3D CAD Models Reconstruction from 2D Drawings: A Survey and Future Challenges

Chao Zhang[1], Romain Pinquié[2], Arnaud Polette[3], Mirai Iida[4], Henri De Charnace[5], Jean-Philippe Pernot[6],

[1]Arts et Métiers Institute of Technology, chao.zhang@ensam.eu
[2]Universit Grenoble Alpes, romain.pinquie@grenoble-inp.fr
[3]Arts et Métiers Institute of Technology, arnaud.polette@ensam.eu
[4]Cognitive Design Systems, mirai@cognitive-design-systems.com
[5]Cognitive Design Systems, henri@cognitive-design-systems.com
[6]Arts et Métiers Institute of Technology, jean-philippe.pernot@ensam.eu

Corresponding author: Jean-Philippe Pernot, jean-philippe.pernot@ensam.eu

Abstract. Even though 3D CAD models are mainstream to support many industrial applications, they are not always available for some companies (e.g. suppliers) that only have access to 2D drawings. Being able to transform 2D drawings into CAD models automatically would be of great interest. 3D reconstruction from 2D drawings is one of the most efficient technologies for producing 3D CAD models. Current methods usually generate dumb B-Rep models, which are not editable and therefore difficult to modify. In this paper, we studied the state-of-the-art solutions to reconstruct parametric CAD models from 2D drawings. The reconstructed 3D model is parameterized, and it can be edited within a CAD modeler. Three types of existing works are analyzed: CSG-based, B-Rep-based, and deep learning methods. This study reveals that CSG-based and B-Rep-based methods do not allow efficient reconstructions, which are therefore hard to edit in a later stage. In contrast, the graph-based and sequence-based representations as well as the deep neural networks such as graph networks and Transformers have the potential to reconstruct 3D parametric and editable CAD models from 2D drawings. The evaluation metrics are summarized and used to compare the reconstruction strategies and methods. The open-source datasets are also analyzed and compared from various criteria to help the future research fast start.

Keywords: CAD model reconstruction, Parametric modeling, CSG and B-Rep modeling, Deep neural networks.
DOI: https://doi.org/10.14733/cadaps.2026.646-678

1 INTRODUCTION

Nowadays, CAD modelers have become popular tools, and 3D models are mainstream all along the Product Development Process (PDP), and in particular to support the simulation and manufacturing phases. Although institutes and companies increasingly use 3D modeling to support their activities, 2D drawings still play an essential role in engineering practices and often serve as definitive contract references that guide the manufacture or assembly of products. Thus, in many cases, 3D CAD models are simply not available. Moreover, the information embedded in 2D drawings cannot be directly used in today's 3D CAD systems, and it is time-consuming to reconstruct them manually, and this is the reason why the automatic conversion of 2D drawings to editable 3D models is still challenging for researchers.

Reconstructing 3D models from 2D drawings is a topic that has received much attention in the last several decades. Indeed, the process of reconstructing 3D models is still long and tedious, which is not good for the competitiveness of companies. Despite the existence of modeling tools, it is still impossible to start with a 2D drawing depicting several views of a solid to reconstruct the 3D CAD model automatically. Current methods often focus on geometric analysis and take little consideration of semantic information in 2D drawings, such as dimensions and annotations. However, this information is crucial for better characterizing certain geometric entities, particularly parametrized surfaces. In addition, these reconstruction methods generally produce non-parameterized CAD models for which the modifications are made through low-level operations. This is not suited to the needs of designers who prefer to edit feature shapes rather than faces, edges, and vertices of the B-Rep (boundary representation) model [54]. Figure 1 (a) shows an example of a 2D orthographic drawing that includes entities such as vertices and edges, and Figure 1 (b) the reconstructed dumb B-Rep that is hard to edit, even using direct modeling techniques [114] with push/pull operations. Ideally, the reconstructed model is to be represented with a succession of parameterized features organized in a building tree, as illustrated on Figure 1 (c), and which can be edited subsequently.

Finally, current approaches struggle to reconstruct the 3D model when the information in the 2D drawing is incomplete and does not ensure the uniqueness of the solution. Indeed, several reconstruction strategies exist for a given 2D drawing, and optimizing how models are reconstructed is a very challenging task. Thus, the objectives of this survey paper are as follows:

- Analyze the alternative solutions to reconstruct 3D CAD models from 2D drawings automatically. Here, the reconstructed CAD model is parameterized in whole or in part, and it should be possible to use an existing CAD modeler to make modifications.
- Discuss the challenges in CAD reconstruction that remain in both rule-based and learning-based methods, and in particular limitations of the algorithms, data representation methods, and quality of the reconstructed results.
- Compare and characterize the evaluation metrics and the open-source datasets use to validate the approaches developed so far.

with the ultimate aim of providing relevant results and analyses that can serve as the basis for next-generation developments in 3D reconstruction. More precisely, we studied the rule-based reconstruction approaches, as well as CSG-based and B-Rep-based methods. The learning-based methods with learnable parameters are also discussed. Figure 2 shows the overview of the methods studied in this survey paper. The CSG-based and B-Rep-based methods first extract the 2D geometric entities, and convert them to 3D primitives and 3D wireframes, respectively. The CAD models are created using the boolean and face identification operations. On the other hand, deep learning approaches show impressive performance for CAD modeling. In this case, the way CAD models are represented is an important issue. Indeed, the 2D and 3D representations must be converted to network-friendly formats, such as images, point clouds, graphs, etc., to adapt to requirements of deep neural networks. After the network processing, the representations produced should be converted back to CAD models. To this aim, the graph representation is a good candidate as it contains the topology

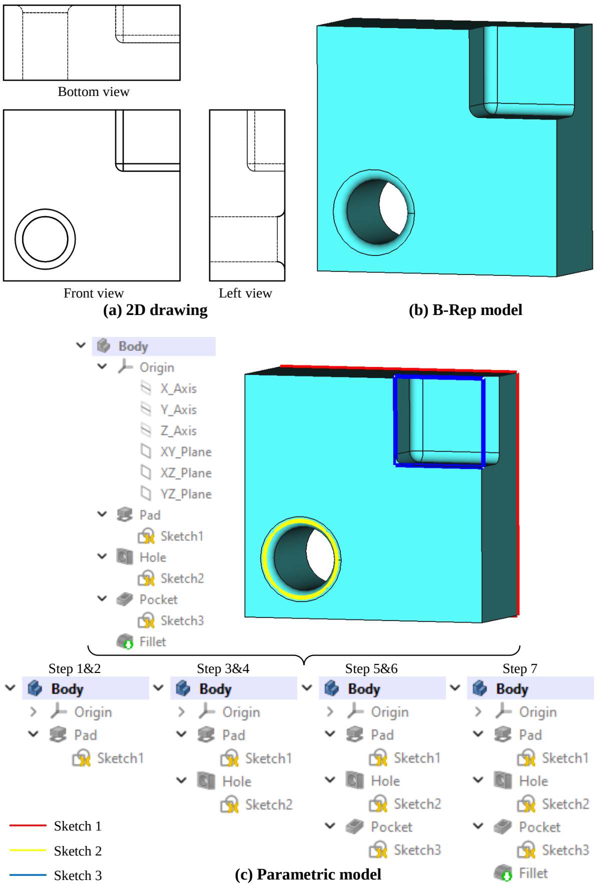

Figure 1: Three types of CAD models. (a) the 2D orthographic drawing with three views; (b) the B-Rep model without building history and parameters; (c) the feature-based parametric model with building tree and the parameters.

and geometric information of CAD models. Also, the parametric modeling operations can also be converted to CAD command sequences consumed by the neural networks. However, despite the efforts made so far, CAD models reconstructed from CSG-based and B-Rep-based methods still output dumb representations. The reconstructed CSG and dumb B-Rep models do not have building tree and editable parameters, and are not yet friendly for downstream applications.

In summary, our contributions are threefold: (i) we compared the rule-based, and learning-based reconstruction methods with respect to different criteria and we summarized the advantages and disadvantages of the different methods; (ii) we compared the evaluation metrics such as Coverage, Jensen-Shannon Divergence, Intersection over Union, etc. (iii) we compared the CAD datasets, the details of CAD representations, labels, and the number of cases, etc. In the end, this study will serve as a reference to help future researchers quickly enter the field of 3D modeling and reconstruction.

This paper is organized as follows. Section 2 introduces the categories of approaches as well as the evaluation criteria. Section 3 compares the existing rule-based works that include CSG-based and B-Rep-based methods. Section 4 introduces the learning-based methods together with the possibly adopted CAD representations. Section 5 describes the open source databases, as well as the data representations. Section 6 introduces the widely used evaluation metrics. Section 7 discusses the challenges and future challenges related to 3D reconstruction. Section 8 concludes the paper and discusses future works.

2 CATEGORIES OF APPROACHES AND ADOPTED EVALUATION CRITERIA

This section introduces the categories of approaches used for reconstructing 3D CAD models from 2D drawings, along with the evaluation criteria adopted to enable a fair comparison among them.

2.1 Categories of Approaches

3D CAD model reconstruction from 2D drawing has been a long-standing research topic since 1970s. Some prior surveys of 3D CAD model reconstruction from 2D drawing and CAD neural representation are provided in [21, 82, 23, 72]. Two main categories can distinguished, the rule-based and learning-based approaches.

On one hand, the rule-based reconstruction approaches, such as CSG-based and B-Rep based methods, are reviewed in [21, 82]. In [21], different approaches have been discussed with respect to bottom-up and top-down reconstruction strategies, and other hyper-parameters, such as type of drawing, hidden lines, etc., have also been considered. In [82], the authors reviewed the methods that reconstruct the 3D solid models from 2D orthographic drawings. However, these surveys were published over a decade ago and do not offer an overview of recent methods. More recently, Feist et al. [23] discussed the different methods used to generate 3D models from 2D drawings. They focused on the coverage and completeness of the reconstruction process. However, they pay more attention to the papers that regard building and environment, and they paid little attention to applications and case studies in the field of mechanical engineering. Finally, these review papers did not discuss the reconstruction methods that involve artificial intelligence (AI) algorithms.

On the other hand, learning-based methods have become popular in CAD modeling. In [72], the authors reviewed geometric representations used in CAD modeling with a focus on learning-based methods, in particular those involving neural networks. Their survey highlights several recent works in neuro-symbolic models to represent 2D and 3D CAD models. However, they did not specifically address the question of reconstructing 3D shapes from 2D drawings.

This paper revisits and extends these prior surveys, offering a deeper analysis based on newly defined comparison criteria introduced in the following section. Figure 2 illustrates the reconstruction steps associated with the three categories methods under study.

2.2 Adopted Evaluation Criteria

To conduct appropriate and systematic comparisons between the CAD reconstruction approaches, several criteria and their gradations are proposed in this section. A total of 17 evaluation criteria are defined across three levels: 2D drawing (10 criteria), 3D reconstruction (4 criteria), and CAD operation (3 criteria).

For 2D drawing, criteria pertain to the number of views, the types of lines and the types of curves. Table 1 outlines the criteria for the number of views in 2D drawings. Here, three situations are considered: the 2D drawing can have two views, three views, or more than three views here considered as a multi-view. Table 2 presents the line types in 2D drawings. Three line types are introduced: continuous, dashed, and center lines. Each type conveys a different meaning. More precisely, the continuous line represents visible entities, while the dashed line represents hidden entities in the current view, and the center line depicts the center of a hole, an axis or a symmetry. Table 3 presents the criteria for curve types in 2D drawing. The curves are projections of 3D shapes onto a 2D view. Four curve types are proposed, which include three regular types: line segment, arc, and circle, along with one irregular type, the Bézier curve.

For the 3D reconstruction itself, two types of evaluation criteria are presented. At the face level, planar, curved, and freeform surfaces are proposed. Table 4 illustrates the details. Further to this, table 5 uses the

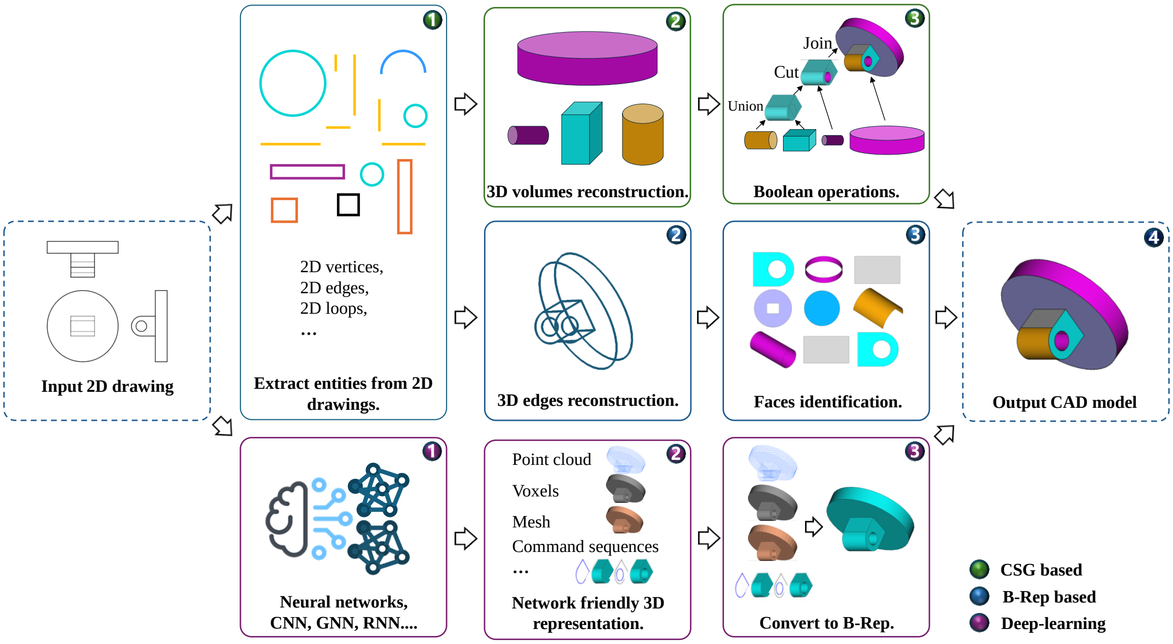

Figure 2: The reconstruction steps of three approaches: CSG-based, B-Rep-based, and learning-based. In CSG-based and B-Rep-based methods, the entities are first extracted from 2D drawings and then followed by reconstruction operations. The 2D and 3D CAD models must represent network-friendly structures and feed to the networks for deep learning methods.

Table 1: Criteria of 2D drawing views.

Drawing level		Gradation of criteria	
Level	Criteria	⊞	⊟
a	two-view	Yes	No
b	three-view	Yes	No
c	multi-view	Yes	No

Table 2: Criteria of 2D drawing line types.

Line level		Gradation of criteria	
Level	Criteria	⊞	⊟
d	continuous	Yes	No
e	dash	Yes	No
f	center	Yes	No

Table 3: Criteria of 2D curve types.

Curve level		Gradation of criteria	
Level	Criteria	$\boxplus$	$\boxminus$
g	line segment	Yes	No
h	arc	Yes	No
i	circle	Yes	No
j	Bézier curve	Yes	No

editable property as a criterion to evaluate whether the method allow the reconstruction of editable parametric model, or not.

Table 4: Criteria of reconstructed face types.

Face level		Gradation of criteria	
Level	Criteria	$\boxplus$	$\boxminus$
k	planar	Yes	No
l	curved	Yes	No
m	freefrom	Yes	No

Table 5: Criteria of CAD editable.

Editable level.		Gradation of criteria	
Level	Criteria	$\boxplus$	$\boxminus$
n	parametric	Yes	No

Finally, the types of operations used in the reconstruction are evaluated using the criteria listed in table 6, and may include extrusion, revolution, and sweep.

Table 6: Criteria of CAD operations.

Operation level		Gradation of criteria	
Level	Criteria	$\boxplus$	$\boxminus$
o	extrusion	Yes	No
p	revolution	Yes	No
q	sweep	Yes	No

2.3 Adopted Geometric Representations

In addition to the previously introduced criteria, the methods evaluated in this survey are also compared based on the types of input and output representations. Indeed, 2D drawings and CAD models often require intermediate processing, as they are generally not compatible with most CAD modeling algorithms, in particular those based on deep learning methods.

The common representations of 2D drawings include vectorized 2D drawings and images. The vectorized 2D drawings can be subsequently converted to vertex adjacency graphs, while images are represented as matrices. Depending on the adopted representation, the architecture of the developed algorithm will be different.

Similarly, 3D CAD models can be represented in various formats, such as dumb models and parametric models. Voxel, point cloud, mesh, and B-Rep formats are considered non-parametric or dumb representations, offering limited editability of CAD models and varying significantly in terms of geometric accuracy (continuous representation vs. discrete approximation). Here also, depending on the architecture of the algorithm, these representations can be subsequently transformed in other ones, such as face adjacency graphs. Ideally, the output of 3D reconstruction from a 2D drawing is a parametric CAD model that encapsulates building information, such as its sequence of CAD modeling operations, and supports easy modifications through parameter adjustments.

3 RULE-BASED CSG AND B-REP RECONSTRUCTION METHODS

In this section, we introduce the CSG-based and B-Rep-based methods, which rely on rules. Figure 2 illustrates the reconstruction pipeline of these methods. These methods will be evaluated according to our established criteria.

3.1 CSG-based Methods

The CSG-based method recognizes volumes by matching patterns and then combines these volumes using Boolean operations to obtain the final 3D models. In the work of [14], they proposed a CSG-based approach that uses three orthographic views as input. The components are constructed by extrusion and revolution operations. The solids are obtained by intersect operations. However, this approach only recognized two types of features, i.e. holes and pockets. A hint-based method is proposed by Lee et al. [52] to handle the solids of revolution from orthographic views. In this method, circular edges are the clues in the first view, and vertices and edges in other views will match the circular edges. Extrusion and Boolean operations are used for the construction of solid objects. However, this method is only utilized to process solids of revolution, not for all types of solids. In the work of [17], Dimri et al. explored how sectional views are allowed as input. This approach constructs the 3D solid by Boolean combinations of elementary solids. The elementary solids are formed by a sweep operation on loops identified in the input sectional views. However, this study requires manual specification of the types of sectional views, such as full sections, removed sections, etc. Moreover, Gorgani et al. [30] try to build a volume set from orthographic views by extrusion operations. A subset of those volumes is selected as the reconstructed 3D model, and new orthographic views of the reconstructed model are generated. The generated orthographic views are compared with the original ones until complete conformity between those views.

The CSG-based methods could be summarized by the following steps:

- Obtaining 2D entities from 2D drawings;
- Reconstructing 3D primary volumes from 2D entities;
- Performing Boolean operations on 3D primitives;
- Creating 3D models as the final object.

Actually, the most challenging aspect of CSG-based methods is extracting entities from 2D drawings and reconstructing them into primitives [22]. Generally, reconstructing primitives from 2D drawings is difficult; the hidden lines are hard to identify (e⊟), and CSG-based CAD models are not easily editable (n⊟). Additionally, there are currently no methods that support the reconstruction of freeform surfaces (m⊟). Table 7 summarizes the evaluation results of the surveyed methods according to the defined criteria.

3.2 B-Rep Based Methods

The B-Rep based method constructs the 3D models by generating 3D vertices followed by 3D edges and then forming the wireframe models. Then, the reconstructed wireframe is used to build the surfaces forming the boundaries of the object.

The wireframe-based method has been proposed in the paper [27], which aimed to construct the 3D curvilinear wireframe from three orthographic views. Moreover, Gong et al. [28] proposed a B-rep-oriented method to reconstruct solid objects. This method proposes a new hybrid wireframe consisting of geometry, topology, vertices, and edges. However, they require three views with perfect line drawings to handle objects like line segments, circles, and arcs. As an extension, the work of [29] introduced an algorithm based on graph theory that could convert the hybrid wireframe to the final B-Rep model by extracting the rest of the planar faces. Another sectional view-based 3D wireframe reconstruction method has been presented in [18]. The 2D views with the same cutting direction are merged into a single view, and the omitted elements are recovered according to information matching. However, the inner wires of the solid are still not completed because the number of sectional views is limited. Furferi et al. [24] developed a MatLab tool that reconstructs 3D wireframes from 2D vectorial drawings by labeling vertices and topological representation of edges. These approaches reconstruct the 3D wireframes from 2D drawings but do not consider the reconstruction of the surfaces from the wireframes (k⊟, l⊟, m⊟).

To identify surfaces from the wireframes, Varley et al. [80] proposed a polynomial-order algorithm that could handle 3D wireframes. Alexei et al. [102] introduced a spectral graph matching algorithm to construct 3D faces of models from wireframes. Five different categories of wire models of structural components are defined and represented as matrices. However, the number of vertices for each type of wire is fixed, which means only five types of faces can be reconstructed (k⊞, l⊞). Hoang et al. [40, 39] uses two orthographic views to create 3D candidate vertices and reconstruct the 3D faces by searching wireframe loops. The proposed method works on perfect 2D drawings, which is limiting since 2D drawings are not always accurate and consistent in engineering.

In the work of [55], Lee et al. presented a cubic corner-based method capable of recovering 3D polyhedral objects from accurate 2D drawings. To deal with the problem of inaccurate 2D drawings, an improved method has been introduced in [56]. The improved method combines the cubic corner method and the optimization method. However, it is less efficient than the cubic corner method. Zhao et al. [110] provide a theoretical possibility of automatically reconstructing compound objects from two 2D views based on AutoCAD software. Besides, Wagh [83] uses intersection operation to reconstruct the 3D points and edges from orthographic views. However, these approaches work only for 3D models with no curved surfaces (l⊟).

Several methods have been proposed to deal with the issue of curved surfaces (l⊞). In [103], the authors have introduced a two-stage method that automatically reconstructs 3D CAD models with curved surfaces from 2D orthographic drawings. However, the method cannot deal with complex faces like free-form surfaces (m⊟), and it generates dumb CAD models that cannot be easily modified later (n⊟). In the work of [11], they developed an algorithm called '3D-Space' that explores the possibility of 3D model solutions starting from a prismatic volume. Moreover, in [108], the authors proposed a 3D reconstructing strategy by merging semantic and graphic process planning information to generate serial 3D models for the dynamic evolution of rotational parts. But, in this work, they were only concerned with the rotational parts. Besides, the proposed method is unsuitable for the part composed of freeform surfaces (m⊟). Sobani et al. [76] presented a method to reconstruct 3D models from 2D image sequences by edge detection and boundary extraction. However, the 2D image sequence requires 36 images, and it is hard to obtain this kind of image as a sequence during the processing of reconstruction. In [78], a method of reconstructing 3D models from 2D drawings by extracting features is proposed. This method defines the sketches of cuboids, cylinders, holes, and fillets as features. Harish et al. [37] have introduced a method that uses OpenCV to detect the contours from the 2D views. The features extracted from 2D images were utilized to recognize the shapes. But, only six basic shapes are considered (triangle, square, rectangle, pentagon, hexagon, and circle). Ramos et al. [71] introduced a contour-aware 3D reconstruction approach that could process the curves of sketches by estimating the normals and depths. However, the proposed method cannot handle the open and overlapping curves.

Another solution is to combine the CGS and B-rep methods. If two primitives are neighbors, they must reference the same part of their wire, so the adjacency information between the primitives is needed. And the intersections between the neighbor primitives define the surface boundaries [6, 7]. Wang et al. [87] proposed an approach, which is a hybrid of the B-rep and CSG methods for the reconstruction of engineering drawings. However, the presented algorithm cannot reconstruct the symmetric rotational objects such as cones and toroids. In addition, Tam et al. [77] extract closed loops from 2D views to build the parts of a solid. Boolean operations then combine these parts to form the 3D models finally. An advantage of this method is that it could produce CSG models. But, it is only suited for simple kinds of representation. Governi et al. [32] proposed a method that starts with drawing orthogonal views. Three orthogonal views are extruded pixel by pixel in the respective normal direction, thus forming three voxel models. A logic intersection combines the obtained 3D models to obtain a wireframe represented by a parametric spline. A major drawback of this approach is that it can be applied only to simple 2D drawings. An improved method for enhancing the surfacing process by exploiting shading information is proposed in [10]. However, these methods mentioned above could be applied to curves and curved surfaces, but they are unsuitable for complex 3D models.

In another study, fuzzy logic and genetic algorithms have been implemented in 3D reconstruction. In [89], the authors presented an approach based on elements of knowledge retrieval, variational geometry, and graph theoretic methods to represent 3D models. The local graphs of each view are created using the variational geometry method. These local graphs are then merged into a global graph. The global graphs are transformed into 3D models using the knowledge retrieval method. However, defining a knowledge system that includes all types of features in 2D drawings is impossible. The method of fuzzy logic theory has been proposed in paper [88], to deal with the issues of semantic information loss when a 3D model is constructed from 2D drawings. However, the ability of the fuzzy classifier to distinguish the different types of elements is limited. Gorgani et al. [31] use the fuzzy theory to analyze and estimate the surface. However, it can be applied only to models with planar faces (k⊞) and does not work with curved faces (l⊟). Chen et al. [13] developed a systematic 3D reconstruction method, which transforms binary images of 2D drawings into 3D models with the aid of the genetic algorithm. Moreover, Liu et al. [62] use a genetic algorithm to identify the faces of the models in the 2D wireframes. They demonstrate the successful results of the proposed method. Also, Gorgani et al. [34] introduced an optimization of the genetic algorithm used to recognize the relationship among the components of various views in 2D drawings. However, both approaches require much tuning, making them difficult to practice.

In summary, the B-rep approach consists of the following steps:

- Transformation of 2D vertices to 3D vertices;
- Generation of 3D edges from 3D vertices;
- Construction of faces from 3D edges;
- Creation of 3D objects from faces.

B-Rep based methods reconstruct 3D models at the face level, as CAD operations are not utilized during the reconstruction process (o⊟, p⊟, q⊟). However, the reconstructed CAD model is a dumb B-Rep without parameters. The reconstructed B-Rep model lacks CAD building information such as the construction tree, features, and constraints, which significantly limits its editability (n⊟). Moreover, these methods require the perfect 2D drawing to extract the entities, such as 2D vertices and edges. Only the existing methods could handle the limited types of edges and surfaces. The curved surface is still not well supported for reconstruction (l⊟). Some methods need to assume the 2D drawing only contains line segments, arcs, and circles. In short, rule-based methods have clear algorithmic processes and are more explainable and understandable. However, these methods still fall short in terms of reconstruction performance when measured against engineering requirements.

Table 7: Summary of rule-based methods according to the evaluation criteria.

Criteria	2D drawing										3D model				Operation		
	Drawing			Line			Curve				Face			Edit	-		
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
Cicek et al. [14]	⊟	⊞	⊟	⊞	⊞	⊞	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊞	⊞	⊟
Lee et al. [52]	⊟	⊞	⊟	⊞	⊟	⊞	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊞	⊟
Dimri et al. [17]	⊟	⊞	⊟	⊞	⊞	⊟	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊟	⊞
Gorgani et al. [30]	⊞	⊟	⊟	⊞	⊟	⊟	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊞	⊟	⊟
Gong et al. [27, 28, 29]	⊟	⊞	⊟	⊞	⊟	⊞	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊟	⊟
Ding et al. [18]	⊞	⊞	⊞	⊞	⊞	⊞	⊞	⊞	⊞	⊟	⊟	⊟	⊟	⊟	⊟	⊟	⊟
Furferi et al. [24]	⊟	⊞	⊟	⊞	⊞	⊟	⊞	⊟	⊟	⊟	⊟	⊟	⊟	⊟	⊟	⊟	⊟
Zakharov et al. [102]	⊟	⊞	⊟	⊞	⊞	⊟	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊟	⊟
Hoang et al. [40, 39]	⊞	⊞	⊟	⊞	⊞	⊟	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊟	⊟
Wagh et al. [83]	⊟	⊞	⊟	⊞	⊟	⊟	⊞	⊞	⊟	⊟	⊞	⊟	⊟	⊟	⊟	⊟	⊟
Zhang et al. [103]	⊟	⊞	⊟	⊞	⊞	⊟	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊟	⊟
Harish et al. [37]	⊟	⊞	⊟	⊞	⊟	⊟	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊟	⊟
Wang et al. [87]	⊟	⊞	⊟	⊞	⊞	⊟	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊟	⊟
Weiss et al. [89]	⊟	⊞	⊟	⊞	⊞	⊟	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊟	⊟
Wang et al. [88]	⊟	⊞	⊟	⊞	⊞	⊟	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊟	⊟
Chen et al. [13]	⊟	⊞	⊟	⊞	⊞	⊟	⊞	⊞	⊞	⊟	⊞	⊞	⊟	⊟	⊟	⊟	⊟

3.3 Summary of Rule-based Methods

The characteristics of the rule-based methods are illustrated in Table 7. Clearly, the orthographic drawings with three views are most commonly used for reconstruction tasks (b⊞). The continuous lines are the essential entities of 2D drawings (d⊞), but only a few methods can extract the hidden entities (dashed lines) and center lines from the 2D drawings. Overall, the regular curves (line, arc, circle) are well supported (g⊞, h⊞, i⊞), but there are no methods that can handle the Bézier curve, and since the Bézier curve is not supported in 2D drawings, freeform surfaces also cannot be reconstructed efficiently (j⊞, m⊞). B-Rep based methods work at the surface level, so they do not require CAD operations (o⊟, p⊟, q⊟). CSG-based methods need CAD operations to reconstruct primitive shapes and combine them. The main limitation of rule-based methods is that they produce non-parametric reconstructions, which are difficult to modify using standard CAD modeling tools (n⊟).

Table 8 summarizes rule-based methods across four aspects: input data, input representation, output representation, and reconstruction algorithms. CSG-based methods usually take orthographic views as input and convert them into vectorized drawings. These methods then use extrusion, revolution, sweep, and Boolean operations to reconstruct CSG shapes. In contrast, B-Rep based methods use orthographic views as input, represented as graphs, matrices, or vectorized drawings. The outputs are dumb B-Rep models that are difficult to edit. The reconstruction algorithms are detailed in Table 8. The limitations of rule-based methods are summarized as follows:

- For CSG-based methods, the challenge lies in efficiently identifying the entities from the 2D drawings to reconstruct the corresponding primitives. As a result, these methods are typically restricted to simple CAD models and generate CSG representations that lack geometric parameters, making them difficult to edit.
- B-Rep based methods reconstruct 3D CAD models in dumb B-Rep format, but the building information is lost, which makes the CAD model hard to edit. Curved surfaces are not well reconstructed from wireframe. Moreover, the computing cost is high for reconstructing complex CAD models.

Given the limitations of rule-based methods, there is a growing need to explore alternative approaches. Learning-based methods, particularly those leveraging deep learning, offer the potential to overcome these challenges by learning mappings from 2D drawings to 3D models, capturing more complex patterns, and enabling reconstruction of more editable and semantically rich CAD representations.

4 ON THE USE OF LEARNING-BASED METHODS

Deep learning has achieved significant success across various domains, driven by the rapid development of computational powers (e.g., GPUs) and the increasing availability of large-scaled datasets. Many machine learning tasks that heavily relied on handcrafted feature engineering, such as digital image processing and natural language processing, have been solved by various neural network architectures like convolutional neural networks (CNNs), recurrent neural networks (RNNs), and graph neural networks (GNNs). This success has boosted research on geometric processing and computer-aided design (CAD). However, since neural networks cannot directly process native B-Rep models, deep-learning approaches in CAD typically require converting CAD models into network-friendly representations, such as voxels, point clouds, meshes, graphs, and sequences. This section reviews deep learning methods according to these representation types. While these methods have not been widely applied to the specific task of 2D-to-3D reconstruction, they offer promising foundations for adaptation to this challenge, which remains underexplored in the current literature.

4.1 Representation of 2D Drawings

2D drawings are often represented using vectors. Indeed, vectorized 2D drawings have several advantages compared to raster images, such as being scale-independent and supporting primitive-level editing. Vector representations are defined on a 2D plane by primitives like vertices, line segments, and curves. However, in many cases, technical drawings are only available in raster form.

To vectorize raster images, [37] proposes a method based on OpenCV to detect contours in the images, which include the vertices. In [20, 41], the authors use object detection methods to detect the annotations in the drawings to understand the semantic information in the images. Transformer-based methods are widely used to convert images into vectorized drawings. In [19], raster images are converted into sequences of primitive parameters, and in [74], images are converted into vectorized drawings with sequences of edges. Furthermore,

Table 8: Summary of the existing rule-based approaches considering the input/output representations and the underlying reconstruction methods.

	References	Input	Input representation	Output representation	Methods
CSG	Cicek et al. [14]	orthographic drawing ($.dxf$)	vectorized drawing	CSG	extrusion, boolean
	Lee et al. [52]	orthographic drawing ($.dxf$)	vectorized drawing	CSG	hint-based recognition, boolean
	Dimri et al. [17]	sectional drawing ($.dxf$)	vectorized drawing	CSG	sweep, boolean
	Gorgani et al. [30]	orthographic drawing (-)	points, curves	CSG	extrusion, comparative
B-Rep	Gong et al. [27, 28, 29]	orthographic drawing (-)	vectors, graphs	dumb B-Rep	graph theory, matching algorithm
	Ding et al. [18]	sectional drawing ($.dxf$)	vectorized drawing	3D wireframes	matching algorithm
	Furferi et al. [24]	orthographic drawing ($.dxf$)	matrices	3D wireframes	rules-based
	Zakharov et al. [102]	orthographic drawing ($.dxf$)	graphs, matrices	dumb B-Rep	graph theory, matching algorithm
	Hoang et al. [40, 39]	orthographic drawing ($.dxf$)	vectorized drawing	dumb B-Rep	rules-based
	Wagh et al. [83]	orthographic drawing (-)	vertices, edges	dumb B-Rep	rules-based
	Zhang et al. [103]	orthographic drawing ($.svg$)	vectorized drawing	dumb B-Rep	pattern-matching, clustering algorithm
	Harish et al. [37]	orthographic drawing (img)	matrices	dumb B-Rep	contour detection, boolean operation
	Wang et al. [87]	orthographic drawing ($.dxf$)	vectorized drawing	dumb B-Rep	boolean operation
	Weiss et al. [89]	orthographic drawing (-)	graphs, vectors	dumb B-Rep	variational geometry, composite graphs methods
	Wang et al. [88]	orthographic drawing ($.dxf$)	vectorized drawing	operation commands	Fuzzy logic recognition
	Chen et al. [13]	orthographic drawing (img)	matrices, graphs	dumb B-Rep	genetic algorithms

2D sketch-based modeling [8] can represent sketches using graphs. In [73], the 2D sketches are depicted as graphs, where a node represents an edge, and the graph link represents a constraint.

In summary, 2D drawings considered in this survey can be represented in various forms, including raster images, parameterized sequences, or feature-enriched graphs obtained through a vectorization step.

4.2 Approximate Representations of CAD Models

This section explores how CAD models can be adapted for processing by deep learning methods, with a particular focus on techniques that approximate these models using voxels, point clouds, or meshes.

Voxels CNN has demonstrated outstanding performance in processing regular structure data, such as image classification and segmentation. However, 3D CAD models are typically represented using B-Rep models, which own irregular structures that are unordered. To adapt to CNN, the existing approaches consider converting 3D representations to regular structures by mapping 3D models to voxels. Voxels are defined in 3D space with regular grids, which makes them suitable for CNN. To recognize machining features in 3D models, Zhang et al. [109] developed a 3D CNN-based classification network that operates on the 3D voxel grid. For reverse engineering, a differentiable autoencoder network is proposed in [50] to predict the image-based sketch profiles (n⊞). The encoder can encode the input voxels into embedding and feed them to the decoder for image profile generation. Li et al. [60] trained a self-supervised network on voxel data to reconstruct the sketch-and-extrude operations (n⊞, o⊟). In [59], an unsupervised network is proposed to predict sketch-based feature modeling operations and reconstruct CAD models from voxels based on boxes and paths. In [93], an RNN-based method is proposed for representing and generating 3D shapes through sequential part assembly. Each 3D part is converted into a voxel representation and encoded into a feature embedding as input of the network. However, 3D CNN involve a large number of learnable parameters, which require substantial computational resources. To mitigate this, CAD models are approximated using low-resolution voxel grids, such as 64^3. While this reduces the computational burden, it also results in significant loss of geometric detail, thereby compromising the accuracy of subsequent processing.

Point clouds Point clouds are essential 3D representations with irregular structures. Several networks are designed to handle point clouds, such as PointNet [68, 69], and DGCNN [86]. CAD model reconstruction from point clouds is an important subject of reverse engineering. In [33] Guo et al. attempted to construct CAD models from point clouds. The authors used a CNN to extract spatial features from point clouds and employed three paths of transformer networks to output the three groups of primitive elements that support freeform surfaces (m⊞). Uy et al. proposed Point2Cyl [79], which transforms a point cloud into a set of extrusion cylinders using a segmentation network. It reconstructs a collection of extrusions that can be manually edited and combined to build models (n⊞, o⊞). Point2CAD [63] segments a point cloud into clusters corresponding to CAD faces, and each face is fitted with a geometric primitive. The surfaces are clipped by extend and intersect operations to build CAD models (m⊞). Furthermore, CAD-SIGNet [46] learns CAD visual-language representations by layer-wise cross-attention between the point cloud and CAD language embedding to recover sketch-and-extrusion operations (n⊞). However, similar to voxels, point clouds also suffer from low resolution due to reduced sampling rates aimed at minimizing computational costs.

Meshes Meshes are also characterized by irregular 3D representations, making them challenging to process directly with standard deep learning models. MeshCNN [36] converts irregular data to the regular structure at the edge level. Each mesh edge contains 5-dimensional relative features invariant to translation, rotation, and uniform scale. In mesh format, Lun et al. [64] proposed a multi-view convolutional network based on depth-map (RGB-D) to reconstruct 3D shapes. But, they need to train 12 models corresponding to each view. Zou et al. [113] presented a generative recurrent neural network named 3D-PRNN that constructs primitive-based 3D shapes from a single-depth image step-by-step. Moreover, in [61], Lin et al. explored how to model 3D shapes using reinforcement learning (RL). The RL model edits the meshes of the primitives to create 3D shapes.

In summary, existing deep learning approaches have demonstrated promising results in handling irregular data for tasks such as reverse engineering and 3D reconstruction. However, voxel, point cloud, and mesh representations are not well-suited for directly reconstructing 3D models from 2D drawings or sketches. Moreover, these representations have poor edit properties, which constrains their usefulness in downstream applications (n⊞).

4.3 Graph Representation of CAD Models

A CAD B-Rep model is composed of two fundamental components: topology and geometry. The topology defines the relationships and connectivity between elements, while the geometry specifies the precise shape and dimensions of each element. Graphs naturally support the representation of both topological relationships and associated geometric attributes. Several methods convert the B-Rep into a graph with topologies and features [5, 9, 85, 44, 45, 15, 98]. A graph is represented as $G = (V, L)$ where V is the set of nodes and L is the set of links between the nodes.

Graph Neural Networks (GNNs) are a class of deep learning methods that are directly applied to graphs, and that can solve the tasks of graph classification, node classification, link prediction, etc. Wu et al. [94] gave a comprehensive review of the graph neural networks. Here, we only briefly introduce GNNs. Inspired by the convolutional neural network, the graph convolutional networks generalize convolution operation from grid data to graph data. Gilmer et al. [26] proposed the message-passing neural network (MPNN) that treats graph convolutions as a message-passing process in which information can be passed directly from one node to another along edges. Several methods are introduced in [38, 16, 48]. Those GCN methods are more efficient, flexible, and have low computational complexity. Hamilton et al. [35] proposed a method well-suited to large graphs, and that represents node embeddings using only a subsample of neighboring nodes instead of the whole graph. Graph attention networks [81] adopt attention mechanisms to learn the relative weights between two connected nodes, and the more important nodes receive larger weights. Wang et al. [86] have proposed Dynamic Graph CNN (DGCNN). Unlike GCNs, the graph of DGCNN is not fixed. So, the graph is updated with the k-nearest neighbors at each layer using the current feature space. Moreover, Zhang et al. [106] proposed a pooling strategy named SortPooling, which performs pooling by rearranging nodes to a meaningful order. Differentiable pooling is proposed in [101], which can generate hierarchical representations of graphs by learning comprehensive node assignments.

With the advancement of graph neural networks (GNNs), numerous methods have emerged that leverage GNNs for CAD modeling tasks. The face-adjacent graph [5] converts B-Rep faces as graph nodes and B-rep edges as graph edges. Cao et al. [9] used a graph neural network to recognize the features of faces by converting B-Rep into a face-adjacency graph. But it can only work on B-Rep models with planar faces (l⊟, m⊟). Wang et al. [85] developed a GNN-based two-stage learning approach called DeepFeature to extract features from the graphs. Here, features with curved faces are considered. As an improvement, UV-Graph [44, 104] used a face-adjacency graph to store the topological information of the B-Rep model. Each surface and curve is represented with a regular UV-grid stored as the node and edge features in the graph. Jones et al. [45] proposed a structured B-Rep graph convolution network (SB-GCN) for assembly tasks. The graphs are built in four levels: vertex, edge, loop, and face. A hierarchical B-Rep graph representation is proposed in [15], with two levels: a B-Rep face adjacency graph level and a mesh face graph level. In [98], Xu et al. presented a novel B-Rep graph representation, referred to as a zone graph, where nodes correspond to solid regions (zones) and edges represent connected surfaces between them. Mahajan et al. [65] proposed a cross-modal retrieval framework that integrates a graph network and a PointNet to align 2D drawings and 3D CAD models using contrastive learning. Lambourne et al. have introduced B-RepNet [51], a neural network designed to segment the B-Rep data with a topological message-passing system. In some other interesting works, the authors attempted to use graphs and language to represent CAD models or sketches. For instance, a graph could define each sketch where nodes and edges denote parametric primitives and constraints, respectively. The parametric primitives (e.g., points, line segments, arcs, circles) and the constraints between them (e.g., coincidence, perpendicularity, mirror, tangent, orthogonal) are represented by different syntax and grammar [73, 67, 25, 91].

In summary, graph-based representations demonstrate strong capabilities for preserving both the topological and geometric information of the CAD models. They are widely used in a variety of CAD modeling tasks, including machining feature recognition [9, 85, 15], classification [44], assembly [45], and reconstruction [98, 104]. Graph construction strategies offer high flexibility, allowing node features and edge connections to be tailored according to the specific requirements of the CAD modeling tasks. Figure 3 shows several examples of CAD graph representations.

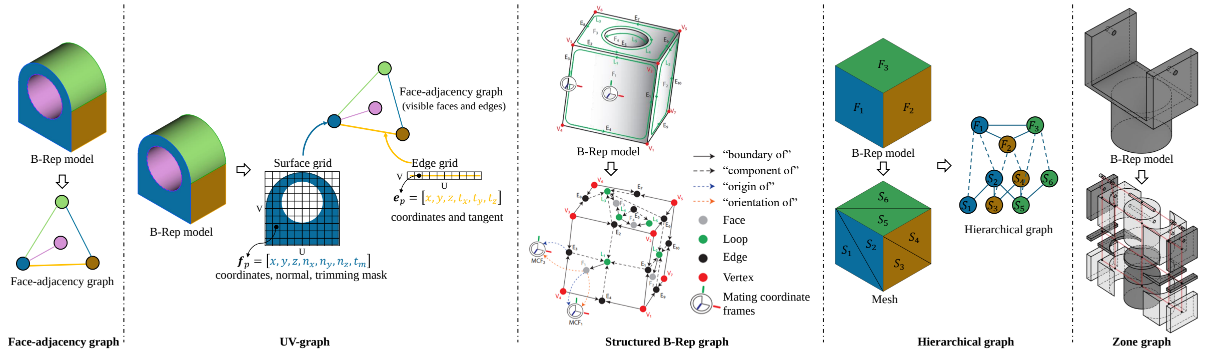

Figure 3: The CAD graph representations. Face-adjacency graph [5, 9]: the faces are nodes, and the links are the shared edges between two faces; UV-graph [44, 104]: based on face-adjacency graph and adding the UV-grid features to the nodes and links; Structured B-Rep graph [45]: building the graph between different level entities: vertices, edges, loops, and faces; Hierarchical graph [15]: building the graph in two levels, that is face and mesh grid. Zone graph [98]: zones are a set of solid regions, and the zone graph sets these zones as nodes. Note that only visible faces are shown in this figure, and the B-Rep models are manifold shapes.

4.4 Sequence-based Representation of CAD Models

Recently, there has been increasing interest in extending CAD command sequences-based approaches for 3D CAD modeling [91, 58, 92, 70]. A CAD model describes parametric modeling as a sequence of operations. For instance, the designer sketches a closed profile in 2D and then extrudes it into a 3D shape, further processed by boolean operations with another already created solid (n⊞).

2D to sequence Li et al. [57] proposed a sketching sequence-based CAD modeling method in which the sketches are parsed to CAD operations by CNN models, allowing an automatic transformation between the two domains. An improved approach is introduced in [58]. The sequence of sketches is converted into a sequence of CAD commands, and a transformer-based neural network segments the line groups of sketches (n⊞). Wang et al. [84] presented a face identification method that uses a transformer-based network to find edge loops from 2D wireframes, with the edge loops corresponding to the actual faces of the 3D shapes. Hu et al. [42] proposed PlankAssembly, which takes 2D line segments as input and outputs the shape sequences. But only line segments are supported to reconstruct cuboids (l⊟, m⊟, n⊟). Qin et al. [70] proposed a sequence-to-sequence model to generate 3D models from 2D vectorized drawings. Moreover, Zhang et al. [105] introduced a DQN-based agent with a parallel environment to reconstruct 3D parametric CAD models by selecting the sequences of loop-path pairs from 2D drawings. Willis et al. [91] proposed a reinforcement learning-based method to construct CAD models using the sketch and extrude sequence (n⊞). Two actions, sketch extrusion and face extrusion, are considered, while the state is represented as a face-adjacency graph.

Sequence generation Mo et al. proposed a hierarchical graph network, StructureNet [66], for 3D shape generation. The shapes are represented by their hierarchical decomposition into sequence parts, with geometric relationships between the parts (n⊟, o⊟). Wu et al. [92] proposed a CAD generative network based on a transformer. They represent a model as a sequence of CAD commands, which can be easily converted into a B-Rep (n⊞, o⊞). And several works for CAD generation based on CAD command sequences have been proposed recently [3, 99]. In [90], Willis et al. proposed a transformer-based engineering sketch generation network to address the engineering drawing generation and synthesis of the 3D solids by extruding the generated sketches. Moreover, in [43], the authors added the index lists to B-Rep models. Each face contains a list of indexes into edges, and each edge is a list of indices into vertices. The proposed transformer and pointer-based neural network can predict the B-Rep models' vertices, edges, and faces. Xu et al. [99] represent sketch-extrude operations as sequences of tokens. A transformer network learns over distribution in this representation and can be synthesized in a novel 3D model. They also presented a diffusion generative model based on the transformer, B-RepGen [97]. The B-RepGen represents a B-Rep model as a structured latent geometry in a hierarchical tree (n⊟). Very recently, another diffusion-based B-Rep generative model, BrepDiff [53], has been proposed, the B-Rep models are generated in a single stage.

Dumb CAD to sequence In [50, 60], the voxels are converted to sketch-and-extrude operations (n⊞). Point2Cyl [79] and CAD-SIGNet [46] approaches recover sketch-and-extrusion operations from point clouds (n⊞). Since the dumb B-Rep models have limited editing, several approaches attempt to convert the dumb B-Rep to parametric CAD models [104, 112, 107]. Firstly, the dumb B-Reps are represented in graphs, such as UV-graph, and then the transformer-based network predicts the CAD command sequences. The sequences can be parsed to parametric models in CAD modelers [104] (n⊞).

In summary, transformer-based approaches have achieved impressive results in processing sequential data. These methods represent 3D shapes as sequences, which can be converted into parametric B-Rep models with excellent editing capabilities through parameter adjustment (n⊞). However, the major limitation of sequence-based methods is their inability to support freeform curves (m⊟) such as Bézier and B-spline. To date, only simple shapes composed of lines, arcs, or circles have been handled.

4.5 Summary of Learning-based Methods

Deep learning methods typically demonstrate strong feature extraction capabilities for regular data such as voxels. Irregular 3D representations, including point clouds and meshes, are also well handled by neural networks. However, these representations offer limited support CAD modifications. While neural networks have been applied directly to B-Rep graphs for classification or segmentation tasks, there is still a lack of works focused on using 2D engineering drawings as input to reconstruct editable 3D CAD models. More attention is needed on developing deep learning approaches capable of generating editable, parametric CAD models directly from 2D engineering drawings, taking into account the following key considerations:

- CAD models can be represented with 3D voxels, point clouds, or meshes and processed with CNN or RNN. These methods make it hard to capture the details of CAD models because of the low resolution and inherent approximation of these discrete representations. Computational cost and system complexity are also usually high;

- Graphs can store both topological and geometric information. B-Rep models can easily converted to graphs, and GNNs are designed to address graph-related tasks. GNN-based methods have received impressive results for 3D modeling, such as machining feature recognition and 3D classification;
- Transformer-based networks achieved good performance for sequence data, which can be used to represent and process CAD models. However, freeform curves are not well supported;
- Reinforcement learning-based methods are promising for generating intermediate model outputs during CAD reconstruction, especially given the inherently sequential nature of 3D reconstruction from 2D drawings – an approach analogous to an agent navigating a step-by-step decision process.
- Most deep learning-based methods are focused on 3D generative and reverse engineering, and only a few works reconstruct 3D CAD models from 2D engineering drawings.

The characteristics of the deep learning-based approaches are summarized in Table 9. Since deep learning methods typically work on 3D models, the list of criteria used to compare the approaches is limited to the ones related to 3D model and Operation. For reverse engineering, such as reconstructing B-Rep models from point clouds, the method based on faces can support freeform surfaces, but the model is not editable (m⊞, n⊟) [33, 63]. The methods focusing on sequences, such as converting the dumb B-Rep to sketch-extrude operations, have good editable properties (n⊞). However, only regular surfaces and extrusion operations are well supported [112, 107].

Table 9: Summary of the learning-based methods according to the evaluation criteria.

Criteria	3D model				Operation		
	Face			Edit	-		
	k	l	m	n	o	p	q
Lambourne et al. [50]	⊞	⊞	⊟	⊞	⊞	⊟	⊟
Guo et al. [33]	⊞	⊞	⊞	⊟	⊟	⊟	⊟
Uy et al. [79]	⊞	⊞	⊟	⊞	⊞	⊟	⊟
Liu et al. [63]	⊞	⊞	⊞	⊟	⊟	⊟	⊟
Lun et al. [64]	⊞	⊞	⊞	⊟	⊟	⊟	⊟
Lin et al. [61]	⊞	⊞	⊞	⊟	⊟	⊟	⊟
Xu et al. [98]	⊞	⊞	⊟	⊞	⊞	⊟	⊟
Willis et al. [91]	⊞	⊞	⊟	⊞	⊞	⊟	⊟
Wu et al. [92]	⊞	⊞	⊟	⊞	⊞	⊟	⊟
Wang et al. [84]	⊞	⊞	⊟	⊟	⊟	⊟	⊟
Hu et al. [42]	⊞	⊟	⊟	⊟	⊟	⊟	⊟
Jayaraman et al. [43]	⊞	⊞	⊟	⊟	⊟	⊟	⊟
Xu et al. [99]	⊞	⊞	⊟	⊞	⊞	⊟	⊟
Zhou et al. [112]	⊞	⊞	⊟	⊞	⊞	⊞	⊟
Zhang et al. [107]	⊞	⊞	⊟	⊞	⊞	⊟	⊟

Table 10 summarizes the learning-based methods across five aspects: input data, input and output representations, underlying method, and the availability of the code.

It clearly appears that GNNs and Transformers show promising results for CAD modeling. The strong performance of this type of model can be attributed to two aspects: data representation and network performance. The excellent performance of Transformers and Graph Neural Networks has been verified, as detailed in section 4.3 and section 4.4. Here, we explain the other aspect: data representation. Graph representation effectively preserves the topological information of CAD models, such as relationships among entities like vertices, lines, and surfaces, stored through face-adjacency and/or vertex-adjacency graphs. Parametric models are created sequentially through sketches and CAD operation representations, such as extrusion, revolution, sweep, and loft. Each operation is recorded in the history of the parametric model, thus facilitating Transformer learning. However, graph representation also has limitations, such as the reconstructed model not being closed and failing to meet industrial-grade watertightness requirements. For Transformers and sequences, CAD sequences are still mainly limited to sketches and extrusions, more complex CAD operations need to be added. Furthermore, the design history, annotations, and constraints of a parametric CAD model can serve as conditions for the generative model to generate new shapes. In the end, graph and sequence representations can be converted to parametric CAD models with editable parameters, making them well-suited for 2D-to-3D reconstruction tasks.

Indeed, based on large volumes of data, learning-based approaches can better extract features from CAD models and provide more efficient representations, such as neural network models proposed for voxels, point clouds, and graphs. However, in engineering implementation, learning-based methods often struggle to provide accurate reconstructions. For example, in industrial applications, learning-based algorithms may fail to reconstruct a perfect arc of circle. Rule-based approaches significantly outperform learning-based methods in terms of data accuracy and interpretability. (section 3) Therefore, hybrid algorithms that combine both have recently been proposed, such as the Point2CAD[63] method. One approach involves first segmenting the point cloud using networks to obtain rough surfaces [75, 100], and then reconstructing the B-Rep using a fitting algorithm. In another example, the authors first convert the dumb B-Rep model to a CAD sequence and then use a feature-matching algorithm to rectify the reconstructed parametric CAD models [104].

Recently, reconstructing 2D drawings into 3D models has found applications in industry. For example, Dassault Systémes demonstrated its reconstruction process at the 3D Experience World 2026.[12] has released its reconstruction software. The reconstruction process in the application can be summarized in the following steps:

- Drawing processing: Segmenting the drawing view using analytical algorithms to obtain independent views and performing noise reduction.
- Drawing analysis: Identifying entities in the drawing, such as line segments and circles, using object detection algorithms. Detected entities are then vectorized.
- Relationship building: Matching entities using rule-based or learning algorithms to determine valid CAD operations, such as reconstructing surfaces or using CAD commands, to obtain the 3D object.
- Reconstructing the 3D model: Combining and optimizing the reconstructed entities to obtain the final model.

5 CAD DATABASES

Over the past several years, the number of CAD datasets has significantly increased. These datasets generally fall into two categories: human-designed CAD models and synthetically generated ones. Table 11 summarizes

[1] https://www.3dexperienceworld.com/ Additionally, Spare Parts 3D
[2] https://spare-parts-3d.com/

Table 10: Summary of the learning-based approaches considering the input/output representations, the reconstruction methods and the availability of the code.

References	Input	Input representation	Output representation	Methods	Code
Zhang et al. [109]	voxels	voxels	labels	3D-CNNs	√
Lambourne et al. [50]	voxels	voxels	image profiles	CNNs, 3D-CNNs	-
Li et al. [60]	voxels	voxels	sketch-extrude sequences	3D-CNNs, MLP	√
Wu et al. [93]	3D Parts	vectors	voxels	3D-CNNs, RNNs	√
Guo et al. [33]	point clouds	point clouds	dumb B-Rep	CNNs, Transformers	√
Uy et al. [79]	point clouds	point clouds	sketch-extrude sequences	PointNet++	√
Liu et al. [63]	point clouds	point clouds	dumb B-Rep	DGCNN	√
Lun et al. [64]	2D sketches	images	meshes	CNNs	√
Zou et al. [113]	depth images	depth images	voxels	LSTM	√
Lin et al. [61]	primitives	vectors	meshes	CNNs, RL	√
Cao et al. [9]	dumb B-Rep	graphs	labels	GNNs	√
Jayaraman et al. [44]	dumb B-Rep	UV-graphs	labels	CNNs, GNNs	√
Colligan et al. [15]	dumb B-Rep	hierarchical graphs	labels	GNNs	√
Xu et al. [98]	dumb B-Rep	zone graphs	sketch-extrude sequences	GCNs, PointNet	√
Willis et al. [91]	sketch-extrude sequences	vectors, graphs	actions sequences	GCNs, RL	√
Li et al. [58, 57]	freehand sketches	images	CAD commands	CNNs, Transformers	√
Wu et al. [92]	sketch-extrude sequences	sketch-extrude sequences	sketch-extrude sequences	Transformers	√
Wang et al. [84]	2D wireframes	co-edges sequences	faces sequences	Transformers	√
Hu et al. [42]	2D drawings	2D edge vector	sequences	Transformers	√
Zhang et al. [105]	2D drawings	loop-path pair	sequence of loop-path	CNNs, RL	-
Mo et al. [66]	3D parts	point clouds, graphs	point clouds, voxels	GNNs	√
Jayaraman et al. [43]	entities sequences	entities sequences	Indexed B-Rep	Transformers	-
Xu et al. [99]	entities sequences	entities sequences	sketch-extrude sequences	codebooks	√
Xu et al. [97]	dumb B-Rep	hierarchical trees	hierarchical trees	Transformers	√
Zhou et al. [112]	dumb B-Rep	graphs	command sequences	CNNs, GNNs, Transformers	-
Zhang et al. [107]	dumb B-Rep	graphs	primitive sequences	CNNs, GNNs, Transformers	√

popular CAD datasets used for tasks such as model reconstruction, generation and classification. For each dataset, we report the number of samples (Samples), the types of model representations (Data types), whether it contains human-annotated labels (Labels), and whether it consists of real-world, human-designed models (Human-designed). The representation types (Data types) include B-Rep, mesh, 2D drawings or sketches (2D), and CAD modeling sequences (Seq.).

Table 11: The open source CAD model datasets.

Dataset	Samples	Data Types				Label	Human-designed
		B-Rep	Mesh	2D.	Seq.		
Fusion360 [91]	8625	⊕	⊕	⊕	⊕	⊖	⊕
Automatic3D [103]	2981	⊕	⊕	⊕	⊕	⊖	⊕
ABC [49]	1M+	⊕	⊕	⊖	⊖	⊖	⊕
DeepCAD [92]	178238	⊖	⊖	⊕	⊕	⊖	⊕
SketchGraphs [73]	15M+	⊖	⊖	⊕	⊕	⊖	⊕
Thingi10k [111]	10000	⊖	⊕	⊖	⊖	⊕	⊕
MCB [47]	58696	⊖	⊕	⊖	⊖	⊕	⊕
ModelNet [95]	127915	⊖	⊕	⊖	⊖	⊕	/
ShapeNet [12]	3M+	⊖	⊕	⊖	⊖	⊕	/
FabWave [4]	100000	⊕	⊖	⊖	⊖	⊖	/
CADParser [112]	10000	⊕	⊖	⊖	⊕	⊖	/
MFCAD [9]	15486	⊕	⊖	⊖	⊖	⊖	⊖
SolidLetters [44]	96861	⊕	⊖	⊖	⊖	⊖	⊖
B-Rep2seq [107]	1M	⊕	⊖	⊖	⊕	⊖	⊖

∗ ⊕ :with this column data; ⊖ :without this column data;
/ :uncertain (data collected from the Internet or CAD modelers).

Fusion 360 Gallery [91] is a human-designed CAD dataset that contains 2D and 3D geometry data derived from parametric CAD models. The reconstruction dataset includes 8625 designs. The 'sketch and extrude' operation sequence is provided for each case to rebuild the final model. Moreover, the CAD models are also represented in B-Rep and mesh formats as a reference for reconstruction tasks. However, this dataset only contains sketch-extrude operations. Automatic3D [103] is built based on the Fusion 360 Gallery to reconstruct 3D CAD models from 2D orthographic drawings. The 2D drawings are generated from original 3D models by the parallel projection rules. The generated 2D drawings are stored in $.svg$ files and are composed of three cleaned orthographic views without annotations. These datasets are usually used for 3D CAD model reconstruction tasks.

ABC [49] dataset introduces over one million real-world geometric models, each with B-Rep and mesh formats. The ABC dataset is built for geometric deep learning, each associated with accurate ground truth information on the decomposition into patches, explicit sharp feature annotations, and analytic differential properties. In addition, the dataset provides a link to Onshape's [1] original CAD design for each CAD model. Based on ABC, DeepCAD [92] is built to train generative networks. They use Onshape domain-specific language to parse CAD operations and parameters in that design. These CAD operations and parameters are converted into CAD sequences to describe CAD designs. However, duplicates and simple models are not filtered in the DeepCAD dataset. Moreover, the parsed CAD sequences are incorrect for a number of models. SketchGraphs [73] is another CAD sequences dataset that contains about 15 million sketches extracted from Onshape CAD models. Each sketch is represented as a geometric constraint graph where edges denote designer-imposed geometric relationships between primitives associated with the graph nodes. These CAD sequence datasets are used for generation tasks and adapt to RNN and Transformer networks.

Generally, CAD datasets with human-annotated labels are used for shape classification tasks. Thingi10k [111] contains 10000 mesh-based distributed across 72 categories. This dataset is also created for shape analysis and optimization. MCB [47] is a large-scale dataset of 3D objects of mechanical components. It has a total number of 58,696 mechanical components with 68 classes. These 3D models are collected from the CAD modelers. ModelNet [95] contains 127915 3D CAD models in mesh format from 662 categories. It has two subsets ModelNet10 and ModelNet40. ModelNet10 includes 4899 models from 10 categories, and ModelNet40 contains 12311 models from 40 categories. And the authors provided the labels. ShapeNet [12] is a well-organized and richly annotated large-scale CAD dataset. The dataset contains 3M+ models for 4K+ categories. ShapeNet was created to research computer graphics and computer vision. ShapeNetCore and ShapeNetSem are the subsets of ShapeNet with manually verified category and alignment annotations. ShapeNetCore builds with single clean 3D models, covering 55 common object categories and about 51,300 unique 3D models. ShapeNetSem is a smaller, richer annotated subset of 12,000 models spread over a broader set of 270 categories. These datasets are usually without B-Rep models.

Machining feature recognition has been widely studied in analyzing B-Rep models. FabWave [4] is created to spur innovations in product design and manufacturing research. It is a feature-rich part dataset represented in B-Rep format. FabWave collected about 100000 CAD models from designers and the Internet. CAD-Parser [112] is also collected from the Internet and from the CAD modeler SolidWorks. It contains about 10000 CAD models with design history. The design information is stored in the sequences. Furthermore, SolidLetters [44] and MFCAD [9] are the synthetic datasets for machining feature recognition tasks. SolidLetters has 96k+ CAD models created by extruding and filleting fonts. It has class labels and style labels for each shape. MFCAD focuses on machining feature recognition tasks from planar B-Rep CAD models. The dataset of 15,486 CAD shapes was created using 24 machining features. The faces of B-Rep models were classified as different machining feature classes. Another synthetic B-Rep model dataset is B-Rep2seq [107], which contains randomly generated CAD models with the modeling information, including principal primitives (cuboids, prisms, cylinders, cones, and spheres) and detail features (slots, semi-circular slots, through-holes, steps, fillets, chamfers, etc.). The CAD models in these datasets lack semantic annotations and are not designed for assembly tasks.

In summary, open-source CAD datasets support a wide range of tasks, including 3D CAD model reconstruction, generative CAD modeling, machining feature recognition, and model classification. Datasets containing 2D drawings or sketches are particularly relevant for 3D reconstruction tasks and are typically associated with B-Rep or mesh 3D models. Sequence-based datasets are the most popular CAD representation for generative tasks, while classification tasks require CAD models with human-annotated labels. For machining feature recognition, synthetic B-Rep models are often generated with annotations at the face or edge levels. These datasets play an essential role in advancing geometric processing research. Figure 4 shows examples from several of these open-source datasets.

6 EVALUATION METRICS

To evaluate the performance of 3D CAD model processing methods, the evaluation metrics can be broadly categorized into two types. These metrics assess the quality of reconstruction or generation, either by comparing two datasets or two CAD models.

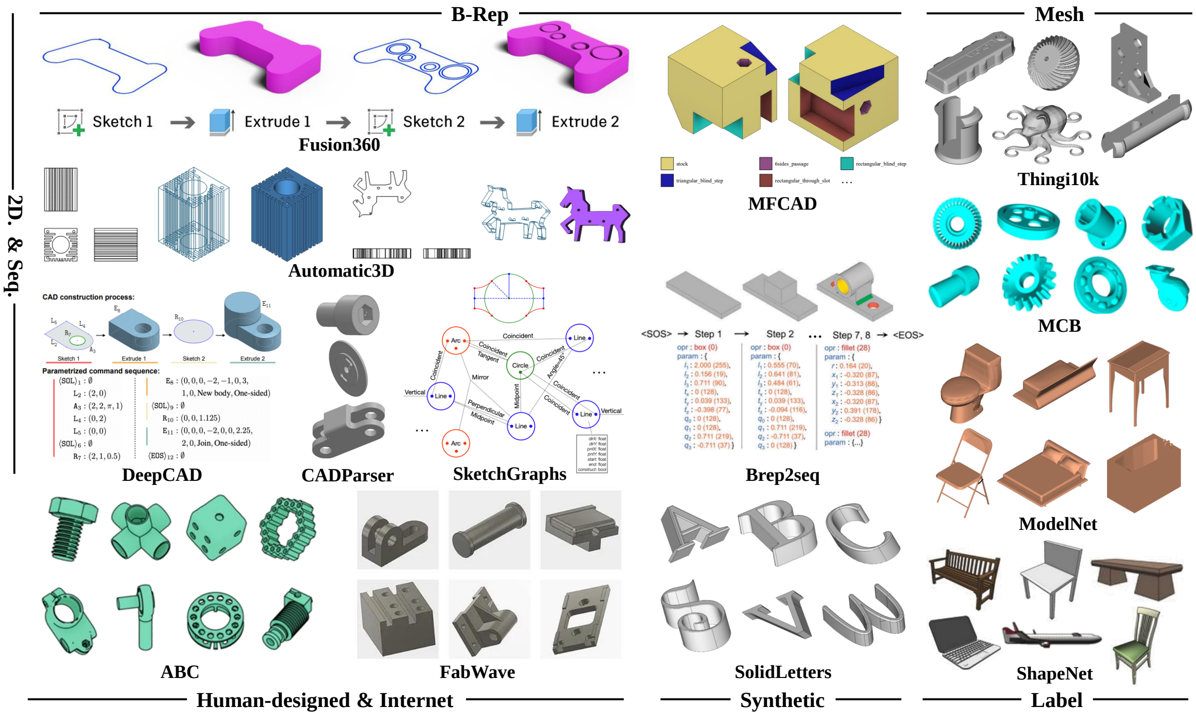

Figure 4: Examples of open-source CAD datasets, with their main characteristics reported in Table 11.

6.1 Evaluation Metrics Between two Datasets

The first category of evaluation metrics helps comparing a set of target CAD models with a set of generated CAD models. Three metrics can be distinguished: Coverage (COV), Minimum Matching Distance (MMD), and Jensen-Shannon Divergence (JSD) [92, 99, 97, 2, 96]. These metrics are usually used to validate the similarity for generative tasks.

COV This metric computes the coverage between two sets by computing the fraction of models in the reference set that are matched by at least one model in the generated set. It is defined as:

$$\mathrm{COV}(\mathcal{S},\mathcal{G}) = \frac{|\{\arg\min_{Y\in\mathcal{S}} d(X,Y) \mid X \in \mathcal{G}\}|}{|\mathcal{S}|} \tag{1}$$

where $\mathcal{S}$ is the reference set, $\mathcal{G}$ is the generated set, X, Y are the models from $\mathcal{S}$ and $\mathcal{G}$, respectively. d denotes the distance between X and Y, which is used to find the closest model Y of model X based on Chamfer Distance (d_{CD}) or Earth Mover's Distance (d_{EMD}) [2].

$$d_{CD}\left(X_p, Y_p\right) = \sum_{x\in X_p} \min_{y\in Y_p} \|x-y\|_2^2 + \sum_{y\in Y_p} \min_{x\in X_p} \|x-y\|_2^2 \tag{2}$$

where $X_p \in \mathbb{R}^{N\times 3}$ and $Y_p \in \mathbb{R}^{N\times 3}$ are sampled points from X and Y, respectively, and $(x,y) \in \mathbb{R}^3 \times \mathbb{R}^3$ are the 3D points.

$$d_{EMD}\left(X_p, Y_p\right) = \min_{\phi: X_p \rightarrow Y_p} \sum_{x \in X_p} \|x - \phi(x)\|_2 \tag{3}$$

where ϕ is a bijective function that maps each element $x \in X_p$ to its closest counterpart $y = \phi(x) \in Y_p$.

In the end, the COV metric measures the diversity of generated shapes. If the generated models only match a few reference models, that leads to low COV scores. Otherwise, the generated set captures all modes of reference set with good fidelity when COV is large.

MMD This metric measures the average minimum matching distance between the reference and generated sets. The distance to its nearest neighbor in the generated set $\mathcal{G}$ is computed for each shape in the reference set $\mathcal{S}$. It is defined as:

$$\mathrm{MMD}(\mathcal{S}, \mathcal{G}) = \frac{1}{|\mathcal{S}|} \sum_{Y \in \mathcal{S}} \min_{X \in \mathcal{G}} d(X, Y) \tag{4}$$

where d is the distance function set as d_{CD} or d_{EMD}. The MMD value will be small when the generated set $\mathcal{G}$ captures all modes of $\mathcal{S}$ with good quality.

JSD This metric evaluates the similarity between two probability distributions. Here, it measures the similarity between the generated set $\mathcal{G}$ and the reference set $\mathcal{S}$ by computing the marginal point distributions:

$$\mathrm{JSD}\left(P_{\mathcal{S}}, P_{\mathcal{G}}\right) = \frac{1}{2} D_{\mathrm{KL}}\left(P_{\mathcal{S}} \| M\right) + \frac{1}{2} D_{\mathrm{KL}}\left(P_{\mathcal{G}} \| M\right) \tag{5}$$

where $M = \frac{1}{2}\left(P_{\mathcal{S}} + P_{\mathcal{G}}\right)$, and D_{KL} is the standard KL divergence. $P_{\mathcal{S}}$ and $P_{\mathcal{G}}$ are marginal distributions of points in the reference and generated sets. A smaller JSD value indicates a higher similarity between the two sets.

Generally, these metrics are used for generative tasks to measure the performance of the generative networks. [92, 99, 97, 2, 96]

6.2 Evaluation Metrics Between two Models

The second category helps comparing a reconstructed or generated CAD model with a target one, by means of several metrics: Symmetric difference (SD), Intersection over Union (IOU), Recall, Precision, and F-score.

SD This metric is a commonly used binary Boolean operation that combines two or more shapes and checks whether an element e lies in either shape but not in their intersection. It is computed and averaged as follows:

$$SD = (X \backslash Y) \cup (Y \backslash X) \tag{6}$$

where X and Y are the target and reconstructed CAD models, respectively, scaled to the same size. Figures 5 (a) and (b) illustrate the difference operations $(X \backslash Y)$ and $(Y \backslash X)$, respectively. The operation $(X \backslash Y)$ represents the set of elements that are in X but not in Y, while $(Y \backslash X)$ applies the same operation with the roles of $(X \backslash Y)$ reversed. A value $SD = 1$ corresponds to a configuration where the target and reconstructed CAD models perfectly match.

IOU It measures the overlap between the reconstructed or generated model and the target model, helping to quantify how well the output of an approach aligns with the original model. Here, the 3D CAD models are scaled into the same size and voxelized to compute the volumes [50, 51, 91]. It is defined as:

$$IOU(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{7}$$

where $| \cdot \cap \cdot |$ denotes the volume of intersection, $| \cdot \cup \cdot |$ denotes the volume of union, as shown in Figures 5 (c) and (d). A higher value of IOU indicates a greater overlap between the two 3D CAD models.

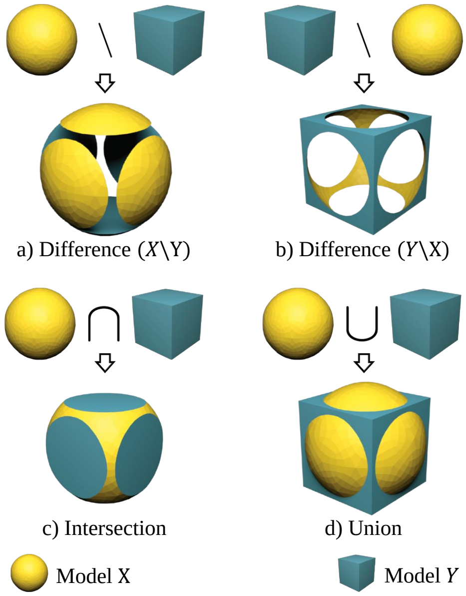

Figure 5: Intersection, Union, and Difference operations between two models.

Precision, Recall, and F-score Based on the IOU metric, the precision, recall, and F-score can be computed by counting the matched edges or faces between the reconstructed and target CAD models [103, 73, 84]. In [103], the authors evaluate the performance of the 3D reconstruction by computing precision, recall, and F-score at wireframe and B-Rep levels. The precision metric evaluates how many of the correct reconstructed edges/faces are in all reconstructed edges/faces. The recall measures how many edges/faces are correctly reconstructed in all original edges/faces. The F-Score combines precision and recall into a single metric using their harmonic mean.

At the level of the wireframes, the $| W_i |$ edges of a reconstructed wireframe W_i are compared to the $| W_i^* |$ edges of the original wireframe W_i^*, and the number of edges that match $| W_i \cap W_i^* |$ is computed. An edge e matches another edge e^* if and only if the coordinates of their vertices are equal within a given tolerance. Thus, the average precision, recall, and F-score for the wireframes are respectively computed as follows:

$$
\begin{cases}
p_w = \dfrac{1}{N_s}\displaystyle\sum_{i=1}^{N_s} \mid W_i \cap W_i^* \mid / \mid W_i \mid \\
r_w = \dfrac{1}{N_s}\displaystyle\sum_{i=1}^{N_s} \mid W_i \cap W_i^* \mid / \mid W_i^* \mid \\
FS_w = \dfrac{2p_w r_w}{p_w + r_w}
\end{cases} \tag{8}
$$

with N_s the number of models wireframes compared. At the level of the B-Rep models, the $\mid F_i \mid$ faces of a reconstructed B-Rep F_i are compared to the $\mid F_i^* \mid$ faces of the original B-Rep F_i^*, and the number of faces that match $\mid F_i \cap F_i^* \mid$ is computed. A face f matches another face f^* if and only if the two sets of edges defining the two faces are equal within a given tolerance. Thus, the average precision, recall, and F-score for the B-Rep models are respectively computed as follows:

$$
\begin{cases}
p_f = \dfrac{1}{N_s}\displaystyle\sum_{i=1}^{N_s} \mid F_i \cap F_i^* \mid / \mid F_i \mid \\
r_f = \dfrac{1}{N_s}\displaystyle\sum_{i=1}^{N_s} \mid F_i \cap F_i^* \mid / \mid F_i^* \mid \\
FS_f = \dfrac{2p_f r_f}{p_f + r_f}
\end{cases} \tag{9}
$$

with N_s the number of model compared. Moreover, the larger precision, recall, and F-score values mean a well-reconstructed CAD model.

In short, the metrics COV, MMD, and JSD are used in generation tasks, and the 3D points need to be sampled from the 3D CAD models. The metrics SD, IOU, Precision, Recall, and F-score are used to evaluate the reconstruction performance between two CAD models directly. In addition, the evaluation metrics Precision, Recall, and F-score are also used in CAD classification and recognition tasks. The CAD tasks and the corresponding metrics are summarized in Table 12.

Table 12: Evaluation metrics for CAD tasks.

CAD tasks	Metrics
Classification	Precision, Recall, and F-score
Generation	COV, MMD, and JSD
Reconstruction	SD, IOU, Precision, Recall, and F-score

7 DISCUSSION AND FUTURE CHALLANGES

In this survey, we reviewed rule-based and learning-based methods for reconstructing parametric CAD models from 2D drawings, with a focus on input types, data representations, and algorithmic strategies. We categorized existing approaches into CSG-based, B-Rep-based, and learning-based methods, and provided a comparative analysis using well-defined evaluation criteria. Various evaluation metrics have also been introduced to facilitate fair comparison of existing methods on a common basis. CAD datasets were also presented to help researchers select appropriate benchmarks based on task type and data format.

Rule-based methods, including CSG and B-Rep approaches, face several limitations. In both cases, reconstructing 3D primitives from 2D geometric entities is challenging, and the resulting models typically lack editable parameters. These methods also generate dumb models that lose operation history and design intent. Additionally, they often struggle to handle complex geometries, particularly curved and freeform surfaces.

Learning-based methods, while still emerging, show promising results for 3D modeling from 2D inputs. These methods often use approximate 3D representations, e.g. voxels, point clouds, or meshes, but these formats tend to lose geometric fidelity when compared to CAD models. In contrast, graph-based representations retain both shape and topological information, making them suitable for learning-based CAD modeling tasks such as feature recognition and classification. Graph neural networks (GNNs) perform well when paired with such representations.

Another promising approach is to represent CAD models using command sequences, such as sequences of sketch and extrude operations. These operations are parameterized with vertices and edges, allowing for easy parsing into editable, feature-based parametric models. Transformer-based networks, which are effective at handling sequential data, have been applied with success. However, current sequence-based methods are mostly limited to basic operations like sketch and extrude. Extending these models to support a richer set of operations (e.g. revolve, chamfer, fillet, and boolean combinations) remains an important direction for future research.

Looking ahead, several key challenges and research opportunities remain:

- *Semantic understanding*: Incorporating higher-level semantic understanding into the modeling process is essential. This includes inferring the designer's intent and assigning meaningful labels to CAD operations, enabling reconstructions that better align with human design reasoning, and which can be better exploited all along the Product Development Process (PDP).
- *Enhanced use of annotations and multi-view inputs*: 2D engineering drawings often include annotations and additional views beyond the standard orthographic projections. Leveraging this additional context through multi-view learning could significantly enhance reconstruction accuracy and aid in resolving ambiguous configurations.
- *Handling incomplete or noisy drawings*: Most existing approaches assume that input 2D drawings are complete and unambiguous, containing all the information needed for accurate reconstruction. In practice, however, real-world engineering drawings often suffer from missing information, occluded features, or noise due to their scan. Addressing these issues would require robust reconstruction models capable of inferring incomplete information and handling uncertainty through context-aware or probabilistic approaches.
- *Assembly-level reconstruction*: Current methods largely focus on single-part reconstruction. However, many real-world and industrial CAD drawings represent assemblies of multiple components. These drawings are often incomplete or ambiguous, requiring reasoning and contextual knowledge to infer undefined shapes or missing views. Being able to directly reconstruct assemblies from 2D drawings would represent a major advancement, transforming workflows in domains such as engineering design and building information modeling.
- *Integration of standard component libraries*: Incorporating prior knowledge from databases of standard components (e.g. screws, bearings, and nuts) could enhance model accuracy and reduce ambiguity. This is particularly true in industrial contexts where the use of well-defined and possibly repeated standard parts is common practice.

8 CONCLUSIONS AND FUTURE WORK

This survey presented a thorough analysis of existing methods for reconstructing parametric CAD models from 2D drawings. We discussed both rule-based and learning-based approaches in terms of input representations, modeling techniques, and algorithmic strategies. While rule-based methods suffer from limited editability and lack of support for complex geometries, learning-based approaches, particularly those using graph and sequence representations, offer promising alternatives.

Graph representations preserve topological and geometric structure, enabling effective use of GNNs in CAD-related tasks. Sequence-based approaches, especially those leveraging transformer architectures, facilitate parametric reconstruction by learning from CAD command sequences. Nonetheless, current implementations are limited in operational scope and need to evolve to support a broader range of design actions.

In future work, we plan to focus on using deep learning methods, in particular GNNs and transformers, to identify geometric entities from 2D engineering drawings. Reconstructed models will be represented as CAD sequences, which can then be converted into fully parametric CAD models. Addressing the open challenges related to semantic reasoning, multi-view integration, assembly modeling, and standard component recognition will be crucial for advancing the field.

REFERENCES

[1] Onshape. https://www.onshape.com/.

[2] Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; Guibas, L.: Learning representations and generative models for 3d point clouds. In International conference on machine learning, 40–49. PMLR, 2018.

[3] Alam, M.F.; Ahmed, F.: Gencad: Image-conditioned computer-aided design generation with transformer-based contrastive representation and diffusion priors. arXiv preprint arXiv:2409.16294, 2024.

[4] Angrish, A.; Craver, B.; Starly, B.: fabsearch: A 3d cad model-based search engine for sourcing manufacturing services. Journal of Computing and Information Science in Engineering, 19(4), 041006, 2019.

[5] Ansaldi, S.; De Floriani, L.; Falcidieno, B.: Geometric modeling of solid objects by using a face adjacency graph representation. ACM SIGGRAPH Computer Graphics, 19(3), 131–139, 1985.

[6] Bénière, R.; Subsol, G.; Gesquière, G.; Breton, F.L.; Puech, W.: Topology reconstruction for b-rep modeling from 3d mesh in reverse engineering applications. In Electronic Imaging, 2012.

[7] Bnire, R.; Subsol, G.; Gesquire, G.; Le Breton, F.; Puech, W.: A comprehensive process of reverse engineering from 3d meshes to cad models. Computer-Aided Design, 45(11), 1382–1393, 2013.

[8] Camba, J.D.; Company, P.; Naya, F.: Sketch-based modeling in mechanical engineering design: Current status and opportunities. Computer-Aided Design, 150, 103283, 2022.

[9] 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. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 84003, V11AT11A003. American Society of Mechanical Engineers, 2020.

[10] Carfagni, M.; Furferi, R.; Governi, L.; Volpe, Y.: 3d geometry reconstruction from orthographic views: An improved method exploiting shading information. Computers in Industry, 92, 137–151, 2017.

[11] Çayıroglu, İ.; Çavusoglu, A.; Çelik, V.: A new method for extracting 3d solid models of objects using 2d technical drawings. Mathematical and Computational Applications, 12(1), 31–39, 2007.

[12] Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012, 2015.

[13] Chen, K.Z.; Feng, X.A.: Solid model reconstruction from engineering paper drawings using genetic algorithms. Computer-Aided Design, 35(13), 1235–1248, 2003.

[14] Çıçek, A.; Gülesın, M.: Reconstruction of 3d models from 2d orthographic views using solid extrusion and revolution. Journal of materials processing technology, 152(3), 291–298, 2004.

[15] 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, 103226, 2022.

[16] Defferrard, M.; Bresson, X.; Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems, 29, 2016.

[17] Dimri, J.; Gurumoorthy, B.: Handling sectional views in volume-based approach to automatically construct 3d solid from 2d views. Computer-Aided Design, 37(5), 485–495, 2005.

[18] Ding, F.; Zhang, H.; Wen, Y.: Converting sectional views to three orthographic views to reconstruct 3d models. Computer-Aided Design and Applications, 8(4), 571–582, 2011.

[19] Egiazarian, V.; Voynov, O.; Artemov, A.; Volkhonskiy, D.; Safin, A.; Taktasheva, M.; Zorin, D.; Burnaev, E.: Deep vectorization of technical drawings. In European conference on computer vision, 582–598. Springer, 2020.

[20] Elyan, E.; Jamieson, L.; Ali-Gombe, A.: Deep learning for symbols detection and classification in engineering drawings. Neural networks, 129, 91–102, 2020.

[21] Fahiem, M.A.; Haq, S.A.; Saleemi, F.: A review of 3d reconstruction techniques from 2d orthographic line drawings. Geometric Modeling and Imaging (GMAI'07), 60–66, 2007.

[22] Fayolle, P.A.; Friedrich, M.: A survey of methods for converting unstructured data to csg models. Computer-Aided Design, 168, 103655, 2024.

[23] Feist, S.; Jacques de Sousa, L.; Sanhudo, L.; Poças Martins, J.: Automatic reconstruction of 3d models from 2d drawings: A state-of-the-art review. Eng, 5(2), 784–800, 2024.

[24] Furferi, R.; Governi, L.; Palai, M.; Volpe, Y.: From 2d orthographic views to 3d pseudo-wireframe: An automatic procedure. International Journal of Computer Applications, 975, 8887, 2010.

[25] Ganin, Y.; Bartunov, S.; Li, Y.; Keller, E.; Saliceti, S.: Computer-aided design as language. Advances in Neural Information Processing Systems, 34, 5885–5897, 2021.

[26] Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E.: Neural message passing for quantum chemistry. In International conference on machine learning, 1263–1272. PMLR, 2017.

[27] Gong, J.H.; Zhang, G.F.; Zhang, H.; Sun, J.G.: Reconstruction of 3d curvilinear wire-frame from three orthographic views. Computers & Graphics, 30(2), 213–224, 2006.

[28] Gong, J.H.; Zhang, H.; Zhang, G.F.; Sun, J.G.: Solid reconstruction using recognition of quadric surfaces from orthographic views. Computer-Aided Design, 38(8), 821–835, 2006.

[29] Gong, J.H.; Zhang, H.; Zhang, Y.W.; Sun, J.G.: Converting hybrid wire-frames to b-rep models. In Proceedings of the 2007 ACM symposium on Solid and physical modeling, 283–289, 2007.

[30] Gorgani, H.H.; Neyestanaki, I.M.S.; Pak, A.J.: Solid reconstruction from two orthographic views using extrusion and comparative projections. Journal of Engineering and Applied Sciences, 12(7), 1938–1945, 2017.

[31] Gorgani, H.H.; Pak, A.J.; Sadeghi, S.: 3d model reconstruction from two orthographic views using fuzzy surface analysis. European Journal of Sustainable Development Research, 3(2), em0081, 2019.

[32] Governi, L.; Furferi, R.; Palai, M.; Volpe, Y.: 3d geometry reconstruction from orthographic views: A method based on 3d image processing and data fitting. Computers in industry, 64(9), 1290–1300, 2013.

[33] Guo, H.; Liu, S.; Pan, H.; Liu, Y.; Tong, X.; Guo, B.: Complexgen: Cad reconstruction by b-rep chain complex generation. ACM Transactions on Graphics (TOG), 41(4), 1–18, 2022.

[34] Haghshenas Gorgani, H.; Jahantigh Pak, A.: A genetic algorithm based optimization method in 3d solid reconstruction from 2d multi-view engineering drawings. Journal of Computational Applied Mechanics, 49(1), 161–170, 2018.

[35] Hamilton, W.; Ying, Z.; Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems, 30, 2017.

[36] Hanocka, R.; Hertz, A.; Fish, N.; Giryes, R.; Fleishman, S.; Cohen-Or, D.: Meshcnn: a network with an edge. ACM Transactions on Graphics (ToG), 38(4), 1–12, 2019.

[37] Harish, A.B.; Prasad, A.R.: Automated 3d solid reconstruction from 2d cad using opencv. arXiv preprint arXiv:2101.04248, 2021.

[38] Henaff, M.; Bruna, J.; LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163, 2015.

[39] Hoang, L.: 3d solid reconstruction from 2d orthographic views. In Mixed Reality and Three-Dimensional Computer Graphics, 77. IntechOpen, 2020.

[40] Hoang, L.; Tien, L.B.: A flexible solid 3d model reconstruction system for mechanical cad/cam systems. Journal of the Korean Society for Precision Engineering, 36(8), 753–759, 2019.

[41] Hu, H.; Zhang, C.; Liang, Y.: Detection of surface roughness of mechanical drawings with deep learning. Journal of Mechanical Science and Technology, 35, 5541–5549, 2021.

[42] Hu, W.; Zheng, J.; Zhang, Z.; Yuan, X.; Yin, J.; Zhou, Z.: Plankassembly: Robust 3d reconstruction from three orthographic views with learnt shape programs. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 18495–18505, 2023.

[43] Jayaraman, P.K.; Lambourne, J.G.; Desai, N.; Willis, K.D.; Sanghi, A.; Morris, N.J.: Solidgen: An autoregressive model for direct b-rep synthesis. arXiv preprint arXiv:2203.13944, 2022.

[44] Jayaraman, P.K.; Sanghi, A.; Lambourne, J.G.; Willis, K.D.; Davies, T.; Shayani, H.; Morris, N.: Uv-net: Learning from boundary representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11703–11712, 2021.

[45] Jones, B.; Hildreth, D.; Chen, D.; Baran, I.; Kim, V.G.; Schulz, A.: Automate: A dataset and learning approach for automatic mating of cad assemblies. ACM Transactions on Graphics (TOG), 40(6), 1–18, 2021.

[46] Khan, M.S.; Dupont, E.; Ali, S.A.; Cherenkova, K.; Kacem, A.; Aouada, D.: Cad-signet: Cad language inference from point clouds using layer-wise sketch instance guided attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4713–4722, 2024.

[47] Kim, S.; Chi, H.g.; Hu, X.; Huang, Q.; Ramani, K.: A large-scale annotated mechanical components benchmark for classification and retrieval tasks with deep neural networks. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16, 175–191. Springer, 2020.

[48] Kipf, T.N.; Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[49] Koch, S.; Matveev, A.; Jiang, Z.; Williams, F.; Artemov, A.; Burnaev, E.; Alexa, M.; Zorin, D.; Panozzo, D.: Abc: A big cad model dataset for geometric deep learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9601–9611, 2019.

[50] Lambourne, J.G.; Willis, K.; Jayaraman, P.K.; Zhang, L.; Sanghi, A.; Malekshan, K.R.: Reconstructing editable prismatic cad from rounded voxel models. In SIGGRAPH Asia 2022 Conference Papers, 1–9, 2022.

[51] Lambourne, J.G.; Willis, K.D.; Jayaraman, P.K.; Sanghi, A.; Meltzer, P.; Shayani, H.: Brepnet: A topological message passing system for solid models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12773–12782, 2021.

[52] Lee, H.; Han, S.: Reconstruction of 3d interacting solids of revolution from 2d orthographic views. Computer-Aided Design, 37(13), 1388–1398, 2005.

[53] Lee, M.; Zhang, D.; Jambon, C.; Kim, Y.M.: Brepdiff: Single-stage b-rep diffusion model. In Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers, 1–11, 2025.

[54] Lee, S.H.; Lee, K.: Partial entity structure: a compact non-manifold boundary representation based on partial topological entities. In Proc. of the sixth ACM Symp. on Solid Modeling and Applications, 159–170, 2001.

[55] Lee, Y.T.; Fang, F.: 3d reconstruction of polyhedral objects from single parallel projections using cubic corner. Computer-Aided Design, 43(8), 1025–1034, 2011.

[56] Lee, Y.T.; Fang, F.: A new hybrid method for 3d object recovery from 2d drawings and its validation against the cubic corner method and the optimisation-based method. Computer-Aided Design, 44(11), 1090–1102, 2012.

[57] Li, C.; Pan, H.; Bousseau, A.; Mitra, N.J.: Sketch2cad: Sequential cad modeling by sketching in context. ACM Transactions on Graphics (TOG), 39(6), 1–14, 2020.

[58] Li, C.; Pan, H.; Bousseau, A.; Mitra, N.J.: Free2cad: parsing freehand drawings into cad commands. ACM Transactions on Graphics (TOG), 41(4), 1–16, 2022.

[59] Li, P.; Guo, J.; Li, H.; Benes, B.; Yan, D.M.: Sfmcad: Unsupervised cad reconstruction by learning sketch-based feature modeling operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4671–4680, 2024.

[60] Li, P.; Guo, J.; Zhang, X.; Yan, D.M.: Secad-net: Self-supervised cad reconstruction by learning sketch-extrude operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 16816–16826, 2023.

[61] Lin, C.; Fan, T.; Wang, W.; Nießner, M.: Modeling 3d shapes by reinforcement learning. In European Conference on Computer Vision, 545–561. Springer, 2020.

[62] Liu, J.; Tang, X.: Evolutionary search for faces from line drawings. IEEE Transactions on Pattern Analysis & Machine Intelligence, 27(06), 861–872, 2005.

[63] Liu, Y.; Obukhov, A.; Wegner, J.D.; Schindler, K.: Point2cad: Reverse engineering cad models from 3d point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3763–3772, 2024.

[64] Lun, Z.; Gadelha, M.; Kalogerakis, E.; Maji, S.; Wang, R.: 3d shape reconstruction from sketches via multi-view convolutional networks. In 2017 International Conference on 3D Vision (3DV), 67–77. IEEE, 2017.

[65] Mahajan, S.N.; Muthuganapathy, R.; et al.: Orthocad-322k: A cross-modal approach for retrieving 3d cad models from orthographic views using a graph-based framework on a developed large-scale dataset. Computers & Graphics, 104357, 2025.

[66] Mo, K.; Guerrero, P.; Yi, L.; Su, H.; Wonka, P.; Mitra, N.; Guibas, L.J.: Structurenet: Hierarchical graph networks for 3d shape generation. arXiv preprint arXiv:1908.00575, 2019.

[67] Para, W.; Bhat, S.; Guerrero, P.; Kelly, T.; Mitra, N.; Guibas, L.J.; Wonka, P.: Sketchgen: Generating constrained cad sketches. Advances in Neural Information Processing Systems, 34, 5077–5088, 2021.

[68] Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 652–660, 2017.

[69] 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.

[70] Qin, F.; Lu, S.; Hou, J.; Wang, C.; Fang, M.; Liu, L.: Drawing2cad: Sequence-to-sequence learning for cad generation from vectorized drawings. arXiv preprint arXiv:2508.18733, 2025.

[71] Ramos, S.; Trevisan, D.F.; Batagelo, H.C.; Sousa, M.C.; Gois, J.P.: Contour-aware 3d reconstruction of side-view sketches. Computers & Graphics, 77, 97–107, 2018.

[72] Ritchie, D.; Guerrero, P.; Jones, R.K.; Mitra, N.J.; Schulz, A.; Willis, K.D.; Wu, J.: Neurosymbolic models for computer graphics. In Computer graphics forum, vol. 42, 545–568. Wiley Online Library, 2023.

[73] Seff, A.; Ovadia, Y.; Zhou, W.; Adams, R.P.: Sketchgraphs: A large-scale dataset for modeling relational geometry in computer-aided design. arXiv preprint arXiv:2007.08506, 2020.

[74] Seff, A.; Zhou, W.; Richardson, N.; Adams, R.P.: Vitruvion: A generative model of parametric cad sketches. arXiv preprint arXiv:2109.14124, 2021.

[75] Sharma, G.; Liu, D.; Maji, S.; Kalogerakis, E.; Chaudhuri, S.; Měch, R.: Parsenet: A parametric surface fitting network for 3d point clouds. In European conference on computer vision, 261–276. Springer, 2020.

[76] Sobani, S.S.M.; Mahmood, N.H.; Zakaria, N.A.; Ariffin, I.: 3d model reconstruction from multi-views of 2d images using radon transform. Jurnal Teknologi, 74(6), 2015.

[77] Tam, K.; Atkinson, J.: An approach for creating solid models from orthogonal views by identification of boolean operations. Journal of materials processing technology, 138(1-3), 163–169, 2003.

[78] Tanaka, M.; Kaneeda, T.: Feature extraction from sketches of objects. Computer-Aided Design and Applications, 12(3), 300–309, 2015.

[79] Uy, M.A.; Chang, Y.Y.; Sung, M.; Goel, P.; Lambourne, J.G.; Birdal, T.; Guibas, L.J.: Point2cyl: Reverse engineering 3d objects from point clouds to extrusion cylinders. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11850–11860, 2022.

[80] Varley, P.A.; Company, P.P.: A new algorithm for finding faces in wireframes. Computer-Aided Design, 42(4), 279–309, 2010.

[81] Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903, 2017.

[82] Veselov, N.; Golovin, S.: A review of methods for computer-aided reconstruction of solid models of objects by their orthographic views. Moscow University Computational Mathematics and Cybernetics, 31(3), 128–132, 2007.

[83] Wagh, K.D.: Fast 3d modelling using orthographic views. In 2014 International Conference on Information Science, Electronics and Electrical Engineering, vol. 3, 1842–1845. IEEE, 2014.

[84] Wang, K.; Zheng, J.; Zhou, Z.: Neural face identification in a 2d wireframe projection of a manifold object. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1622–1631, 2022.

[85] Wang, P.; Yang, W.A.; You, Y.: A hybrid learning framework for manufacturing feature recognition using graph neural networks. Journal of Manufacturing Processes, 85, 387–404, 2023.

[86] Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog), 38(5), 1–12, 2019.

[87] Wang, Z.; Latif, M.: Reconstruction of a 3d solid model from orthographic projections. In 2003 International Conference on Geometric Modeling and Graphics, 2003. Proceedings, 75–82. IEEE, 2003.

[88] Wang, Z.; Latif, M.: Reconstruction of 3d solid models using fuzzy logic recognition. In World Congress on Engineering, 37–42, 2007.

[89] Weiss-Cohen, M.: 3d reconstruction of solid models from engineering orthographic views using variational geometry and composite graphs. Computer-Aided Design and Applications, 4(1-4), 159–167, 2007.

[90] Willis, K.D.; Jayaraman, P.K.; Lambourne, J.G.; Chu, H.; Pu, Y.: Engineering sketch generation for computer-aided design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2105–2114, 2021.

[91] Willis, K.D.; Pu, Y.; Luo, J.; Chu, H.; Du, T.; Lambourne, J.G.; Solar-Lezama, A.; Matusik, W.: Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences. ACM Transactions on Graphics (TOG), 40(4), 1–24, 2021.

[92] Wu, R.; Xiao, C.; Zheng, C.: Deepcad: A deep generative network for computer-aided design models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 6772–6782, 2021.

[93] Wu, R.; Zhuang, Y.; Xu, K.; Zhang, H.; Chen, B.: Pq-net: A generative part seq2seq network for 3d shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 829–838, 2020.

[94] Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 32(1), 4–24, 2020.

[95] Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE conference on computer vision and pattern recognition, 1912–1920, 2015.

[96] Xu, X.; Jayaraman, P.K.; Lambourne, J.G.; Willis, K.D.; Furukawa, Y.: Hierarchical neural coding for controllable cad model generation. arXiv preprint arXiv:2307.00149, 2023.

[97] Xu, X.; Lambourne, J.; Jayaraman, P.; Wang, Z.; Willis, K.; Furukawa, Y.: Brepgen: A b-rep generative diffusion model with structured latent geometry. ACM Transactions on Graphics (TOG), 43(4), 1–14, 2024.

[98] Xu, X.; Peng, W.; Cheng, C.Y.; Willis, K.D.; Ritchie, D.: Inferring cad modeling sequences using zone graphs. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 6062–6070, 2021.

[99] Xu, X.; Willis, K.D.; Lambourne, J.G.; Cheng, C.Y.; Jayaraman, P.K.; Furukawa, Y.: Skexgen: Autoregressive generation of cad construction sequences with disentangled codebooks. arXiv preprint arXiv:2207.04632, 2022.

[100] Yan, S.; Yang, Z.; Ma, C.; Huang, H.; Vouga, E.; Huang, Q.: Hpnet: Deep primitive segmentation using hybrid representations. In Proceedings of the IEEE/CVF international conference on computer vision, 2753–2762, 2021.

[101] Ying, Z.; You, J.; Morris, C.; Ren, X.; Hamilton, W.; Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. Advances in neural information processing systems, 31, 2018.

[102] Zakharov, A.; Zhiznyakov, A.: Synthesis of three-dimensional models from drawings based on spectral graph theory. In Applied Mechanics and Materials, vol. 756, 598–603. Trans Tech Publ, 2015.

[103] Zhang, C.; Pinquié, R.; Polette, A.; Carasi, G.; De Charnace, H.; Pernot, J.P.: Automatic 3d cad models reconstruction from 2d orthographic drawings. Computers & Graphics, 114, 179–189, 2023.

[104] Zhang, C.; Polette, A.; Pinquié, R.; Carasi, G.; De Charnace, H.; Pernot, J.P.: ecad-net: Editable parametric cad models reconstruction from dumb b-rep models using deep neural networks. Computer-Aided Design, 103806, 2024.

[105] Zhang, C.; Polette, A.; Pinquié, R.; Iida, M.; De Charnace, H.; Pernot, J.P.: Reinforcement learning-based parametric cad models reconstruction from 2d orthographic drawings. Computer-Aided Design, 103925, 2025.

[106] Zhang, M.; Cui, Z.; Neumann, M.; Chen, Y.: An end-to-end deep learning architecture for graph classification. In Proceedings of the AAAI conference on artificial intelligence, vol. 32, 2018.

[107] Zhang, S.; Guan, Z.; Jiang, H.; Ning, T.; Wang, X.; Tan, P.: Brep2seq: a dataset and hierarchical deep learning network for reconstruction and generation of computer-aided design models. Journal of Computational Design and Engineering, 11(1), 110–134, 2024.

[108] Zhang, S.; Shi, Y.; Fan, H.; Huang, R.; Cao, J.: Serial 3d model reconstruction for machining evolution of rotational parts by merging semantic and graphic process planning information. Computer-aided design, 42(9), 781–794, 2010.

[109] Zhang, Z.; Jaiswal, P.; Rai, R.: Featurenet: Machining feature recognition based on 3d convolution neural network. Computer-Aided Design, 101, 12–22, 2018.

[110] Zhao, H.S.; Wu, Y.X.; Wu, Y.Z.: A new approach to reconstructing 3d solid from the two 2d views based on autocad. In Applied Mechanics and Materials, vol. 88, 697–702. Trans Tech Publ, 2011.

[111] Zhou, Q.; Jacobson, A.: Thingi10k: A dataset of 10,000 3d-printing models. arXiv preprint arXiv:1605.04797, 2016.

[112] Zhou, S.; Tang, T.; Zhou, B.: Cadparser: A learning approach of sequence modeling for b-rep cad. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, 1804–1812, 2023.

[113] Zou, C.; Yumer, E.; Yang, J.; Ceylan, D.; Hoiem, D.: 3d-prnn: Generating shape primitives with recurrent neural networks. In Proceedings of the IEEE International Conference on Computer Vision, 900–909, 2017.

[114] Zou, Q.; Feng, H.Y.: Push-pull direct modeling of solid cad models. Advances in Engineering Software, 127, 59–69, 2019.