

# CADDreamer: CAD Object Generation from Single-view Images

Yuan Li<sup>1</sup> Cheng Lin<sup>2†</sup> Yuan Liu<sup>3</sup> Xiaoxiao Long<sup>4</sup> Chenxu Zhang<sup>5</sup>  
 Ningna Wang<sup>1</sup> Xin Li<sup>6</sup> Wenping Wang<sup>6</sup> Xiaohu Guo<sup>1†</sup>

<sup>1</sup>University of Texas at Dallas <sup>2</sup>The University of Hong Kong <sup>3</sup>Hong Kong University of Science and Technology  
<sup>4</sup>Nanjing University <sup>5</sup>ByteDance <sup>6</sup>Texas A&M University

†Corresponding Authors

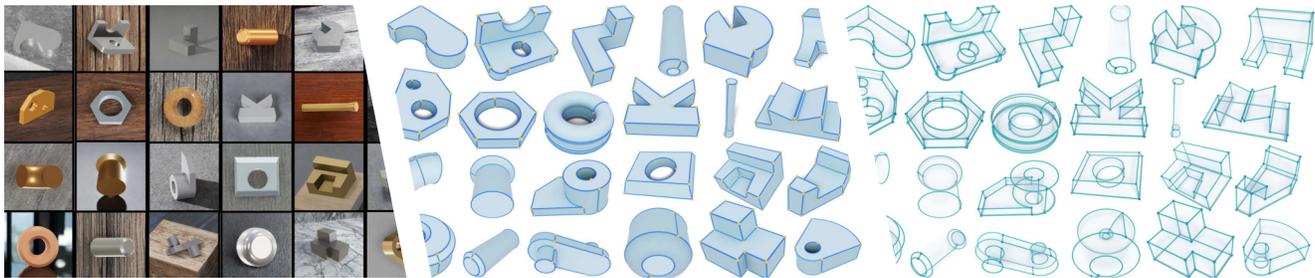


Figure 1. A gallery of our reconstructed CAD models (middle and right) from single-view RGB images (left). The reconstructed shapes are shown in light blue (middle), while their topological representations of B-rep vertices and edges are shown in green (right).

## Abstract

Diffusion-based 3D generation has made remarkable progress in recent years. However, existing 3D generative models often produce overly dense and unstructured meshes, which stand in stark contrast to the compact, structured, and sharply-edged Computer-Aided Design (CAD) models crafted by human designers. To address this gap, we introduce CADDreamer, a novel approach for generating boundary representations (B-rep) of CAD objects from a single image. CADDreamer employs a primitive-aware multi-view diffusion model that captures both local geometric details and high-level structural semantics during the generation process. By encoding primitive semantics into the color domain, the method leverages the strong priors of pre-trained diffusion models to align with well-defined primitives. This enables the inference of multi-view normal maps and semantic maps from a single image, facilitating the reconstruction of a mesh with primitive labels. Furthermore, we introduce geometric optimization techniques and topology-preserving extraction methods to mitigate noise and distortion in the generated primitives. These enhancements result in a complete and seamless B-rep of the CAD model. Experimental results demonstrate that our method effectively recovers high-quality CAD objects from single-view images. Compared to existing 3D generation

techniques, the B-rep models produced by CADDreamer are compact in representation, clear in structure, sharp in edges, and watertight in topology.

## 1. Introduction

Recently, the field of image-based 3D generation has undergone significant advancements [10, 25–27, 32, 36, 40]. Progress in diffusion models trained on large-scale datasets [5, 10, 34] has greatly streamlined the transformation of 2D images into 3D models, heralding a revolution in 3D content creation.

Despite the increasing amount of work on image-based 3D generation, a critical issue remains. The meshes produced by these methods are typically derived from low-level representations such as neural implicit fields [30, 37], leading to over-tessellated meshes that lack explicit geometric structure and semantics. In contrast, human designers often conceptualize 3D objects at a higher level, perceiving shapes as compositions of basic primitives and feature curves [24]. As a result, designer-created 3D meshes are compact in representation while preserving clear structures and sharp edges. This significant disparity limits the practical applicability of 3D generative models in scenarios demanding high-quality, structured 3D models, such as gaming, manufacturing, and product design.

To bridge the quality gap between shapes created by human designers and those generated by 3D generative models, this paper focuses on reconstructing Computer-Aided Design (CAD) models from single-view images using diffusion-based generative models. CAD objects offer a higher level of abstraction in shape modeling and ensure structural integrity aligned with human perception. Consequently, generating CAD objects requires diffusion models to understand and interpret high-level structures and geometric primitives. However, single-view images capture only partial information about 3D shapes, necessitating the inference of complete CAD models comprising structured geometric primitives. Additionally, reconstructing the watertight boundary representation (B-rep) of CAD objects poses significant challenges due to inevitable noise and distortion in the generated geometric primitives. Even minor deviations in primitives can disrupt intersection edges, resulting in misaligned primitives or non-watertight B-reps that are unsuitable for manufacturing and product design.

To address these challenges, we propose *CADDreamer*, a method for reconstructing CAD objects from single-view images. It consists of the following two main modules.

**Module 1 - Multi-view Generation:** *CADDreamer* reconstructs a complete 3D mesh and segments it into patches representing six types of geometric primitives: planes, cylinders, cones, spheres, tori, and boundary feature lines. Specifically, we employ a cross-domain multi-view diffusion model [29] to jointly predict normal maps and semantic primitive maps. The normal maps are fed into NeuS [37] to reconstruct the complete 3D mesh. Semantic primitive maps are back-projected onto the mesh faces, and a Graph Cut process divides the reconstructed mesh into patches, each corresponding to a geometric primitive.

**Module 2 - Geometric and Topological Extraction:** *CADDreamer* estimates the primitive parameters for each patch and calculates the vertices, edges, and faces of B-rep by determining intersections between primitives. To ensure accuracy, a geometry optimization algorithm corrects noisy primitive parameters, restoring relationships such as parallelism, perpendicularity, collinearity, and intersections. This ensures proper intersection curves are computed. The resulting topological representation, derived from the segmented mesh, guides the intersection of geometric primitives, ultimately producing a watertight B-rep model with accurate CAD vertices, edges, and faces.

The contributions of this paper can be summarized as:

- **CADDreamer Framework:** We propose *CADDreamer*, a two-module framework for high-quality CAD reconstruction from single-view images. The first module leverages large-scale diffusion models [34], exhibiting superior generative capabilities and a powerful capacity to handle diverse geometric shapes. With cross-view and cross-domain attention mechanisms, the diffusion model

ensures high consistency in both geometry and semantics across multiple views. However, the reconstructed mesh from multi-view images may inevitably suffer from distortions and noise. Even with semantic guidance, these inaccuracies can lead to failed computations of geometric primitive intersections, resulting in misaligned primitives or non-watertight B-reps. To address these issues, the second module of *CADDreamer* is designed to solve the associated geometric and topological challenges.

- **Semantic-enhanced Multi-view 2D Diffusion:** We propose a multi-view 2D diffusion model that perceives high-level semantics by encoding semantic information into the color space. This approach differs from existing methods [25, 27, 40], which primarily focus on generating low-level information such as colors and normals. By enforcing strong priors in diffusion models, our method aligns operations with well-defined primitives, inherently enabling the model to understand and interpret high-level structures of CAD objects.
- **Geometry Optimization Algorithm:** We introduce a geometry optimization algorithm to refine the parameters of geometric primitives. This ensures the preservation of topological and geometric relationships between neighboring primitives, such as intersections, parallelism, perpendicularity, and collinearity.
- **Topology-preserving B-rep Construction:** We extract a topological representation from segmented meshes and use it to guide the topology-preserving extraction of CAD vertices, edges, and faces. This process ultimately contributes to generating a watertight and accurate B-rep of CAD objects.

## 2. Related work

**Traditional CAD reconstruction methods.** Before the advent of learning-based reconstruction, a plethora of traditional methods had been developed for B-rep reconstruction, including parameter space-based methods [23, 33], primitive growing/fitting-based methods [3, 21, 31, 35], and variational surface fitting-based methods [42, 45]. For an overview of these traditional methods, we refer readers to the survey [13]. A major criticism of these traditional methods is that they require users' aid, e.g., specifying primitive fitting types manually [35]. Conversely, *CADDreamer* utilizes a multi-view diffusion module to estimate fitting primitive types and avoid human intervention.

**Learning-based CAD Reconstruction Methods.** Recently, with the advancement of deep learning, numerous learning-based CAD generation methods have been proposed. These methods can be divided into three main categories, including CAD instructions-based methods, retrieval and assembly-based methods, and parameter surface-based methods. Firstly, CAD instructions-based methods [17, 19, 39] reduce the B-rep generation into the

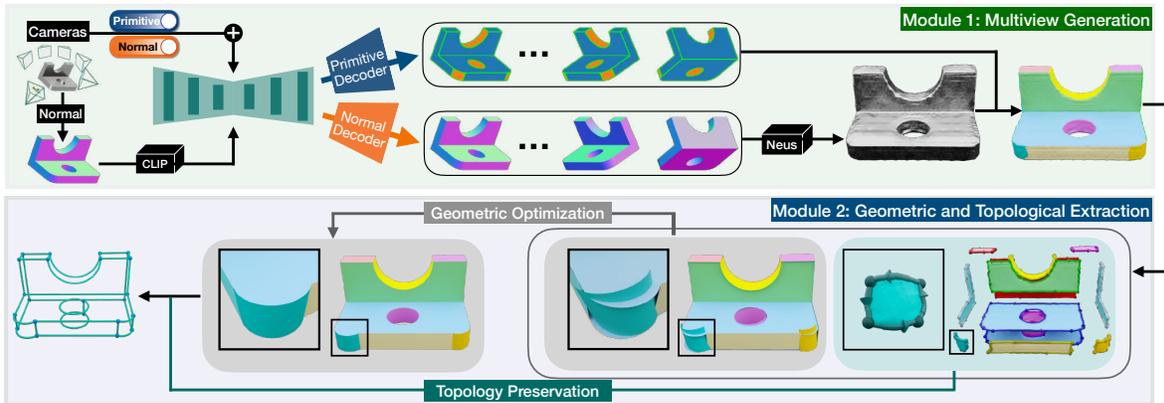


Figure 2. The pipeline of CADDreamer. In the first module, the given single-view RGB image is converted as a normal map. Using the normal map as input, the generation module uses a diffusion process to generate multi-view normal and semantic primitive maps. Inputting the multi-view normal map into Neus [37], we obtain 3D meshes; back-projecting semantic primitive maps into 3D meshes, we segmented the mesh into several patches with a Graph Cut process. In the second module, geometric optimization corrects the noisy primitive parameters, while the topology-preserving extraction computes their topology-guided intersections and reconstructs a watertight B-rep CAD model.

natural language generation. By utilizing a large amount of CAD instructions as training data, these methods aim to generate new CAD instructions to achieve CAD generation. However, these methods rely on massive CAD objects containing CAD instructions, resulting in unacceptable data preparation costs and limiting the potential applications of these methods. Secondly, retrieval and assembly-based methods [12, 14, 17, 44] transform the problem of B-rep reconstruction into a retrieval problem. Using the given images or shapes as keys, these methods retrieve several CAD objects with similar shapes and learn a potential assembly scheme to achieve B-rep reconstruction, i.e., CSG trees [14, 44]. Because maintaining a vast and comprehensive CAD database for retrieving typically incurs massive data preparation costs, these methods are not the first choice for image-to-CAD conversion. Recently, parameter surface-based methods [22, 28, 41] have made significant progress in CAD reconstruction. These methods first divide the shape into several parameter surfaces (e.g., primitives), and then estimate the parameters and topologies of these parameter surfaces to implement CAD reconstruction. These methods have made rapid progress in CAD reverse engineering, but due to the constraints of their pipelines, they cannot take images as inputs and achieve image-to-CAD conversion.

**Image2CAD.** Most existing Image2CAD methods rely on a retrieval-and-assembly pipeline based on implicit surfaces rather than directly generating B-reps [7, 9, 11, 16, 18]. Another solution [1, 4, 43] generates B-reps with the conventional sketch-extrude commands. While these methods can produce B-reps directly, the generated objects are limited to planes and cylinders. In contrast, CADDreamer overcomes these limitations by directly generating B-reps with

a diverse set of primitives.

**Single-view Reconstruction Methods.** Large-scale diffusion models [34], pre-trained on billion-scale data, have shown tremendous potential and robust performance in both 2D and 3D generation tasks. Numerous methods based on large-scale diffusion modules have been applied to tackle the problem of single-view 3D reconstruction and have achieved state-of-the-art performance [38, 40], such as Wonder3D [29], SyncDreamer [27], and LRM [10]. Despite their success, these approaches typically depend on low-level representations to reconstruct 3D shapes, which lack high-level shape comprehension. This often results in significant noise and distortions in the reconstructed meshes. In contrast, CADDreamer is designed to capture both low-level geometric representations and high-level structured representations of 3D shapes, including seven types of primitives, to mitigate reconstruction noises and contribute to more compact geometric representations.

### 3. Method

As shown in Figure 2, CADDreamer consists of two main modules: a multi-view generation module and a geometric and topological extraction module. First, the multi-view generation module takes a single-view RGB image to reconstruct a complete 3D triangular mesh and then decomposes this mesh into distinct patches, each corresponding to a primitive. Second, the geometric and topological extraction module refines the parameters of each primitive and performs CAD reconstruction through topology-preserving intersections between neighboring primitives. Sec. 3.1 details the multi-view generation module, while Sec. 3.2 elaborates on the geometric and topological extraction module.

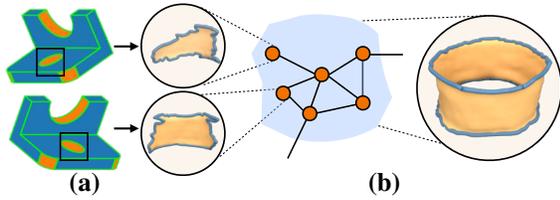


Figure 3. An example of the Graph Cut process to obtain complete 3D mesh patches representing primitives. (a) depicts the mesh patches generated by feature-line cut and back-projection. (b) illustrates the process of Graph Cut algorithm to obtain mesh patches corresponding to primitives.

### 3.1. Multi-view Generation Module

The multi-view generation module is designed to reconstruct a complete 3D mesh and then partition the triangles into distinct patches, each representing a CAD primitive. While numerous existing multi-view diffusion modules achieve state-of-the-art performance in 3D mesh reconstruction, these methods devote significant effort to reconstructing multi-view textures and surface materials, which are unnecessary for CADDreamer. Instead, our multi-view generation module focuses on both reconstructing a complete 3D mesh and generating a structured shape representation—specifically, segmenting the mesh into triangle patches that represent CAD primitives.

Thus, we develop a new multi-view generation module, as illustrated in Figure 2. The entire process can be summarized in four steps. First, we use a popular normal predictor [2, 20] to generate a normal map from the single-view RGB image. Second, using the generated normal map as input, we fine-tune the popular cross-domain diffusion generative model, Wonder3D [29], to produce  $m$  multi-view normal maps and  $m$  multi-view semantic primitive maps, where  $m = 6$  and feature lines have a width of three pixels in this paper. In these maps, the RGB values in the normal maps indicate normal directions, while those in each primitive map represent primitive types, including feature lines that separate neighboring primitives. To enhance generation performance, we fine-tune two separate VAE decoders for normal and primitive map generation, respectively. Third, we input multi-view normal maps into NeuS [37] for 3D mesh reconstruction. We remove the multi-view color inputs and associated texture reconstruction loss from NeuS, as CADDreamer does not require texture reconstruction. Finally, we utilize feature lines to divide each primitive map into several 2D patches. These 2D patches are then back-projected onto the mesh, where a Graph Cut process is applied to merge the back-projected patches into a cohesive set of 3D mesh patches that represent different primitives.

**Graph Cut Process.** As shown in Figure 3, each back-projected mesh patch is treated as an individual node, and connectivity edges are added between adjacent or overlapping patches. Mesh patches sharing triangle boundaries are

Primitives	Parameters
Plane	$\vec{x}, p$ : normal and position
Cylinder	$\vec{x}, p$ : axis and position, $r$ : radius
Cone	$\vec{x}, p$ : axis and center position, $\alpha$ : semi-angle, $h$ : height
Torus	$\vec{x}, p$ : axis and center position $r_l$ : major radius, $r_s$ : minor radius
Sphere	$p$ : center position, $r$ : radius
Relationships	Determination
Topology Relationship	Intersected : ( Intersection is not $\emptyset$ )
Geometric Relationships	Perpendicular: ( $\vec{x}_1 \cdot \vec{x}_2 = 0$ ) Parallel: ( $\vec{x}_1 \cdot \vec{x}_2 = 1$ ) Collinear : ( $\exists t : p_1 = p_2 + t\vec{x}_2$ )

Table 1. Definitions of parameters and relationships for primitives.

considered adjacent, while those with overlaps are overlapped. Using Efficient RANSAC [35], we extract primitive parameters and calculate their cosine similarity as edge weights. Since the 3D patches derived from the same images correspond to different primitives, merging patches from the same images is considered an error in the graph cut methodology. Given that the number of primitives in CAD objects typically exceeds two, an initial cut threshold is set to divide the mesh into two surface patches. This threshold is then gradually increased to yield more connected components. The minimum cut threshold that results in the fewest errors is selected as the final cut threshold. Thus, through the Graph Cut process, the multi-view generation module effectively separates the constructed mesh into distinct surface patches, each representing a CAD primitive.

Compared to previous models, our multi-view generation module incorporates two significant improvements. First, we introduce an off-the-shelf normal prediction model that converts the original images into normal maps, thereby alleviating the interference caused by image textures and shading variations. As a result, the impact of diverse textures and shading can be minimized, significantly reducing the problem’s complexity and contributing to a more robust reconstruction process. Second, we implement a new cross-domain generation strategy for multi-view normal and semantic primitive maps. By generating these maps together, we not only reconstruct a 3D mesh but also segment it into a structured representation of shapes, i.e., the five types of primitives illustrated in Table 1.

### 3.2. Geometric and Topological Extraction Module

The multi-view generation module in Sec.3.1 converts the given image into a 3D mesh and decomposes it into multiple mesh patches, each representing a distinct primitive. The geometric optimization module then aims to extract prim-

itive surfaces and generate a CAD model by finding intersections between these primitives. This process consists of four main steps:

- First, we input the vertices of each mesh patch into the RANSAC [35] algorithm to extract the parameters for each primitive, as defined in Table 1.
- Next, we introduce a novel geometric optimization algorithm to recover the topological relationships (e.g., intersections) and geometric relationships (e.g., parallelism, perpendicularity, and collinearity) between primitives, with detailed definitions provided in Table 1.
- Third, we extract a topological representation from the segmented mesh patches, encompassing vertices, edges, and faces, as referred to Figure A1 in the supplement.
- Finally, using the extracted topological representation, we perform topology-preserving intersection operations between each primitive surface and its neighboring primitives to create CAD models’ vertices, edges, and faces.

The details of each step of the process are outlined below.

**Primitive Extraction.** In the first step, we build on previous studies [35] by using the RANSAC algorithm to estimate the parameters of primitives. Unlike direct least squares fitting, which can be significantly affected by outlier data, RANSAC helps to mitigate the impact of such outliers. However, due to the inevitable distortion in 3D mesh reconstruction from single-view images, the primitive parameters obtained from RANSAC may still be inaccurate. As a result, some essential topological and geometric relationships between the primitives might not be preserved. As illustrated in Figure 4(a), the expected intersecting primitive surfaces may fail to intersect, leading to incorrect topological relationships and reconstruction failures for vertices and edges. Furthermore, since the parameters of primitives are based on points, lines, and directions, it is crucial to preserve key geometric relationships between these elements, such as collinearity among points and lines, as well as parallelism and perpendicularity among directions. As shown in Figure 4(b), (c), and (d), neglecting these vital geometric relationships can result in completely erroneous intersection curves, which may cause gaps in the reconstructed CAD model. Therefore, in the next step, we aim to establish these important relationships between primitives before implementing the intersection reconstruction strategy.

**Geometric Optimization.** In the second step, we introduce a novel geometric optimization algorithm designed to refine primitive parameters and restore four types of primitive relationships. Specifically, our algorithm includes a primitive stitching process aimed at recovering intersection relationships while incorporating geometric relationships as constraints to ensure that the final primitive parameters conform to these geometric relationships. The primitive stitching process can be divided into three key sub-steps.

First, we derive primitive intersection relationships from

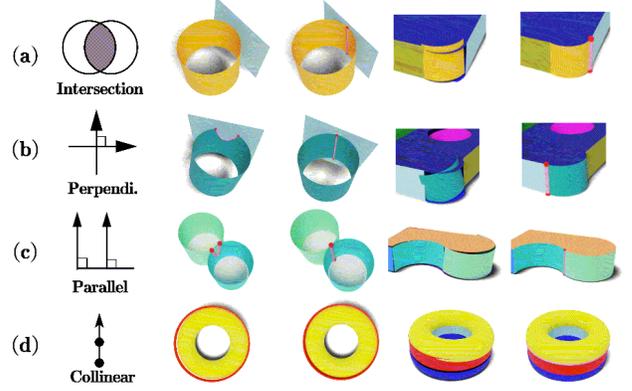


Figure 4. Four key primitive relationships: (a-b) cylinder-plane intersection and perpendicularity; (c) parallel cylinders; (d) collinear cylinder and torus. Incorrect relationships (columns 2 & 4) yield flawed intersections, while correct ones (columns 3 & 5) produce accurate results.

adjacent patches on segmented meshes: primitives intersect if and only if neighboring patches share boundaries. Second, for each intersection relationship, we sample  $k$  boundary vertices closest to intersecting primitives as *stitching vertices* (Figure 5(b)). We uniformly sample candidate points along mesh segmentation boundaries (line-strings) and select  $k$  points with minimal projection distances to the intersecting primitives as stitching vertices. Third, we project each stitching vertex onto the two primitive surfaces and minimize the distance between the corresponding projection points by optimizing the parameters of the primitive surfaces, as shown in Figure 5(c-f).

The optimization function for a stitching vertex  $v_i$  between the two corresponding primitive surfaces  $A$  and  $B$  is defined as follows:

$$f_{stch}(v_i) = \|\pi(v_i, P_A) - \pi(v_i, P_B)\|, \quad (1)$$

where  $P_A$  and  $P_B$  are the optimizable shape parameters of the primitive surfaces  $A$  and  $B$ , and  $\pi(v_i, P_A)$  and  $\pi(v_i, P_B)$  represent the projection points of  $v_i$  on  $A$  and  $B$ , respectively. By minimizing the sum of optimization functions for all stitching vertices  $\sum_{v_i \in \mathcal{V}} f_{stch}(v_i)$ , the primitive stitching algorithm reduces the distances between primitive surfaces that should intersect, helping to prevent intersection failures and preserve correct topological relationships between neighboring primitives.

We also incorporate certain geometric relationships as constraints within the primitive stitching process to ensure that the resulting parameters satisfy these relationships.

- A *parallel* relationship means that two primitives share the same axis direction, allowing us to optimize a single axis direction while maintaining consistency between the axes of both primitives.
- A *collinear* relationship implies that the position of prim-

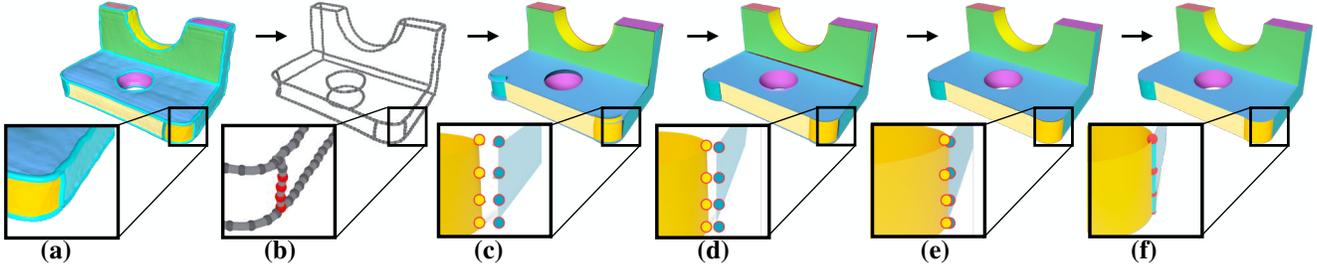


Figure 5. An example of primitive stitching process with four stitching vertices. (a) shows the patch boundaries on reconstructed mesh, (b) represents the sampled stitching vertices ( $k=4$ ), (c) illustrates the initial primitives and the projected points of stitching vertices, (d) depicts the stitching result after the first step of optimization, (e) shows the result at the 100th step, and (f) represents the final stitching result.

itive  $A$  (denoted as  $p_A$ ) can be expressed in terms of the parameters of primitive  $B$ :  $p_A = p_B + \vec{x}_B * t$ , where  $\vec{x}_B$  is the axis direction of primitive  $B$ , as shown in Table 1. Therefore, to enforce collinearity constraints, we optimize the parameter  $t$  instead of  $p_A$ .

- For a *perpendicular* relationship, we utilize the dot product between the two axes as an additional optimization function. Given the set of all stitching points  $\mathcal{V}$  and all perpendicular relationships  $\mathcal{P}$ , the optimization function can be expressed as:  $\sum_{v_i \in \mathcal{V}} f_{stch}(v_i) + \sum_{(C,D) \in \mathcal{P}} \vec{x}_C \cdot \vec{x}_D$ , where  $C$  and  $D$  are perpendicular primitives, and  $\vec{x}_C$  and  $\vec{x}_D$  are their axis directions, respectively.

In our implementation, the three geometric relationships are detected if the equations from Table 1 hold within a tolerance of 0.05. By employing the L-BFGS algorithm [6] to optimize the objective function, we effectively minimize the distances between primitive surfaces that should intersect and reconstruct the geometric relationships between primitives. Figure 5 illustrates an example of four stitching points. As shown in Figure 5 (d), by enforcing constraints to restore parallel and collinear relationships, the first step of the optimization process re-establishes the parallel relationship between the axes of the cylinders and planes. Subsequently, as the optimization progresses, we gradually restore the perpendicular and intersection relationships between the primitives, as shown in Figure 5 (e-f).

**Extraction of Topological Representation.** In the third step, we extract a watertight topological representation from the mesh to guide the intersection of the primitives. Specifically, each 3D mesh patch corresponds to a topological face, while the edge curves adjacent to two patches are considered topological edges. Additionally, mesh vertices connected to more than two patches are regarded as topological vertices. The half-edge structure of the meshes provides directionality for the patch boundaries, and we inherit these directions as the orientations of the topological edges. This process enables us to extract a topological representation from the segmented mesh, including topological vertices, edges, and faces (see Figure A1 in the supplementary material). Since the reconstructed meshes are watertight, this topological representation is also watertight.

**Topology-preserving CAD Reconstruction.** In the final step, we utilize this watertight topology as guidance to construct the CAD model using the following intersection strategy, which is divided into two main sub-steps.

First, we reconstruct the CAD curves for each face. Referring to Figure A1(a) in the supplement, each topological edge’s neighborhood comprises two patches corresponding to two primitives. We then compute the intersection of these two primitives to obtain CAD curves. If multiple intersection curves exist between two primitives, we select the one with the closest projection distance to the corresponding topological edge. By applying this process to all topological edges, we generate the CAD curves for each one.

Second, we reconstruct the CAD vertices, edges, and faces using the following intersection strategy. Each topological vertex connects two topological edges, or CAD curves. We compute the intersection of these two corresponding CAD curves as the intersection point of the three related primitives, given that each CAD curve is associated with two neighboring primitives and one primitive is shared between the two CAD curves. If multiple intersected vertices are present, we select the vertex closest to the topological vertex as the reconstructed vertex. By applying this process to all topological vertices, we obtain an equal number of CAD vertices. Using these CAD vertices to trim the CAD curves allows us to create CAD edges. Furthermore, by utilizing these CAD edges to trim the primitives, we obtain CAD faces. By merging these CAD vertices, edges, and faces, we create a complete and watertight boundary representation (B-rep) for the CAD model.

## 4. Experiments

**Dataset.** We begin by training and evaluating CADDreamer on synthetic images, followed by testing it on real-world images. For the synthetic dataset, we curate 30,000 seamless CAD models from the ABC [15] and DeepCAD [39] datasets. These CAD objects are enhanced with textures and backgrounds, then rendered into multi-view images, resulting in a training set of 29,000 samples and a testing set of 1,000 samples. To maintain consistency, we adopt the

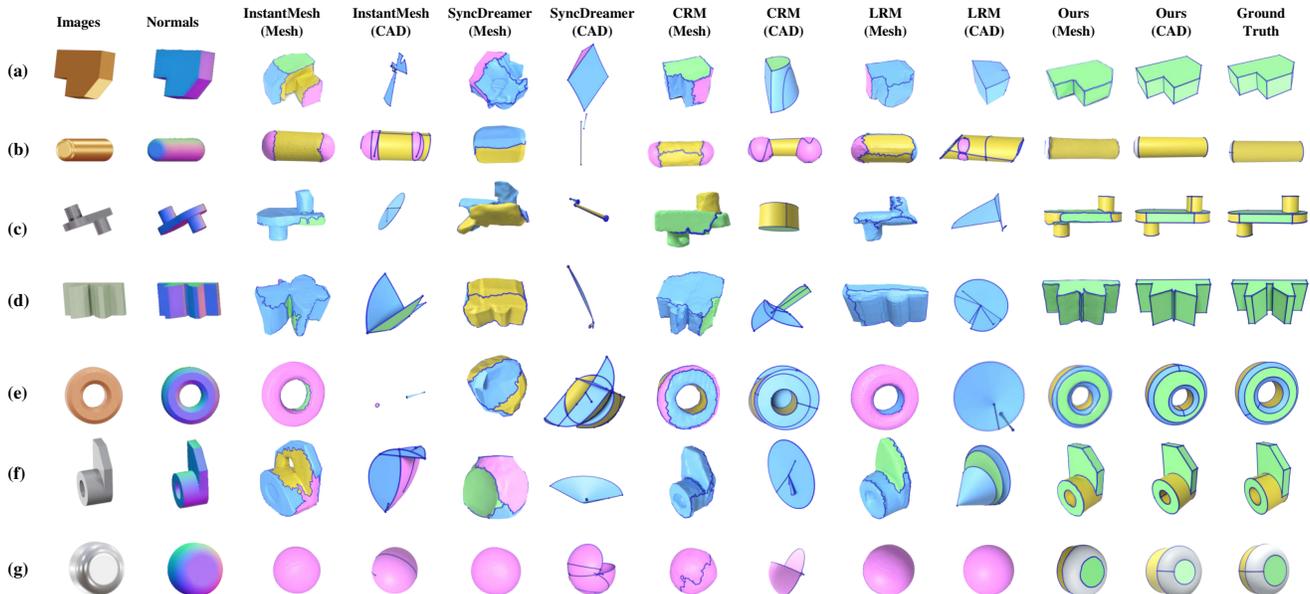


Figure 6. The segmentation results and reconstructed B-rep models. Electric blue (●) represents segmentation boundaries and edges of B-rep objects. Other colors represents different primitives: plane (●), cylinder (●), cone (●), sphere (●), and torus (●).

camera configurations and image resolution settings used in Wonder3D [29]. Furthermore, we capture RGB images of real CAD objects using handheld devices to assess our model’s generalization performance on real-world objects.

**Baselines.** Currently, popular approaches for reconstructing B-reps fall into two categories: conventional sketch-extrude methods [1, 4, 43] and more recent intersection-based methods [28]. Because our dataset includes CAD models with a diverse range of primitives—such as spheres, tori, and cones—that cannot be reconstructed using conventional sketch-extrude methods [1, 4, 43], we focus on comparing intersection-based models, which can handle these varied primitive types. Our evaluation framework integrates state-of-the-art single-view reconstruction methods (e.g., SyncDreamer [27], LRM [10], CRM [38], and InstantMesh [40]) with B-rep reconstruction techniques like Point2CAD [28]. For benchmarking, we apply these single-view generation methods to create meshes, and then use Point2CAD to segment the meshes into primitive shapes, estimate primitive parameters, and convert them into B-reps. Additional comparative assessments of the traditional sketch-extrude approach within point cloud inputs are documented in the supplementary material.

**Metrics.** We evaluate generation quality across three aspects: meshes, primitives, and B-reps. First, an ideal reconstructed mesh should minimize reconstruction distortion and align closely with the original B-reps (i.e., primitives). To assess geometric alignment between the reconstructed mesh and the ground truth, we utilize two widely-used metrics: Chamfer Distance (CD) and Normal

Methods	CD (↓)	NC (↑)	SEG(V) (↑)	SEG(P) (↑)
CRM [38]	3.97	64.4	40.2	49.3
LRM [10]	4.26	63.6	38.4	46.8
InstantMesh [40]	4.61	58.3	35.1	41.7
SyncDreamer [27]	5.49	48.9	29.8	33.2
CADDreamer	<b>1.27</b>	<b>92.6</b>	<b>95.7</b>	<b>97.9</b>

Table 2. Statistical results of reconstructed meshes and extracted primitives, including Chamfer distance (CD  $\times 100$ ), normal consistency (NC, %), segmentation accuracy based on vertices (SEG(V), %), and segmentation accuracy based on primitive counts (SEG(P), %). Best values are highlighted in bold.

Consistency (NC) [8]. Second, incorrect segmentation results—particularly the omission of primitives—can cause significant fitting errors and reconstruction failures. To address this, we evaluate both the percentage of correctly classified mesh vertices (SEG(V)) and the accuracy of predicted primitive counts (SEG(P)). Additionally, fitting errors in primitives can disrupt correct intersection relationships, resulting in hanging faces or non-closed B-reps. To quantify these issues, we calculate the proportion of hanging faces (HF) in the resulting B-reps and measured the deviation of the reconstructed B-reps from the ground truth using Chamfer Distance (CD).

**Settings.** We use Wonder3D [29] as our pre-trained model to ensure more stable and faster convergence. All experiments are conducted on a single machine equipped with eight NVIDIA A100 GPUs (80GB each) and an AMD EPYC 7313 CPU. The details of the comparative analyses are presented below, while ablation studies and limitation analyses are provided in the supplementary materials.

Methods	HF ( $\downarrow$ )	CD ( $\downarrow$ )
CRM [38]	35.2	9.74
LRM [10]	39.6	11.6
InstantMesh [40]	43.6	13.1
SyncDreamer [27]	58.5	15.4
CADDreamer	<b>2.4</b>	<b>1.36</b>

Table 3. Statistical results of reconstructed B-reps, including the percentage of B-reps with hanging faces (HF, %), and Chamfer distance ( $CD \times 100$ ) between the reconstructed B-reps and ground truth. Best values are highlighted in bold.

#### 4.1. Comparisons on Synthetic Images

Figure 6 visually showcases the reconstruction results from synthetic images, while Table 2 and 3 provide quantitative comparisons. These results demonstrate that CADDreamer offers significant advantages in mesh generation quality, primitive extraction, and the topological fidelity of the reconstructed B-reps.

CADDreamer’s key strength lies in its ability to simultaneously capture low-level geometric features (normal maps) and high-level semantic shape understanding (primitive maps), resulting in meshes with minimal distortion. As shown in Figure 6, our method generates clean, compact meshes with sharp edges, minimizing fitting errors and reconstruction failures. In contrast, competing methods, which lack high-level semantic understanding, generate distorted reconstructions that significantly deviate from the primitives. Quantitative results in Table 2 further validate CADDreamer’s performance, showing that its meshes exhibit the least geometric deviation from the ground truth, with the lowest CD and highest NC values.

In addition, CADDreamer employs a back-projection and Graph Cut process for primitive extraction, rather than relying on traditional mesh or point cloud segmentation methods. This approach enables more precise primitive extraction. Single-view RGB image reconstructions often produce meshes with errors such as uneven surfaces and over-smoothed boundaries. When these flawed meshes are processed through segmentation-based methods for primitive extraction, the reconstruction errors are amplified. For example, two twisted planes with a smooth transition might be misidentified as a single cone, as shown in Figure 6(a) LRM. Using such incorrectly identified primitives for mesh reconstruction inevitably results in excessive fitting errors and reconstruction failures. In contrast, our method demonstrates superior primitive extraction that aligns closely with ground truth CAD models. Our segmentation approach achieves more precise primitive labeling than other methods, with 97.9% of meshes correctly matching the primitive count of the ground truth, as shown in Table 2.

Thirdly, CADDreamer refines critical relationships between adjacent primitives, such as intersection, paral-

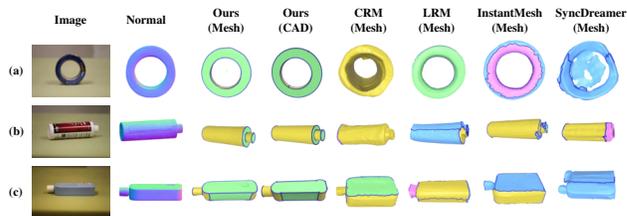


Figure 7. Segmentation results and reconstructed B-reps from real images (color definitions are the same as in Figure 6).

lelism, perpendicularity, and collinearity. It also employs a topology-preserving intersection strategy to reconstruct B-reps with accurate topology, reducing two common errors: hanging faces that arise from incorrect primitive intersections, and massive non-closed B-reps caused by non-intersection, as shown in Figure 6(a-f). As shown in Table 3, CADDreamer minimizes the percentages of hanging faces (HF) while significantly reducing face reconstruction failures caused by non-intersection, resulting in the lowest CD among compared approaches.

#### 4.2. Comparisons on Real Images

We capture RGB images of real-world CAD objects using a Canon EOS R5 camera with a Canon EF 70 lens. Era3D [20] is employed for background removal and as a normal predictor to generate accurate normal maps, which are subsequently used as inputs to evaluate our method’s generalization capability on real-world CAD objects. As shown in Figure 7, despite the fact that real images possess more complex lighting and shadow effects, greater perspective distortion, and introduce larger errors in normal estimation, our method still successfully reconstructs high-quality CAD models, which demonstrates the generalization potential of our approach in handling real images.

### 5. Conclusion and Future Work

CADDreamer introduces a novel cross-domain, multi-view generation strategy, along with a geometric optimization and topological extraction process, enabling the accurate generation of B-reps with diverse primitives. However, two key limitations remain. First, the method’s performance is constrained by image quantity and resolution, sometimes failing to detect extremely fine geometric features, which leads to fitting inaccuracies and incomplete reconstructions. Second, like existing single-view reconstruction approaches, the system struggles with challenging viewpoints and complex occlusions, especially when the captured angles do not reveal all the primitives. Our future work will explore multi-view generation techniques that incorporate additional viewpoints and enhanced resolution to address these limitations.

## References

- [1] Md Ferdous Alam and Faez Ahmed. GenCAD: Image-conditioned computer-aided design generation with transformer-based contrastive representation and diffusion priors. *arXiv preprint arXiv:2409.16294*, 2024. 3, 7
- [2] Gwangbin Bae and Andrew J. Davison. Rethinking inductive biases for surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 4
- [3] Jean-Philippe Bauchet and Florent Lafarge. Kinetic shape reconstruction. *ACM Transactions on Graphics*, 39(5):1–14, 2020. 2
- [4] Tianrun Chen, Chunan Yu, Yuanqi Hu, Jing Li, Tao Xu, Runlong Cao, Lanyun Zhu, Ying Zang, Yong Zhang, Zejian Li, and Linyun Sun. Img2CAD: Conditioned 3D CAD model generation from single image with structured visual geometry. *arXiv preprint arXiv:2410.03417*, 2024. 3, 7
- [5] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023. 1
- [6] Yun Fei, Guodong Rong, Bin Wang, and Wenping Wang. Parallel L-BFGS-B algorithm on GPU. *Computers & Graphics*, 40:1–9, 2014. 6
- [7] Daoyi Gao, Dávid Rozenberszki, Stefan Leutenegger, and Angela Dai. DiffCAD: Weakly-supervised probabilistic CAD model retrieval and alignment from an RGB image. *arXiv preprint arXiv:2311.18610*, 2023. 3
- [8] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9785–9795, 2019. 7
- [9] Can Gümeli, Angela Dai, and Matthias Nießner. ROCA: Robust CAD model retrieval and alignment from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4022–4031, 2022. 3
- [10] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. *arXiv preprint arXiv:2311.04400*, 2023. 1, 3, 7, 8
- [11] Hamid Izadinia, Qi Shan, and Steven M Seitz. IM2CAD. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5134–5143, 2017. 3
- [12] Pradeep Kumar Jayaraman, Joseph G Lambourne, Nishkrit Desai, Karl DD Willis, Aditya Sanghi, and Nigel JW Morris. SolidGen: An autoregressive model for direct B-rep synthesis. *arXiv preprint arXiv:2203.13944*, 2022. 3
- [13] Adrien Kaiser, Jose Alonso Ybanez Zepeda, and Tamy Boubekeur. A survey of simple geometric primitives detection methods for captured 3D data. In *Computer Graphics Forum*, pages 167–196, 2019. 2
- [14] Kacper Kania, Maciej Zieba, and Tomasz Kajdanowicz. UCSG-NET – unsupervised discovering of constructive solid geometry tree. *Advances in Neural Information Processing Systems*, 33:8776–8786, 2020. 3
- [15] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. ABC: A big CAD model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 6
- [16] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. Mask2CAD: 3D shape prediction by learning to segment and retrieve. In *European Conference on Computer Vision*, pages 260–277, 2020. 3
- [17] Joseph George Lambourne, Karl Willis, Pradeep Kumar Jayaraman, Longfei Zhang, Aditya Sanghi, and Kamal Rahimi Malekshan. Reconstructing editable prismatic CAD from rounded voxel models. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 2, 3
- [18] Florian Langer, Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. SPARC: Sparse render-and-compare for CAD model alignment in a single RGB image. *arXiv preprint arXiv:2210.01044*, 2022. 3
- [19] Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J Mitra. Free2CAD: Parsing freehand drawings into CAD commands. *ACM Transactions on Graphics*, 41(4):1–16, 2022. 2
- [20] Peng Li, Yuan Liu, Xiaoxiao Long, Feihu Zhang, Cheng Lin, Mengfei Li, Xingqun Qi, Shanghang Zhang, Wenhan Luo, Ping Tan, Wenping Wang, Qifeng Liu, and Yike Guo. Era3D: High-resolution multiview diffusion using efficient row-wise attention. *arXiv preprint arXiv:2405.11616*, 2024. 4, 8
- [21] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. GlobFit: Consistently fitting primitives by discovering global relations. In *ACM SIGGRAPH 2011 Papers*, pages 1–12, 2011. 2
- [22] Yuanqi Li, Shun Liu, Xinran Yang, Jianwei Guo, Jie Guo, and Yanwen Guo. Surface and edge detection for primitive fitting of point clouds. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 3
- [23] Frederico A Limberger and Manuel M Oliveira. Real-time detection of planar regions in unorganized point clouds. *Pattern Recognition*, 48(6):2043–2053, 2015. 2
- [24] Cheng Lin, Tingxiang Fan, Wenping Wang, and Matthias Nießner. Modeling 3D shapes by reinforcement learning. In *European Conference on Computer Vision*, pages 545–561, 2020. 1
- [25] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3D mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2
- [26] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.
- [27] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. SyncDreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 1, 2, 3, 7, 8

- [28] Yujia Liu, Anton Obukhov, Jan Dirk Wegner, and Konrad Schindler. Point2cad: Reverse engineering cad models from 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3763–3772, 2024. [3](#), [7](#)
- [29] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9970–9980, 2024. [2](#), [3](#), [4](#), [7](#)
- [30] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020. [1](#)
- [31] Sven Oesau, Florent Lafarge, and Pierre Alliez. Planar shape detection and regularization in tandem. In *Computer Graphics Forum*, pages 203–215, 2016. [2](#)
- [32] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. *arXiv preprint arXiv:2209.14988*, 2022. [1](#)
- [33] Tahir Rabbani, Sander Dijkman, Frank van den Heuvel, and George Vosselman. An integrated approach for modelling and global registration of point clouds. *ISPRS journal of Photogrammetry and Remote Sensing*, 61(6):355–370, 2007. [2](#)
- [34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. [1](#), [2](#), [3](#)
- [35] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. In *Computer Graphics Forum*, pages 214–226, 2007. [2](#), [4](#), [5](#)
- [36] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3D generation. *arXiv:2308.16512*, 2023. [1](#)
- [37] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. [1](#), [2](#), [3](#), [4](#)
- [38] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. CRM: Single image to 3D textured mesh with convolutional reconstruction model. In *European Conference on Computer Vision*, pages 57–74, 2024. [3](#), [7](#), [8](#)
- [39] Rundi Wu, Chang Xiao, and Changxi Zheng. DeepCAD: A deep generative network for computer-aided design models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6772–6782, 2021. [2](#), [6](#)
- [40] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. InstantMesh: Efficient 3D mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. [1](#), [2](#), [3](#), [7](#), [8](#)
- [41] Xiang Xu, Joseph G Lambourne, Pradeep Kumar Jayaraman, Zhengqing Wang, Karl DD Willis, and Yasutaka Furukawa. BrepGen: A B-rep generative diffusion model with structured latent geometry. *arXiv preprint arXiv:2401.15563*, 2024. [3](#)
- [42] Dong-Ming Yan, Wenping Wang, Yang Liu, and Zhouwang Yang. Variational mesh segmentation via quadric surface fitting. *Computer-Aided Design*, 44(11):1072–1082, 2012. [2](#)
- [43] Yang You, Mikaela Angelina Uy, Jiaqi Han, Rahul Thomas, Haotong Zhang, Suya You, and Leonidas Guibas. Img2CAD: Reverse engineering 3D CAD models from images through VLM-assisted conditional factorization. *arXiv preprint arXiv:2408.01437*, 2024. [3](#), [7](#)
- [44] Fenggen Yu, Zhiqin Chen, Manyi Li, Aditya Sanghi, Hooman Shayani, Ali Mahdavi-Amiri, and Hao Zhang. CAPRI-Net: Learning compact CAD shapes with adaptive primitive assembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11768–11778, 2022. [3](#)
- [45] Long Zhang, Jianwei Guo, Jun Xiao, Xiaopeng Zhang, and Dong-Ming Yan. Blending surface segmentation and editing for 3D models. *IEEE Transactions on Visualization and Computer Graphics*, 28(8):2879–2894, 2020. [2](#)