturability Analysis

Xiuling Li ¹, Bo Huang ^{2,3,*}, Xuewu Li ¹, Fusheng Li ¹, Peng Wang ⁴ and Shusheng Zhang ⁵¹ School of Mechanical and Electrical Engineering, Zheng Zhou Railway Vocational & Technical College, Zhengzhou 450052, China; 10897@zzrvtc.edu.cn (X.L.)² Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China³ Shenzhen Key Laboratory of Precision Engineering, Shenzhen 518055, China⁴ College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China⁵ School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China

* Correspondence: bo.huang@siat.ac.cn

Abstract

Problems such as unreasonable processability or model defects generated in the design stage will lead to continuous rework during the manufacturing process, which greatly increases the manufacturing cost of the product. Through manufacturability analysis, the process designer can find design defects that are difficult to manufacture, impossible to manufacture, or have high manufacturing costs as early as possible, so as to reduce the number of round trips between design and process, and shorten the product development cycle. However, it is difficult for the current rule-based manufacturability analysis method to obtain professional knowledge and construct a complete manufacturability analysis rule repository. Therefore, a manufacturability analysis method based on a graph neural network is proposed. First, the attribute adjacency graph and UV gridding are combined to characterize the part model data, which can effectively represent the topological information and geometric information on the part model. At the same time, parameter information on the spherical coordinate system is used to ensure rotation and translation invariance; then, based on the graph representation of the part model, a hierarchical graph neural network is constructed, which is divided into three levels, edge, node, and graph, for encoding, information transmission and updating, and expanding the receptive field for better node classification to support manufacturability analysis. Finally, graph contrastive learning is used as a regularization technique in the pre-training stage to maximize the similarity of graph representations between different views to improve prediction performance. Manufacturability analysis tests were carried out on the constructed part model dataset, and the experimental results showed that the method has good performance.

Keywords: manufacturability analysis; attribute adjacency graph; manufacturing semantic; graph neural network; graph contrastive learning



Academic Editors: Panagiotis Kyratsis and Pierre Vella

Received: 19 October 2025

Revised: 21 November 2025

Accepted: 24 November 2025

Published: 26 November 2025

Citation: Li, X.; Huang, B.; Li, X.; Li, F.; Wang, P.; Zhang, S. Hierarchical Graph Neural Network for Manufacturability Analysis. *Machines* **2025**, *13*, 1091. <https://doi.org/10.3390/machines13121091>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous iteration and improvement of new products and the increasing market competition, manufacturing companies have also put forward higher requirements for the R&D cycle of new products and the manufacturability and economy of products. At the same time, with the widespread application of 3D CAD technology and MBD technology, companies often face the following problems in the product design process:

(1) Product design problems are difficult to find, and manual review is time-consuming and labor-intensive: when facing large and complex parts, it is usually difficult for humans to judge the subtle process design problems, which can easily lead to omissions. If they are checked one by one, it can take a lot of time and manpower and is prone to errors. (2) Professional knowledge is difficult to obtain, and manufacturability analysis results rely on professionals: when there is no complete manufacturability analysis rule repository, only discrete rule management, this requires designers to query manuals, design specifications, etc. In this way, review efficiency is very low, and personnel turnover also leads to the loss of corporate knowledge. (3) Manufacturability review lacks a unified standard, and the communication cost is high: under the manual review method, there is a lack of a common language, specification description, and management method between the structural design stage and the process design stage in the enterprise, resulting in low review communication efficiency and high labor costs. It is understood that 80% of product defects are introduced during the design phase, and the cost increase caused by rework accounts for about 45% of the product cost. Therefore, it is imperative to develop an effective manufacturability analysis tool, which will greatly enhance the competitiveness of enterprises. The tool can automatically inspect and analyze product design models, helping structural designers, process designers, and manufacturing engineers to find design defects that are difficult to manufacture, impossible to manufacture, or have high manufacturing costs as early as possible, thereby effectively reducing manufacturability analysis time, design and process rework times, and thus shortening product development cycles.

At present, rule-based manufacturability analysis systems are also being deployed and applied by enterprises. As shown in Figure 1, the idea is to construct a manufacturability analysis rule repository based on standards formulated at all levels, such as national standards, industry standards, or enterprise standards, as well as the process experience accumulated by structural designers, process designers, and manufacturing engineers. The system performs manufacturability analysis on the designed 3D model through rule reasoning and generates an analysis report. Structural designers modify the design model based on the feedback analysis report [1,2]. The system has achieved good application results in the case of product type determination. However, products are currently being diversified, which will require a more complete rule repository. However, it is difficult to construct a complete rule repository, and it results in a heavy reliance on experienced process designers. Therefore, this method still cannot solve the problems existing in the manufacturability analysis process of the above-mentioned enterprises. The data-driven method achieves various tasks by mining implicit knowledge in the data, reducing the reliance on professionals, and having better flexibility. At present, the application of deep learning to the design or manufacturing field is becoming popular, such as part classification [3], engineering drawing recognition [4], machining feature recognition [5], machining parameter prediction [6], machining process planning [7], manufacturing cost estimation, etc., and has achieved good results. However, the application of deep learning in the field of manufacturability analysis is still less studied. It is mainly aimed at simple parts with single features or general databases, which is difficult to apply in practice. At the same time, most methods use voxels or point clouds to represent 3D geometry, which result in a large amount of data processing and easily lose the topological information of the parts and the accuracy requirement information, etc., resulting in a poor application effect when dealing with complex parts.

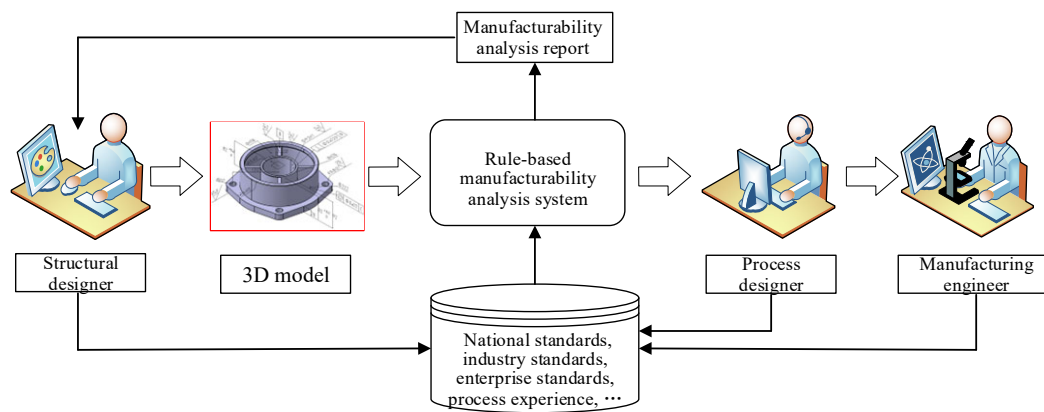


Figure 1. The rule-based manufacturability analysis approach.

Based on the above background and requirements, a manufacturability analysis method based on graph neural networks is proposed. First, the attribute adjacency graph and UV gridding are combined to characterize the part model data, which can effectively represent the topological and geometric information on the part model. At the same time, the parameter information on the spherical coordinate system is used to ensure rotation and translation invariance; then, based on the graph representation of the part model, a hierarchical graph neural network is constructed, which is divided into three levels, edge, node, and graph, for encoding, information transmission, and updating. The receptive field is expanded for better node classification to support manufacturability analysis. Finally, in the pre-training stage, graph contrastive learning is used as a regularization technique to maximize the similarity of graph representations between different views and improve prediction performance. The effectiveness of the proposed method is verified through experimental tests.

2. Related Work

Part manufacturability analysis and deep learning applications in manufacturing are the key issues of the proposed work. We will provide a brief review of each part, respectively.

2.1. Part Manufacturability Analysis

Part manufacturability analysis is an important link between product design and manufacturing, and is the basis for subsequent applications such as machining process planning and NC programming. At present, research work is mainly divided into two categories: model defect detection and manufacturability analysis. Model defect detection mainly analyzes the errors or defects generated in the part model during the design stage, and provides accurate models for NC machining process design or NC programming. Manufacturability analysis aims to identify whether there are structures that are difficult to manufacture, impossible to manufacture, or have high manufacturing costs in the design results, providing an effective tool for DFM (Design for Manufacturing).

In terms of model defect detection, Huang et al. [8] analyzed common model defects for specific parts, such as aircraft structural parts, constructed seven basic heuristic rules based on the geometric characteristics of the defects, and detected geometric defects on the part model through rule reasoning. Yang et al. [9] developed a knowledge-based part model quality analysis system for MBD models, which constructed a knowledge representation framework through object chains and parameter tables, and introduced model definition units to convert model information into model semantic quality to support model defect detection.

In terms of manufacturability analysis, Bidkar et al. [10] developed a manufacturability analysis method for injection molding and die casting parts. Kerbrat et al. [11] used an OC tree to decompose the 3D model and used it to evaluate the manufacturability of the parts by subtractive or additive manufacturing. Xu et al. [12] analyzed the design model through machining feature recognition and converted the design model to a structuralized feature model. The knowledge rules were detailed by meta-model expression, and finally, the manufacturability analysis was automated through knowledge reasoning. Matuszek et al. [13] designed a fuzzy logic-based method to evaluate the manufacturability of new products in small-lot and unit production. Shi et al. [14] used the Heat Kernel Signature method to identify geometric features and manufacturing constraints, and then conducted a comprehensive analysis of the design model from the perspective of manufacturability. Yeo et al. [15] summarized 19 common criteria in turning, milling, and drilling processes and extracted the information required for evaluation by identifying the machining features of the parts. Finally, the summarized criteria were used to evaluate the manufacturability of the parts. Chen et al. [16] represented the design model as a polygonal mesh, sampled the 3D tool geometry as line segments, and calculated the geometric accessibility of the design model as the basis for the manufacturability evaluation of five-axis milling. Mayerhofer et al. [17] developed a manufacturability analysis system for additive manufacturing parts, which used ontology to construct a knowledge repository and used triangle-based mesh processing algorithms to recognize features to support the manufacturability analysis of additive manufacturing.

Whether it is model defect detection or manufacturability analysis, both are, in essence, rule-based methods. The basic idea is to first construct a rule repository, then analyze and characterize the design model, and finally judge the rationality of the design model through rule reasoning. This type of method can achieve good application results for specific types of parts, but it requires a large amount of work to construct heuristic rules and relies heavily on experienced designers.

2.2. Deep Learning Applications in Manufacturing

Compared with traditional machine learning methods, such as support vector machine and other shallow neural networks, deep learning has better generalization ability and learning effect. After more than ten years of development, deep learning technology has made significant progress and breakthroughs in various fields, and its applications have become more mature. In the field of product design and manufacturing, deep learning methods have also been applied in manufacturing application scenarios, such as machining feature recognition, manufacturing cost estimation, and machining process planning, showing good application potential.

Zhang et al. [18] proposed a FeatureNet framework that segments the 3D model into voxel units and uses a deep 3D convolutional neural network to learn machining features. Lei et al. [19] represented the 3D model in point cloud data format and designed a point cloud-based deep neural network based on selective downsampling to recognize manufacturing features, reducing the complexity of point cloud computing. Ning et al. [20] proposed using a 3D convolutional neural network to train the voxel data representing the part model to obtain the mapping relationship, thereby realizing manufacturing cost estimation. Yoo et al. [21] designed a deep learning framework that incorporates 3D gradient-weighted class activation mapping, which can recognize and visualize the machining features of the 3D CAD model that are influencing the increase in manufacturing costs, thereby improving the interpretability of the prediction process. Zhao et al. [22] designed a process planning framework based on deep learning. In the first stage, the input model to be manufactured is represented as an attribute adjacency graph, and a graph

neural network is used for feature recognition. In the second stage, the divided machining features are voxelized and a convolutional neural network is used for process planning. In terms of manufacturability analysis, Balu et al. [23] proposed using a 3D-CNN based on voxel data to analyze the manufacturability of parts. This method uses voxels to represent the 3D model, and augments the voxel data with surface normals of the object boundary. Finally, a 3D-CNN is trained with this augmented data and the local critical features are identified for decision-making using 3D gradient-weighted class activation maps. Ghadai et al. [24] proposed adding the surface normal information of the voxel to the voxel data and using its data to train the 3D-CNN. Peddireddy et al. [25] processed the voxel data of the part model through a 3D CNN network to classify the manufacturability of the part (e.g., milling, turning).

Since the method of representing 3D geometry with point clouds or voxels is prone to losing the topological information and precision requirements of parts, the use of graphs to represent data has received increasing attention in the field of machining processes. Wu et al. [26] designed a new graph neural network AAGNet to perform semantic segmentation on the adjacency graph of geometric attributes representing parts to achieve automatic feature recognition. Zhang et al. [27] proposed applying deep reinforcement learning algorithms to feature recognition, especially when dealing with difficult-to-recognize intersecting machining features. Jing et al. [28] combined graph embedding algorithms with graph neural networks to achieve manufacturing knowledge recommendations to help designers understand the manufacturability of parts. Xie et al. [29] proposed a graph neural network-based engineering drawing classification method, which vectorized engineering drawings and used a sequential graph neural network to detect and eliminate dimension lines to achieve classification, so as to support the query of part manufacturing methods. Deng et al. [30] constructed an attribute adjacency graph of machining parameters and geometry for the machining process data of aircraft structural parts, and determined the relationship between geometry and machining parameters through graph neural networks to support the decision-making of machining parameters. Zhang et al. [31] proposed a manufacturing cost estimation method based on a convolutional graph neural network, which incorporates accuracy information into geometry for data processing and develops spectral-based and spatial-based convolutional layer training learning data to achieve manufacturing cost estimation.

Inspired by the successful application of these studies, our method is based on graph data structures, which can effectively retain topological information and integrate geometric and semantic information. In this way, key topological connections and hierarchical information can be preserved without loss, providing more accurate inputs for the model. At the same time, graph neural networks are introduced for manufacturability analysis to identify design defects that affect programming on parts, and structures that are difficult to manufacture or impossible to manufacture. Compared to rule-based methods, it has good adaptability and can expand the recognition range through simple training. At the same time, it has better interpretability and can provide geometric and topological information of unreasonable structures while judging the results, helping designers make reasonable modifications and effectively supporting subsequent manufacturing processes.

3. Overview of the Approach

The rationality and standardization of manufacturability analysis can effectively avoid design defects, rework, and other issues, thereby significantly reducing costs and enhancing the production and development efficiency of products. The traditional method of manufacturability analysis relies on constructing heuristic rules to make judgments, and the completeness of the rules greatly affects the results of the analysis. Compared to the

data representation of point clouds and voxels, graph representation can preserve the topological information of part structure and is more suitable as a data processing format for part manufacturability analysis. Therefore, a manufacturability analysis method based on graph neural networks is proposed. The geometric and topological information of part design data is characterized through attribute adjacency graphs, and a hierarchical graph neural network is designed to obtain information at different levels to support more accurate manufacturability analysis. At the same time, graph contrastive learning is introduced for data augmentation to alleviate the problem of deep learning methods requiring a large amount of labeled data.

Figure 2 shows the general framework of our approach. It can be seen that our approach contains three parts: data processing, hierarchical graph neural network, and graph contrastive learning.

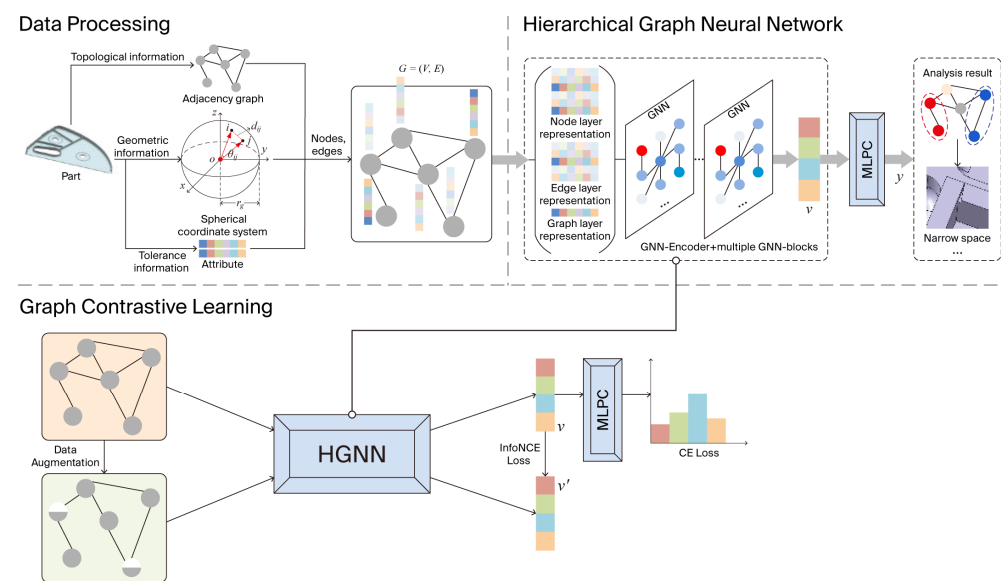


Figure 2. General framework of our approach.

- (1) **Data processing:** For design parts that require manufacturability analysis, they are first converted into the form of attribute adjacency graphs to characterize the topological relationship between geometric elements in the design structure. Meanwhile, a UV grid is used to characterize the geometric information of the design structure, and based on this, the geometric information of the geometric elements themselves and the geometric relationship between the geometric elements are extracted. In addition, by extracting and characterizing tolerance information, the high-level information of the overall geometric representation is effectively supplemented.
- (2) **Hierarchical graph neural network:** After constructing a graph representation for the design model, a hierarchical graph neural network is designed to embed the graph into a fixed-size graph-level latent representation for node classification to support manufacturability analysis. The graph neural network is mainly divided into three modules: (a) a Graph Neural Network Encoder (GNNEncoder), which is designed to encode the original edge and node feature vector sets into high-level representations and calculate the graph feature vector from the encoded node feature vector set; (b) a Graph Neural Network Block (GNN block) based on a Gate Recurrent Unit, which is used to expand the range of the receptive field and hierarchically update the latent feature vectors of edges, nodes and graphs; and (c) a Multilayer Perceptron Classifier (MLPC), which is applied to classify the node feature vectors represented above, so as to achieve the purpose of manufacturability analysis.

- (3) **Graph contrastive learning:** Graph contrastive learning is used as a regularization technique to maximize the similarity of graph representations between different views to improve prediction performance. Different views are provided by partial edge removal and node attribute masking, and then the model is trained to learn the embedding of the occluded part and a linear model is applied to the embedding to reconstruct the node/edge attributes. The model parameters are optimized using the cross entropy loss for classification and the contrast loss for different views.

4. Hierarchical Graph Neural Network for Manufacturability Analysis

4.1. Data Processing

The design model to be analyzed is the basic carrier for subsequent process planning, assembly, functional application, etc., so its manufacturability, assemblability, functionality, etc., are considered in the design stage. It is based on the B-rep data structure and contains topological information, geometric structure, manufacturing semantics, and other information. Among them, points, lines, and faces are the basic components of geometric bodies. Geometric bodies can be directly constructed by face sets and the connection relationship between faces. The connection relationship between faces constructed by shared edges is the topological information of the design model. The geometric structure of the design model can be reflected by the geometric shapes of faces and shared edges and the geometric relationships between these elements, such as the curvature of the face, the length of the edge, the angle between adjacent faces, etc. Manufacturing semantics is high-level information that reflects the design intent. The core of it is the tolerance information marked on the design model. There may be problems of process rationality in the tolerance itself, between tolerances, and between tolerances and associated geometry. Although expressive, the design part model is a complex data structure and is difficult to feed to neural networks in its original form. Our goal is to extract the most informative geometric and topological information from the design model, and convert it into a representation that can easily and efficiently work alongside existing neural network architectures.

4.1.1. Topological Information

The topological information of the design model can be effectively represented by constructing the attribute adjacency graph of faces and edges. The attribute adjacency graph is generalized from the concept of a graph and is a graph structure that is very suitable for describing the geometry and topological information of the part model. It is expressed as $G = \langle V, E, H_V, H_E \rangle$, where V represents a non-empty set of nodes. For a certain node V_i in set V , there is a unique face f_i corresponding to it in the part model; E is a set of edges, which represents the adjacency relationship between faces. For any two faces f_i and f_j in the part model, if there is an adjacency relationship, there is a unique edge E_i corresponding to them in set E ; H_V represents the attribute set of the face in node V . The attributes of the face in the part model include the type of face, the concavity of the face, the area of the face, etc.; H_E represents the attribute set of the edge E . The attributes of the edge in the part model include the type of edge, the concavity of the edge, the length of the edge, etc. The attributes here mainly refer to the attributes of geometric elements such as faces and edges.

In 3D design software such as CATIA and UG, the properties of each face and edge in the part model can be obtained through its underlying functions. The property table of faces and edges is shown in Table 1.

Table 1. Geometric attributes.

ID	Attribute Type	Attribute Representation	Attribute Description
Node V	Geometry Type	GT_v	Plane, cylindrical surface, conical surface, etc.
	Concavity	Cov_v	Concave, convex, flat
	Area	$Area_v$	Area of face
Edge E	Type	GT_e	Line edge, arc edge, etc.
	Concavity	Cov_e	Concave, convex, tangent
	Length	L_e	Length of edge

4.1.2. Geometric Information

The attribute adjacency graph can effectively characterize the topological information, while the geometric information can only be briefly characterized by attributes. However, geometric information is crucial in three-dimensional structures and directly affects the process rationality of parts. In order to represent geometric information, the current common practice is to train by directly obtaining Cartesian coordinate information. Obviously, the directly generated Cartesian coordinates cannot satisfy the rotation or translation invariance. In our method, the distance, angle, dihedral angle, and other information are obtained through the spherical coordinate system to represent the geometric information, which themselves satisfy the rotation and translation invariance. In terms of the overall geometric representation, in addition to the aforementioned attributes, the distance from the center of mass of the 3D geometric body to the center of the face is used as the attribute of the node in the graph, and the angle between the center of mass and the line connecting the two face centers is used as the attribute of the edge. These two attributes serve as supplements to the overall geometric representation.

After the overall geometric representation, the relative position of the face in the geometric body is determined, and it is also necessary to represent the geometry of the face itself and the geometric relationship between the faces in order to fully represent the geometric information. The faces in the 3D model are divided into geometric faces and topological faces. The topological faces represent the connection relationship between geometric elements, and the geometric faces contain geometric information. Among them, the geometric face can be further represented by the U_V grid, that is, the discrete parameterized representation. For example, the geometric face $F(u, v)$ can be represented by the parameter domain $[u_{min}, u_{max}] \times [v_{min}, v_{max}]$. Furthermore, the 3D geometric face can be represented in the 2D parameter domain through 2D sampling with a step size of $s = (u_{max} - u_{min}) / (M - 1)$, where M is the number of samples in a single direction. The parameter domain can completely surround the face and divide it into a visible region and a trimmed region. The visible area is inside the entity, and the trimmed area is the part outside the entity within the parameter domain. Thus, at each grid point (k, l) , we attach the following local features encoding the geometry of the surface as channels: (a) Distance, angle, and dihedral, representing the geometric position. The encoding requires normalizing the geometry into a sphere with a radius of 1, with the center of the sphere as the center of mass and the U_V direction as the base direction, and then calculating each value, as shown in Figure 3. (b) Maximum curvature and minimum curvature, representing the geometric shape. (c) A flag bit with a value of 1 or 0, indicating whether the sampling point is in the visible region or the trimmed region.

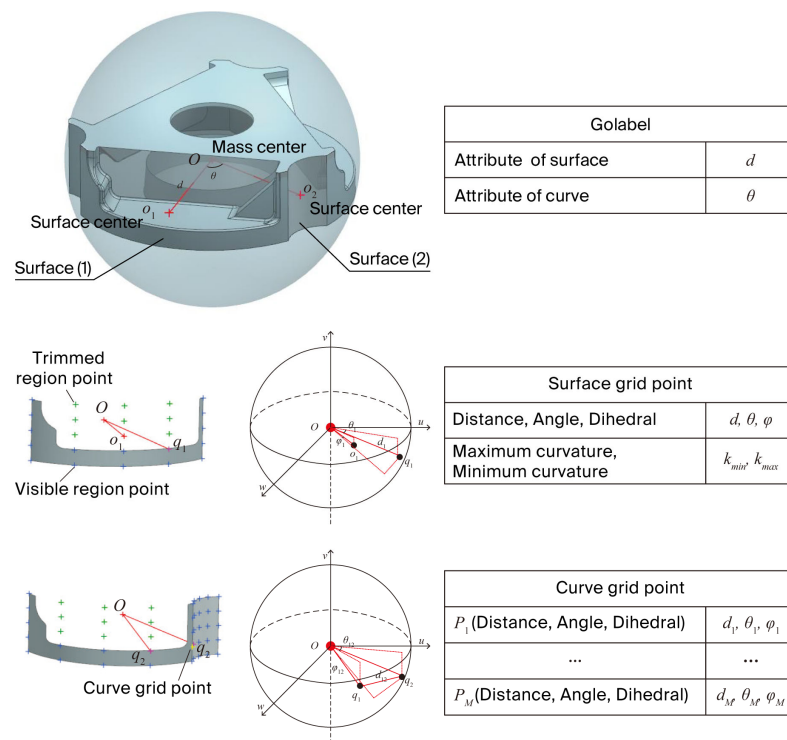


Figure 3. Geometric information.

The basic attributes of the edge have been expressed above. For the connecting edge of two faces, it is more important to express the spatial relationship between the two faces. Similarly, the UV grid can also sample and parameterize the edges. Inspired by the D_2 shape descriptor, the spatial relationship between the two faces is also characterized by the distance between the sampling points on the two faces, and the angle and dihedral angle are calculated to further enrich the expression. The sampling points on the edge are used as guide points, based on which the sampling points on the face with the same sampling step are obtained, such as q_1 and q_2 in Figure 3. In this way, each point on the edge of the UV grid has M groups of attributes, each with three parameters.

The input encoding of geometric information uses a 2D CNN and a 1D CNN, which are used for the face encoder and edge encoder, respectively. The face UV grid represented in 5D form is input into the face encoder, and a 64D vector is output to represent the extracted face feature information. The edge input dimension is related to the number of samples, and $M = 8$ is used in our method. In this way, the edge UV grid represented in 24D form is input into the edge encoder, and a 64D vector is also output to represent the feature information of the edge.

4.1.3. Tolerance Information

Tolerance information mainly refers to accuracy requirement information, that is, the tolerance requirements marked, which include dimensional tolerance, form and position tolerance, and roughness. From another perspective, accuracy requirement information is divided into those associated only with the machining surface itself and those associated with multiple machining surfaces. The marking of tolerance requirements is related to geometry, and the geometry of the machining surface (such as size) affects machinability and economy. In addition, there are also relationships between tolerances, such as that the geometric tolerance grade is higher than the dimensional tolerance grade. In the MBD model, tolerance marking is associated with low-level geometric elements such as points, lines, and surfaces. In order to better represent the data, it is necessary to further associate

it with the two-level elements of face and edge. At the same time, when there are the tolerances with datum or other tolerances associated with multiple machining surfaces, this type of association relationship does not have pure geometric information, but expresses high-level design intent. In order to better express this type of association relationship, the concept of semantic edges is introduced, and the original attribute adjacency graph constructed based on geometric elements is optimized by adding semantic edges. As shown in Figure 4, there was no edge between f_1 and f_2 . Since the machining surface f_1 is marked with coaxiality based on f_2 , the originally independent surfaces f_1 and f_2 are directly connected by constructing semantic edges. There are dimensional tolerances between surfaces f_3, f_4, f_5 and surfaces f_6, f_7 , and a closer relationship between these surfaces is constructed through semantic edges. Similarly, in 3D design software such as CATIA and UG, the tolerance annotation information in the MBD model can be obtained through its underlying functions. The specific tolerance information is shown in Table 2.

Table 2. Tolerance information.

ID	Attribute Type	Attribute Representation	Attribute Description
Node V	Roughness	Ra	Eight commonly used surface roughness values: 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.3, 12.5
	Form and position tolerance	FT	Form and position tolerance is associated with a single machining surface, and mainly includes tolerance types such as straightness, flatness, roundness, cylindricity, line profile, and surface profile. The tolerance values are divided into 12 levels.
	Dimensional tolerance	DT_v	This type of dimensional tolerance is only related to the machining surface itself, and its tolerance value is divided into 18 levels.
Edge E	Type	T_e	Geometric edge, semantic edge
	Position tolerance	PT	Position tolerance is associated with multiple machining surfaces, and mainly includes tolerance types such as line profile, surface profile, parallelism, perpendicularity, inclination, position, coaxiality, symmetry, circular runout, and total runout. The tolerance values are divided into 12 levels.
	Dimensional tolerance	DT_e	This type of dimensional tolerance is related to multiple machining surfaces, and its tolerance values are divided into 18 levels.

The attributes of geometric elements and precision requirements can be attached to nodes or edges, namely face attributes and edge attributes, and two MLP models are used as attribute encoders for their inputs. The input dimensions of faces and edges are both 8D, and form and position tolerance are 2D, which are type and value, respectively. Both input vectors are converted into 64D feature embeddings by the MLP model, which includes linear functions, normalization, and activation functions. The weights of the face encoders, edge encoders, and attribute encoders of all nodes and edges in the graph are shared to ensure permutation invariance.

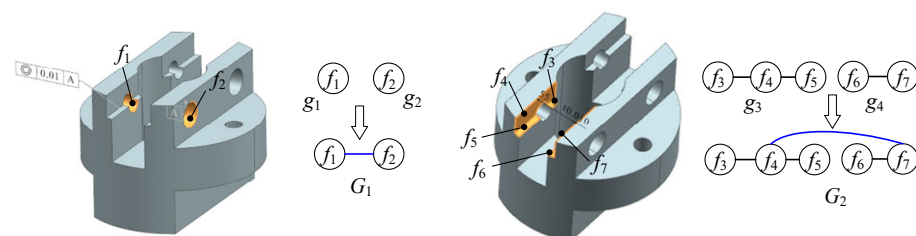


Figure 4. Semantic edges.

4.2. Hierarchical Graph Neural Network

After representing the geometry, topology, precision requirements, and other information of the design model, a hierarchical graph neural network is proposed to embed the graph structure data to support subsequent manufacturability analysis tasks. It mainly

consists of three parts: (1) a GNN encoder; (2) GNN blocks; and (3) a multilayer perceptron classifier (MLPC).

4.2.1. GNN Encoder

The GNN encoder encodes the nodes and edges in the attribute adjacency graph into high-dimensional representations of nodes, edges, and graphs. First, the encoded edge feature vector is obtained by encoding the original edge e'_{ij} and the corresponding nodes v_i and v_j . The specific calculation is as follows:

$$e'_{ij} = \text{MLP}_{\text{encoder}}^e([e_{ij}, v_i, v_j]) \quad (1)$$

where $\text{MLP}_{\text{encoder}}^e$ is an MLP operation that performs nonlinear transformation, and $[e_{ij}, v_i, v_j]$ represents the concatenation of e_{ij} , v_i , and v_j . Then, the encoding node feature vector v'_i is obtained by using the original node feature vector v_i and the adjacent edge feature vector updated in the previous step. The specific calculation is as follows:

$$v'_i = \text{MLP}_{\text{encoder}}^v([v_i, \sum_{j \in N(v_i)} e'_{ij}]) \quad (2)$$

where $N(v_i)$ is the neighbor set of node i , and similarly, $\text{MLP}_{\text{encoder}}^v$ is an MLP operation that performs nonlinear transformation.

Finally, the feature vectors of all encoded nodes in the graph are used to obtain the feature vector g' of the graph. The specific calculation is as follows:

$$g'_i = \text{MLP}_{\text{encoder}}^g(\sum_{i=1}^{N_v} v'_i) \quad (3)$$

4.2.2. GNN Blocks

In order to expand the receptive field, our method stacks multiple GNN blocks. In this way, distant edges or nodes can influence each other, thus achieving a stable representation. Each GNN block updates the edge, node, and graph feature vector in turn, as shown in Figure 5.

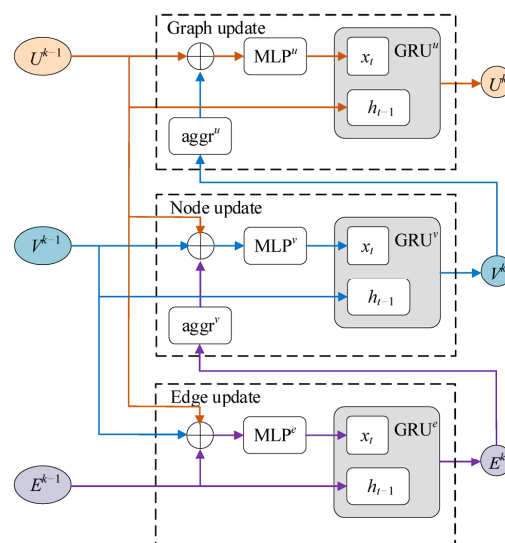


Figure 5. GNN blocks.

First, the edge feature vector e_{ij}^k of layer k is calculated using the edge feature vector e_{ij}^{k-1} of layer $k-1$, two node feature vectors v_i^{k-1} and v_j^{k-1} , and the graph feature vector g^{k-1}

as input. The input is fed into the nonlinear transformation MLP^e to obtain the intermediate output $e_{ij}^{k'}$, and further GRU^e is used to perform nonlinear weighted transformation. In this way, the updated edge feature vector e_{ij}^k of layer k can be calculated as follows:

$$e_{ij}^{k'} = MLP^e([e_{ij}^{k-1}, v_i^{k-1}, v_j^{k-1}, g^{k-1}]) \quad (4)$$

$$e_{ij}^k = GRU^e(e_{ij}^{k'}, e_{ij}^{k-1}) \quad (5)$$

Next, the updated feature vectors of all adjacent edges of node i are aggregated into its neighbor edge feature vector. The intermediate output $v_i^{k'}$ is obtained by performing nonlinear transformation MLP^v on all adjacent edge feature vectors, the previous layer node feature vectors, and the previous layer graph feature vectors. Furthermore, GRU^v weights $v_i^{k'}$ and v_i^{k-1} to obtain the updated node feature vector, which is calculated as follows:

$$v_i^{k'} = MLP^v\left(v_i^{k-1}, \sum_{j \in N(v_i)} e_{ij}^k, g^{k-1}\right) \quad (6)$$

$$v_i^k = GRU^v(v_i^{k'}, v_i^{k-1}) \quad (7)$$

Finally, all node feature vectors and the previous layer graph feature vector are concatenated as input, and the intermediate graph feature vector $g^{k'}$ is calculated through nonlinear transformation MLP^g . Then, GRU^g is used to obtain the updated graph feature vector g^k , which is calculated as follows:

$$g^{k'} = MLP^g\left(\sum_{i=1}^{N_v} v_i^k, g^{k-1}\right) \quad (8)$$

$$g^k = GRU^g(g^{k'}, g^{k-1}) \quad (9)$$

4.2.3. Multilayer Perceptron Classifier

Four stacked GNN blocks are used to obtain four node feature vectors, and these four vectors are concatenated as the final representation of the target. Furthermore, the concatenated feature vectors are input into a multilayer perceptual classifier to obtain the node classification probability:

$$\hat{\mathbf{y}} = \text{softmax}(W_2 \max(0, W_1[v^1, \dots, v^K] + b_1) + b_2) \quad (10)$$

4.3. Graph Contrastive Learning

Contrastive learning has achieved good results in graph data pre-training and unsupervised learning. The essence of using contrastive learning methods in graph structures is to generate two views of the same sample through data augmentation, and then maximize the similarity between their encoded representations, which provides great help for pre-training. As shown in Figure 6, the specific idea is as follows: First, based on the original graph, an enhanced graph is generated by removing some edges, masking some attributes of nodes, or injecting noise. The nodes in the original graph and the enhanced graph are encoded to obtain node embeddings; the model is trained with labeled data, and pseudo labels are predicted for unlabeled data; and pseudo labels are used to guide the construction of positive and negative samples within or between views; in this way, the cross entropy error obtained from the original graph and the contrast error obtained from the contrast process are combined to provide feedback and optimize the parameters of the model.

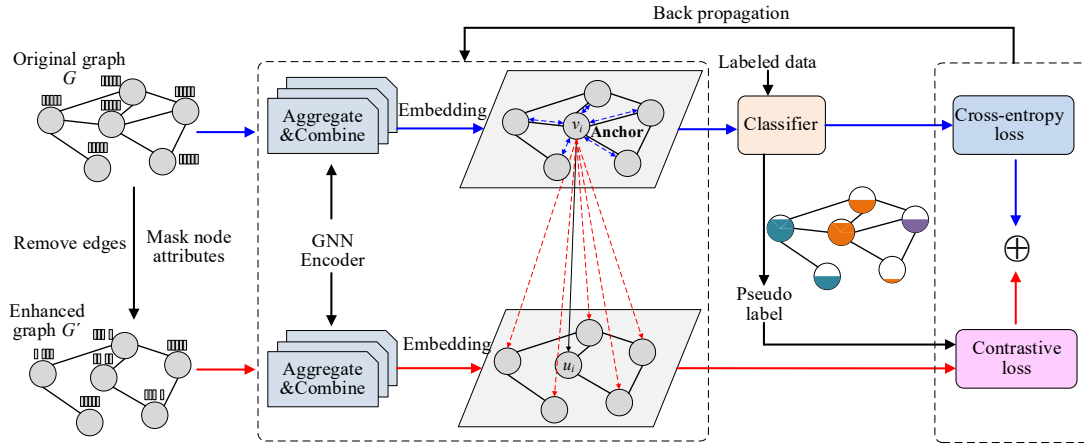


Figure 6. Graph contrastive learning.

In order to generate different views, data augmentation methods that remove some edges and mask some attributes of nodes are adopted to perturb the original graph from both structural and attribute aspects. In this way, node embeddings under different views are compared by constructing diverse node contexts. More specifically, the following method is used: (1) First, a mask matrix $\mathbf{R} \in \{0,1\}^{N \times N}$ is obtained through Bernoulli distribution, and then it is used for random sampling to remove edges to obtain the adjacent matrix $\mathbf{A}'$ of the new view. (2) Similarly, Bernoulli distribution is used to randomly mask some attributes in the node features to obtain a new node feature representation $\mathbf{v}'$. In this way, after these two steps, a new view G' can be obtained. The specific calculation is as follows:

$$\mathbf{A}' = \mathbf{A} \circ \mathbf{R} \quad (11)$$

$$\mathbf{v}' = [v_1 \circ m'; v_2 \circ m'; \dots; v_N \circ m']^T \quad (12)$$

The loss function is used to measure the deviation between the model's prediction and the actual result, and to guide the learning and optimization of the model. In our method, a combination of the contrast loss function and the classification loss function is adopted as the final loss function. In contrastive learning, the InfoNEC loss function is a commonly used contrastive loss function. For the sampling node, when the center sample is similar to the positive sample and has a large gap with the negative sample, the loss value should be small. On the contrary, when the center sample is far away from the positive sample and similar to the negative sample, the loss value should be large, which will prompt the model to update the parameters. The target for each positive sampling pair is defined as follows:

$$l(u_i, v_i) = -\log \frac{\exp(s(u_i \cdot v_i)/\tau)}{\exp(s(u_i \cdot v_i)/\tau) + \sum_{j \in S(i)} \exp(s(u_i \cdot v_j)/\tau)} \quad (13)$$

where $j \in S(i)$ represents the negative sample selected by node i , and $s(\cdot)$ represents the function for calculating the similarity between two samples. τ is the temperature coefficient used to control the distribution of similarity scores. Since the two views have a symmetric relationship, the loss function of the other view is defined as $l(v_i, u_i)$. Finally, the total contrast loss function can be defined as the average of all positive pairs:

$$L_{cl} = -\frac{1}{2N} \sum_{i=1}^N [l(u_i, v_i) + l(v_i, u_i)] \quad (14)$$

As for the classification function, after passing through the GNN encoder, the complete representation of each node $\mathbf{h} = (h_1, h_2, \dots, h_M)$ can be obtained, where M represents

the defect category in the manufacturability analysis. Then, the normalized exponential function softmax is used to obtain the score vector $\mathbf{p} = (p_1, p_2, \dots, p_M)$. The defect category with the maximum score is designated as the evaluation label. Manufacturability analysis is a multi-classification problem, and the loss function is defined as the commonly used cross entropy loss function. The calculation process is an extension of the binary classification, as follows:

$$L_{ce} = -\frac{1}{N} \sum_i \sum_{c=1}^M y_{ic} \log(p_{ic}) \quad (15)$$

where y_{ic} is the sign function (0 or 1), if the defect category of sample i is c , it takes 1, otherwise it takes 0; and p_{ic} is the predicted probability that the observed sample i belongs to defect category c .

In this way, the overall loss function L is obtained by combining the cross entropy loss function L_{CE} and the contrast loss function L_{CL} . The specific calculation is as follows:

$$L = L_{ce} + \lambda L_{cl} \quad (16)$$

where λ is a hyperparameter used to balance the two terms.

5. Experiments

The construction of the dataset in our method is based on the model data obtained by manufacturing enterprises, and the model data is expanded by adding features or changing features. As shown in Figure 7, a total of 5000 models are built in the dataset, of which 2000 are model data obtained and accumulated from manufacturing enterprises, and the rest are obtained by model expansion. The model extension method using feature addition can generate models with significant topological differences, which can effectively support research that requires high diversity of models. As shown in Figure 8, adding features to the input base model mainly requires two steps: (1) randomly selecting typical features from the typical feature template library; and (2) determining the positional and dimensional parameters of features. The model extension method using feature changes is similar to the feature addition method, but based on existing models. In this way, serialized models can be generated, supporting the construction of model data with different labels of the same type. Eighty percent of the models in the dataset will be used for training, and the remaining twenty percent will be used for testing.

The test was run on a workstation equipped with an Intel Core i9 CPU, 128GB of memory, and an NVIDIA GeForce RTX 3090 GPU. The proposed framework was developed using PyTorch1.12.0, a general-purpose deep learning framework, and the ADAM optimizer. We set the batch size to 256, and the learning rate used an initial value of 0.01.

5.1. Manufacturability Analysis

In order to evaluate the manufacturability analysis effects of different models, three indicators are selected: accuracy, average class accuracy, and Intersection over Union. The accuracy A is the ratio of correctly classified samples to the total number of samples, and the specific calculation is as follows:

$$A = \frac{\sum_{i=1}^M TP_i}{\sum_{i=1}^M (TP_i + FP_i)} \quad (17)$$

where M is the number of categories, TP_i is the number of samples correctly classified in the i -th category, and FP_i is the number of samples incorrectly classified in the i -th category.

The average class accuracy A_{pc} refers to the average of all class accuracy, which is calculated as follows:

$$A_{pc} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i} \quad (18)$$

The Intersection over Union (IoU) is the ratio of the intersection and union of the predicted results and the true results, which can quantify the overlap between the predicted results and the true results.

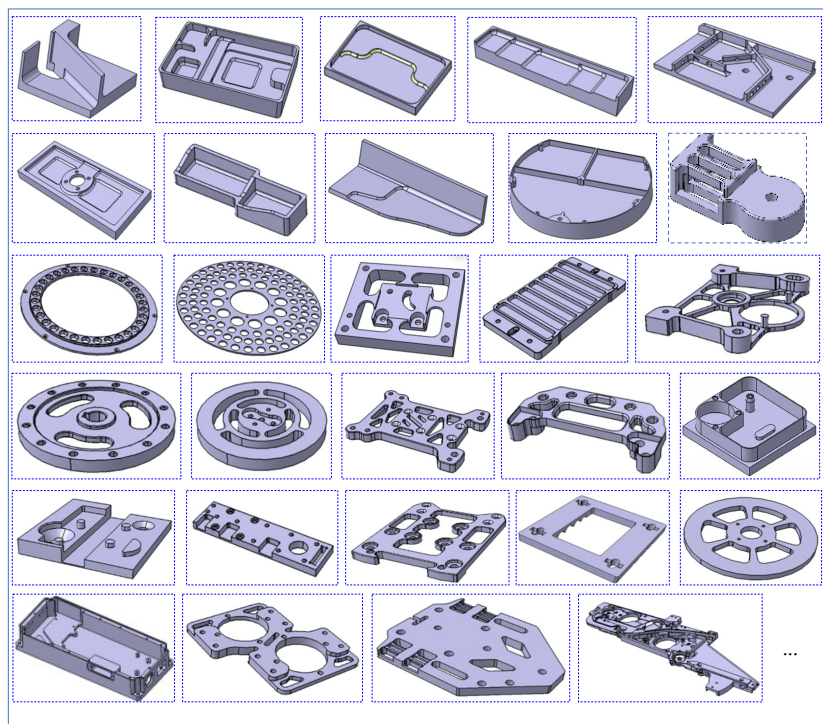


Figure 7. CAD models of experimental dataset (partial).

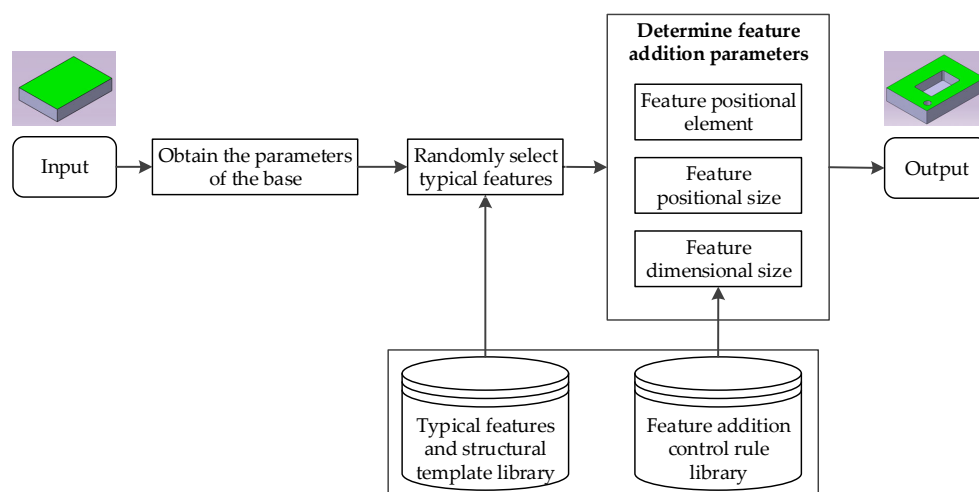


Figure 8. The model expansion method using feature addition.

The proposed method, PointNet, and the 3D-CNN calculate the manufacturability analysis results of each model. After training and adjustment, the accuracy curves and IoU curves of the three models in the verification phase on the part dataset constructed in our method are shown in Figure 9.

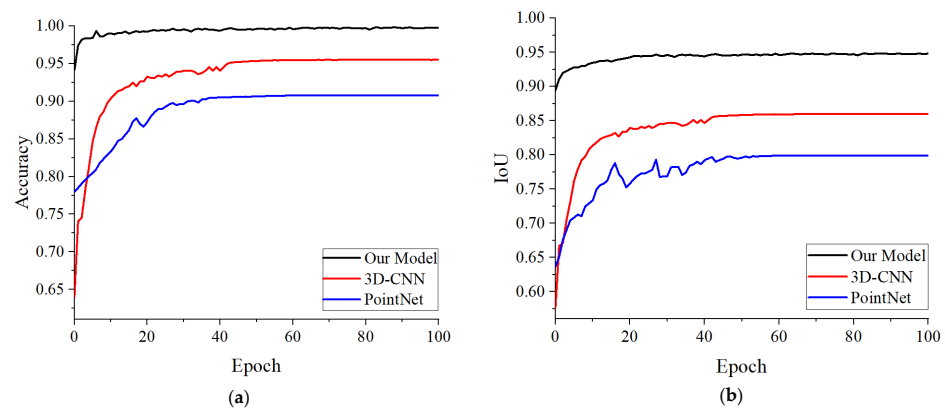


Figure 9. Accuracy and IoU curves of part model validation set. (a) Accuracy curve (Our model is represented in black, 3D-CNN in red, and PointNet in blue). (b) IoU curve (Our model is represented in black, 3D-CNN in red, and PointNet in blue).

Table 3 shows the comparison between the proposed method and other models. The 3D-CNN method uses voxel representation, and the amount of voxel data is much larger than the amount of data of the point cloud network and the graph structure. At the same time, the training cost of 3D convolution is high, and the training time of voxel data is twice that of point cloud data training. In addition, voxel representation of parts causes a large number of grids to be blank (the inside of the parts is actually blank), and it is difficult for the network to extract useful information by convolution. Point cloud data is extracted from the surface of the parts, and this type of data is relatively small. In terms of accuracy, the proposed method is 98.79%, which is higher than the 3D-CNN and PointNet methods. Among them, the 3D-CNN method based on voxel data has a good effect on the manufacturability analysis of simple parts. However, when the shape of the parts is more complex, voxel data has difficulty representing some detailed features, and only higher resolution voxelization can achieve better results. However, this increases the amount of data that needs to be processed geometrically. The proposed method is essentially based on the graph data structure and UV grid, combining macro and micro levels to obtain data. In this way, compared with the point cloud data of PointNet and the voxel data of the 3D-CNN, the method needs to process less data, and less useless data will be extracted. At the same time, the topological information retained by the graph data structure is useful to the manufacturability analysis, and the UV grid data also ensures sufficient resolution.

Table 3. Comparisons of our approach with existing approaches.

Method	Data Source	Data Format	Network Output	Accuracy (%)	Average Class Accuracy (%)	IoU (%)
3D-CNN	Simple hole feature	Voxel	Binary classification	95.56	85.56	86.61
PointNet	Simple hole feature	Point cloud	Multi-classification	90.85	74.92	79.90
Our approach	Enterprise model and 3D model extensions	Graph structure	Multi-classification	98.79	95.21	94.80

5.2. Ablation Experiment

In order to verify the influence of input features and various components in the manufacturability analysis network on the final analysis effect, four parameter states are set for comparison with the complete network, as follows: (1) Removing face attributes: this group of settings removes some face attribute features in the input features. (2) Removing edge attributes: this group of settings removes the adjacent edge attribute features in the

input features and only retains the node features. (3) Removing UV network structure: this group of settings removes the UV network part and reduces the geometric information of the micro level. (4) Removing the graph contrast learning part: this group of settings removes the graph contrast learning part and only uses the graph neural network to perform manufacturability analysis.

Using the above four settings, ablation experiments are carried out on the constructed part model dataset. The experimental results are shown in Table 4. Compared with the complete network model, the removing partial face attribute feature groups and the removing adjacent edge attribute feature groups showed a slight decrease in all three indicators, indicating that the contribution of adjacent edge features to the manufacturability analysis effect is more significant compared to face features. The three indicators of the model with the UV grid removed all decreased, indicating that the geometric information extracted from the UV grid can better capture model defects or unreasonable process structures. The decline in network model indicators is more significant when graph contrastive learning is removed. Based on the graph contrastive learning architecture, a data augmentation mechanism is introduced to prompt the model to pay more attention to local neighborhood information to enhance the central node features.

Table 4. Comparison of ablation experiment results with different parameters.

Model	A (%)	A_{pc} (%)	IoU (%)
Complete network	98.79	95.21	94.80
Removing face attributes	98.71	94.97	94.65
Removing edge attributes	97.78	93.52	92.17
Removing UV network structure	96.55	92.98	91.69
Removing the graph contrast learning part	95.01	91.57	91.17

The results of the ablation experiment further demonstrate the advantages of our proposed method. Our method utilizes the features and attributes of the faces and edges of the part model, and further obtains geometric information through the UV grid. At the same time, graph contrastive learning is used for pre-training to improve the model performance. By combining hierarchical graph neural networks, better capture of global and local feature information has been achieved. These are organically combined to form an efficient and accurate manufacturability analysis framework, exhibiting excellent analysis results and generalization ability.

5.3. Case Study

In order to verify the manufacturability analysis effect of the proposed method, manufacturability analysis tests were carried out on the constructed test model and the actual part model of the enterprise. The constructed test model contains a variety of part defects or unreasonable process structures, while the actual part model of the enterprise may have unreasonable process structures or reasonable part models. The manufacturability analysis results are shown in Figure 10. For Part 1, it contains a structure with residual faces, which is a redundant region for the part and one that the cutting tool cannot enter. The corner radius on the pocket in Part 2 is 0.5 mm, which is too small to select a suitable tool for machining. Part 3 has a missing filet. The side wall of the pocket is usually cut by the side edge of the milling cutter. The missing filet causes the pocket to have a sunken sharp corner, which will not be processed, and the NC programmer needs to spend a lot of time establishing auxiliary lines. The side wall thickness of some structures in Part 4 is too small, which may cause large deformation or even collapse during cutting, and is a difficult-to-manufacture structure. The ratio of the depth to the diameter of the hole on Part 5 is too large, which causes the cutting tool to vibrate easily during machining, so the hole is also difficult to manufacture.

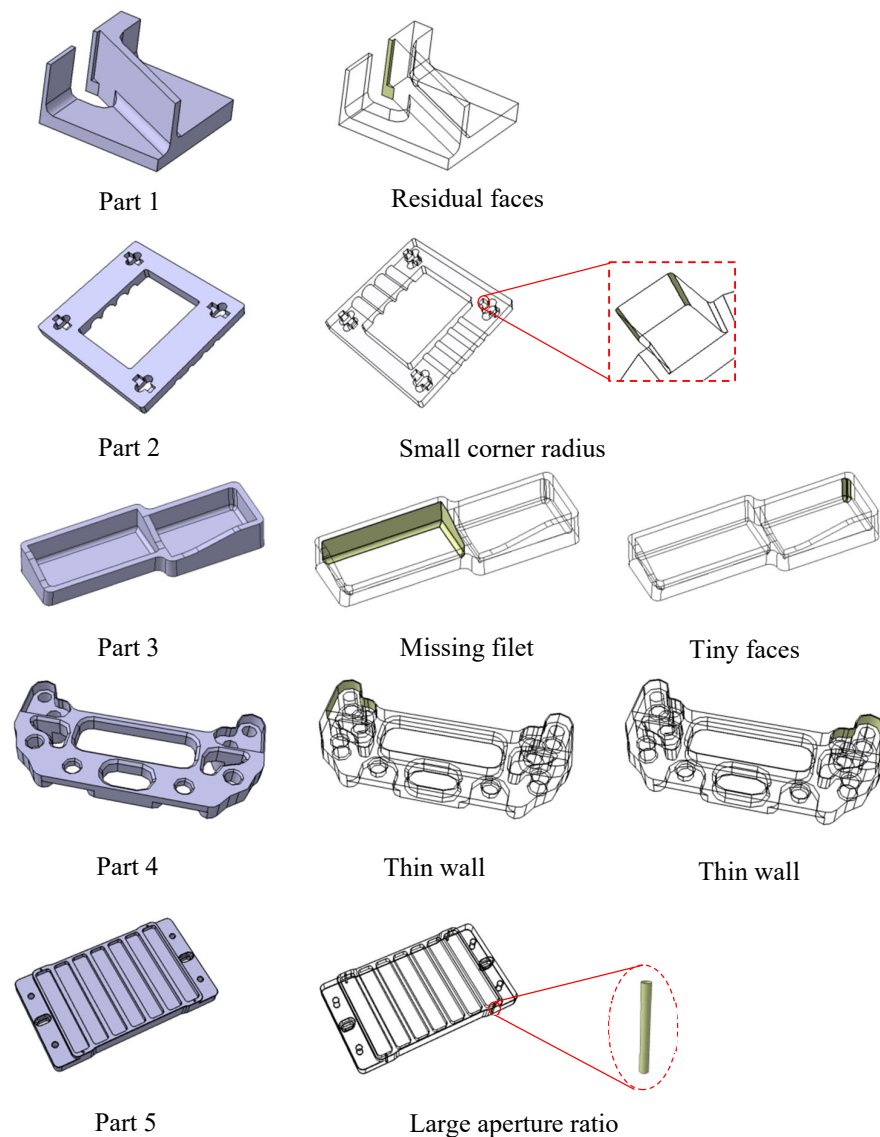


Figure 10. Manufacturability analysis results.

6. Conclusions and Future Work

In this paper, a manufacturability analysis method based on a graph neural network is proposed. First, the attribute adjacency graph and the UV grid are combined to characterize part model data, which can effectively represent the topological information and geometric information on the part model. At the same time, the parameter information on the spherical coordinate system is used to ensure rotation and translation invariance; then, based on the graph representation of the part model, a hierarchical graph neural network is constructed, which is divided into three levels: edge, node, and graph, for encoding and information transmission and updating, and the receptive field is expanded for better node classification to support manufacturability analysis. Finally, graph contrast learning is used as a regularization technique in the pre-training stage to maximize the similarity of graph representations between different views to improve prediction performance. Manufacturability analysis tests are carried out on the constructed part model dataset, and the experimental results show that this method has good performance. It specifically includes the following achievements:

- (1) By comparing with methods based on voxel and point cloud representation, it can be verified that our method performs well in accuracy, average class accuracy, IoU, and other indicators;
- (2) Through ablation experiments, it was verified that the representation forms of surface attributes, edge attributes, and UV network structure, as well as the adoption of graph contrast learning, can effectively improve the accuracy of manufacturability analysis results;
- (3) Manufacturability analysis was conducted on the constructed experimental model and the actual part model of the enterprise, verifying the ability to detect various part defects or structures with unreasonable processes, such as residual faces, small corner radius, tiny faces, thin wall, and large aperture ratio.

In the near future, several issues are worth further exploration to improve the practicability of our approach: (1) further expanding the part model database, enhancing the diversity and complexity of parts, expanding the situations that are difficult to manufacture, impossible to manufacture, or have high manufacturing costs, and modifying the network model structure according to the test results to enhance the practicality of the method; and (2) extending the method in this paper to edge- and graph-level tasks, such as part classification, typical structure recognition, and design intent capture. This method constructs a hierarchical graph neural network with three levels: node, edge, and graph; only the node level is used. Extending it to the edge and graph levels could improve the application value of this method.

Author Contributions: X.L. (Xiuling Li): conceptualization, original draft; B.H. and S.Z.: methodology formal analysis, supervision; X.L. (Xuewu Li): funding acquisition; F.L.: editing, project administration; P.W.: software, investigation. All authors have read and agreed to the published version of the manuscript.

Funding: The authors are grateful for the financial support provided by the 2022 annual key research and development and promotion projects in Henan Province (No. 222102220065), and the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515012096).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Battaia, O.; Dolgui, A.; Heragu, S.S.; Meerkov, S.M.; Tiwari, M.K. Design for manufacturing and assembly/disassembly: Joint design of products and production systems. *Int. J. Prod. Res.* **2018**, *56*, 7181–7189. [\[CrossRef\]](#)
2. Formentini, G.; Boix Rodríguez, N.; Favi, C. Design for manufacturing and assembly methods in the product development process of mechanical products: A systematic literature review. *Int. J. Adv. Manuf. Technol.* **2022**, *120*, 4307–4334. [\[CrossRef\]](#)
3. Shi, P.; Tong, X.; Xu, Y.; Cai, M. New classification method of nonstandard parts for CNC selection based on machine learning and geometric analysis. *J. Intell. Manuf.* **2025**, *36*, 1–24. [\[CrossRef\]](#)
4. Lin, Y.H.; Ting, Y.H.; Huang, Y.C.; Cheng, K.-L.; Jong, W.-R. Integration of deep learning for automatic recognition of 2D engineering drawings. *Machines* **2023**, *11*, 802. [\[CrossRef\]](#)
5. Li, Y.; Li, E.; Lenover, M.; Mann, S.; Bedi, S. Edge adjacency graph and neural network architecture for machining feature recognition. *Int. J. Adv. Manuf. Technol.* **2025**, *136*, 897–908. [\[CrossRef\]](#)
6. Li, W.; Li, B.; He, S.; Mann, S.; Bedi, S. A novel milling parameter optimization method based on improved deep reinforcement learning considering machining cost. *J. Manuf. Process.* **2022**, *84*, 1362–1375. [\[CrossRef\]](#)
7. Zhang, H.; Wang, W.; Zhang, S.; Zhang, Y.; Zhou, J.; Wang, Z.; Huang, B.; Huang, R. A novel method based on deep reinforcement learning for machining process route planning. *Robot. Comput.-Integr. Manuf.* **2024**, *86*, 102688. [\[CrossRef\]](#)
8. Huang, B.; Xu, C.; Huang, R.; Zhang, S. An automatic 3D CAD model errors detection method of aircraft structural part for NC machining. *J. Comput. Des. Eng.* **2015**, *2*, 253–260. [\[CrossRef\]](#)

9. Yang, W.; Fu, C.; Yan, X.; Chen, Z. A knowledge-based system for quality analysis in model-based design. *J. Intell. Manuf.* **2020**, *31*, 1579–1606. [\[CrossRef\]](#)
10. Bidkar, R.A.; McAdams, D.A. Methods for automated manufacturability analysis of injection-molded and die-cast parts. *Res. Eng. Des.* **2010**, *21*, 1–24. [\[CrossRef\]](#)
11. Kerbrat, O.; Mognol, P.; Hascoët, J.Y. A new DFM approach to combine machining and additive manufacturing. *Comput. Ind.* **2011**, *62*, 684–692. [\[CrossRef\]](#)
12. Xu, T.; Xue, J.; Chen, Z.; Li, J.; Jiao, X. A systematic method for automated manufacturability analysis of machining parts. *Int. J. Adv. Manuf. Technol.* **2022**, *122*, 391–407. [\[CrossRef\]](#)
13. Matuszek, J.; Seneta, T.; Moczyła, A. Assessment of the design for manufacturability using fuzzy logic. *Appl. Sci.* **2020**, *10*, 3935. [\[CrossRef\]](#)
14. Shi, Y.; Zhang, Y.; Baek, S.; De Backer, W.; Harik, R. Manufacturability analysis for additive manufacturing using a novel feature recognition technique. *Comput.-Aided Des. Appl.* **2018**, *15*, 941–952. [\[CrossRef\]](#)
15. Yeo, C.; Cheon, S.; Mun, D. Manufacturability evaluation of parts using descriptor-based machining feature recognition. *Int. J. Comput. Integr. Manuf.* **2021**, *34*, 1196–1222. [\[CrossRef\]](#)
16. Chen, N.; Frank, M.C. Design for manufacturing: Geometric manufacturability evaluation for five-axis milling. *J. Manuf. Sci. Eng.* **2021**, *143*, 081007. [\[CrossRef\]](#)
17. Mayerhofer, M.; Lepuschitz, W.; Hoebert, T.; Merdan, M.; Schwentenwein, M.; Strasser, T.I. Knowledge-driven manufacturability analysis for additive manufacturing. *IEEE Open J. Ind. Electron. Soc.* **2021**, *2*, 207–223. [\[CrossRef\]](#)
18. Zhang, Z.; Jaiswal, P.; Rai, R. FeatureNet: Machining feature recognition based on 3d convolution neural network. *Comput.-Aided Des.* **2018**, *101*, 12–22. [\[CrossRef\]](#)
19. Lei, R.; Wu, H.; Peng, Y. Mfpoinetnet: A point cloud-based neural network using selective downsampling layer for machining feature recognition. *Machines* **2022**, *10*, 1165. [\[CrossRef\]](#)
20. Ning, F.; Shi, Y.; Cai, M.; Xu, W.; Zhang, X. Manufacturing cost estimation based on a deep-learning method. *J. Manuf. Syst.* **2020**, *54*, 186–195. [\[CrossRef\]](#)
21. Yoo, S.; Kang, N. Explainable artificial intelligence for manufacturing cost estimation and machining feature visualization. *Expert Syst. Appl.* **2021**, *183*, 115430. [\[CrossRef\]](#)
22. Zhao, C.; Dinar, M.; Melkote, S.N. Deep learning and sequence mining for manufacturing process and sequence selection. *Int. J. Prod. Res.* **2024**, *62*, 5293–5314. [\[CrossRef\]](#)
23. Balu, A.; Ghadai, S.; Young, G.; Sarkar, S.; Krishnamurthy, A. A machine-learning framework for design for manufacturability. *arXiv* **2017**, arXiv:1703.01499. [\[CrossRef\]](#)
24. Ghadai, S.; Balu, A.; Sarkar, S.; Krishnamurthy, A. Learning localized features in 3D CAD models for manufacturability analysis of drilled holes. *Comput. Aided Geom. Des.* **2018**, *62*, 263–275. [\[CrossRef\]](#)
25. Peddireddy, D.; Fu, X.; Shankar, A.; Wang, H.; Joung, B.G.; Aggarwal, V.; Sutherland, J.W.; Jun, M.B.-G. Identifying manufacturability and machining processes using deep 3D convolutional networks. *J. Manuf. Process.* **2021**, *64*, 1336–1348. [\[CrossRef\]](#)
26. Wu, H.; Lei, R.; Peng, Y.; Gao, L. AAGNet: A graph neural network towards multi-task machining feature recognition. *Robot. Comput.-Integr. Manuf.* **2024**, *86*, 102661. [\[CrossRef\]](#)
27. Zhang, H.; Wang, W.; Zhang, S.; Zhang, Y.; Zhou, J.; Wang, Z.; Huang, B.; Huang, R. A novel method for intersecting machining feature segmentation via deep reinforcement learning. *Adv. Eng. Inform.* **2024**, *59*, 102256. [\[CrossRef\]](#)
28. Jing, Y.; Zhou, G.; Zhang, C.; Chang, F.; Yan, H.; Xiao, Z. XMKR: Explainable manufacturing knowledge recommendation for collaborative design with graph embedding learning. *Adv. Eng. Inform.* **2024**, *59*, 102339. [\[CrossRef\]](#)
29. Xie, L.; Lu, Y.; Furuhashi, T.; Yamakawa, S.; Zhang, W.; Regmi, A.; Kara, L.; Shimada, K. Graph neural network-enabled manufacturing method classification from engineering drawings. *Comput. Ind.* **2022**, *142*, 103697. [\[CrossRef\]](#)
30. Deng, T.; Li, Y.; Liu, X.; Wang, P.; Lu, K. A data-driven parameter planning method for structural Parts NC machining. *Robot. Comput.-Integr. Manuf.* **2021**, *68*, 102080. [\[CrossRef\]](#)
31. Zhang, H.; Wang, W.; Zhang, S.; Huang, B.; Zhang, Y.; Wang, M.; Liang, J.; Wang, Z. A novel method based on a convolutional graph neural network for manufacturing cost estimation. *J. Manuf. Syst.* **2022**, *65*, 837–852. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.