CAD-MLLM: Unifying Multimodality-Conditioned CAD Generation With MLLM

Jingwei Xu*, Chenyu Wang*, Zibo Zhao, Wen Liu, Yi Ma, Shenghua Gao

Abstract—This paper aims to design a unified Computer-Aided Design (CAD) generation system that can easily generate CAD models based on the user's inputs in the form of textual description, images, point clouds, or even a combination of them. Towards this goal, we introduce the CAD-MLLM, the first system capable of generating parametric CAD models conditioned on the multimodal input. Specifically, within the CAD-MLLM framework, we leverage the command sequences of CAD models and then employ advanced large language models (LLMs) to align the feature space across these diverse multi-modalities data and CAD models' vectorized representations. To facilitate the model training, we design a comprehensive data construction and annotation pipeline that equips each CAD model with corresponding multimodal data. Our resulting dataset, named Omni-CAD, is the first multimodal CAD dataset that contains textual description, multi-view images, points, and command sequence for each CAD model. It contains approximately 450K instances and their CAD construction sequences. To thoroughly evaluate the quality of our generated CAD models, we go beyond current evaluation metrics that focus on reconstruction quality by introducing additional metrics that assess topology quality and surface enclosure extent. Extensive experimental results demonstrate that CAD-MLLM significantly outperforms existing conditional generative methods and remains highly robust to noises and missing points. The project page and more visualizations can be found at: https://cad-mllm.github.io/

Index Terms—Computer-Aided Design Models, Multimodal Large Language Models, Multimodality Data

1 Introduction

Computer-Aided Design (CAD) is the use of computers to aid the creation, modification, and optimization of objects. It plays a pivotal role in industrial design and manufacturing and has been widely used for architectural design, shipbuilding, automobile, and aerospace industries, etc. Classical CAD workflow usually involves the design of 2D sketches (e.g., circles, lines, splines) and 3D operations (e.g., extrusion, loft, fillet) of these 2D elements, which is a sequence of actions with fixed types but various parameters, which can be explicitly and precisely represented by text. Then, the final CAD models are shaved as boundary representations (B-Rep), which facilitates the control of the design history and modification of the models. However, current CAD software requires experts to design and modify the model, while the CAD needs to be frequently updated by communicating with the users. It is desirable to develop a toolbox with which the expert, or even the non-expert, can easily design the CAD models by using simple instructions and illustrations to make the ideas in their mind easily come true.

With the advancement of generative models, recent approaches have explored CAD generation, of which, DeepCAD [1] is a very representative one. DeepCAD leverages an autoencoder to generate CAD models from command sequence representation, but it operates exclusively within a latent space and is not initially designed for conditional generation, which could not meet the users' needs for interactive design, where the users' input can be images, textual descriptions, or point clouds. To tackle this issue, Img2CAD [2] and GenCAD [3] have been proposed to generate a CAD model based on the input images. Text2CAD [4], Query2CAD [5] have been proposed to generate a CAD model based on the text. Point2cyl [6], TransCAD [7] have been proposed to generate a CAD model based on the point cloud. However, all these methods propose different methods for conditions of different modalities. It is desirable to design a unified framework to tackle the CAD generation task with different input conditions or even multiple conditions.

On the other hand, multimodal large language models (MLLMs) have demonstrated their capability in content generation across different modalities [16], [17]. However, the use of MLLMs for CAD generation remains unexplored. While MLLMs support direct input from various modalities, meeting the requirements for conditional generation, there are two main challenges when applying MLLMs to conditional CAD generation: (1) the lack of an efficient representation that MLLMs can interpret and manipulate the CAD models, and (2) the unavailability of a large scale multimodal CAD dataset to align CAD models to the text, image, and point cloud modalities. To be specific, CAD models require a high level of specificity in terms of dimensions, connectivities, and functional requirements, while current LLMs are primarily trained in natural language. Thus, it is desirable to find a suitable CAD representation for LLM generation. Also, from the perspective of

- *Jingwei Xu and Chenyu Wang contributed equally to this work;*
- *Corresponding Author: Shenghua Gao; E-mail: gaosh@hku.hk*
- *Jingwei Xu and Zibo Zhao are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China. Email: xujw2023@shanghaitech.edu.cn, zhaozb@shanghaitech.edu.cn*
- *Chenyu Wang is with Transcengram. Email: wangchy@transcengram.com*
- *Wen Liu is with DeepSeek AI. Email: liuwen@deepseek.com*
- *Yi Ma, and Shenghua Gao are with the University of Hong Kong, Hong Kong SAR, China. E-mail: mayi@hku.hk, gaosh@hku.hk*

Dataset	Publication	CAD Model Representation‡	CAD Model Size	Input Condition
ABC [8]	CVPR 2019	B-rep	1,000,000+	Uncondition
CC3D-Ops [9]	3DV 2022	B-rep	37,000+	Uncondition
CADParser [10]	IJCAI 2023	Command Sequence	40,000+	Uncondition
DeepCAD [1]	ICCV 2021	Command Sequence	179,133	Uncondition
Fusion360 [11]	TOG 2021	Command Sequence	8,625	Uncondition
SketchGraphs [12]†	Arxiv, 2020.7	Command Sequence	15,000,000+	Uncondition
Free2CAD [13]	SIGGRAGH 2022	Command Sequence	210,000+	User Drawing
Img2CAD [2]	Arxiv, 2024.7	Command Sequence	4,574	Single Image
OpenECAD [14]	Arxiv, 2024.6	Command Sequence	200,000+	Single Image
ABC-mono [15]	Arxiv, 2024.10	Command Sequence	208,853	Single Image
Query2CAD [5]	Arxiv, 2024.5	Python Macro	57	Text
Text2CAD [4]	NeurIPS 2024	Command Sequence	158,000+	Text
Omni-CAD(Ours)		Command Sequence	453,220	Multi-view Images/Text/Point

TABLE 1: Comparison of previous datasets and our proposed dataset. Our proposed Omni-CAD dataset is the only dataset available that simultaneously supports multi-view images, text, and point cloud conditioned data for CAD modeling. Notably, our dataset includes a large-scale collection of CAD models, second only to the ABC [8] and SketchGraphs [12] datasets. †: SketchGraphs [12] focuses on the 2D CAD sketches instead of the 3D CAD models. ‡: Command Sequence Representation can convert to the B-rep representation.

user-system interaction, how to bridge CAD models with text, image, and point cloud, these three modalities into a unified framework remains a significant challenge. Each of these modalities represents information in vastly different formats. The text describes concepts and attributes, images capture visual details, point clouds represent spatial data, and CAD models require precise geometric and structural definitions. From the dataset side, the current CAD datasets with command sequence, including the Fusion360 [11] and DeepCAD [1], are on a relatively small scale (8,625 and 179,133, respectively). A detailed comparison of datasets is provided in Tab. 1. More importantly, current CAD datasets do not contain paired multimodal CAD data. To support the training of MLLMs, it is desirable to have an even larger scale dataset with CADs paired with different modalities to support the conditional CAD generation with MLLMs.

To address these challenges, we present CAD-MLLM to unleash the potential of MLLMs for CAD generation conditioned on multimodality inputs. Given that the primitive boundary representation of CAD models is non-sequential and unsuitable for an auto-regressive pipeline, motivated by the DeepCAD [1], we instead utilize the command sequences, vectorizing them into a condensed sequential data flow that is more efficient for MLLMs to learn from. Combined with multimodality data, our model is capable of constructing complete CAD models by conditioning on text, images, point clouds, and any combination of them. When multiple modalities are input as a combination, our multimodal model demonstrates its strength by supplementing missing or suboptimal information from one data modality with input from another. To support our methodology, we propose a data annotation pipeline combined with a data augmentation method to generate a new multimodality-conditioned CAD dataset named Omni-CAD. Omni-CAD includes text descriptions, multi-view images, point clouds, and their corresponding constructive modeling command sequences. Omni-CAD reaches 453,220 models after data augmentation.

To evaluate the quality of the generated CAD models, although some previous CAD reconstruction works [18], [19], [20], [21] have proposed some well-established metrics for performance evaluation, such as utilizing sampled point clouds and patch topology to assess reconstruction fitting quality and patch structure fidelity, these metrics overlook an important nature of the CAD model: the overall topology quality of the CAD model in its final mesh representation. As a remedy, we propose three topology metrics, **Segment Error (SegE)**, **Dangling Edge Length (DangEL)**, and **Self-Intersection Ratio (SIR)**, to evaluate the topological quality of the final generated model. Additionally, since CAD models use boundary representation to form closed surfaces, we introduce **Flux Enclosure Error (FluxEE)** to quantify how well the generated model encloses space. These metrics are broadly applicable to general models in mesh representation as well.

We conduct extensive evaluations on the proposed benchmark, and our experiments demonstrate that our method achieves state-of-the-art performance compared to other CAD generation methods and shows high robustness under various data flaws at the inference stage.

Our contributions can be summarized as follows:

- We propose a unified multimodality-conditioned CAD generation method by leveraging the pretrained MLLM, and the condition can be text, images, point clouds, and any combination of these modalities.
- We present a data annotation pipeline and create a large-scale dataset, named Omni-CAD, the first multimodality CAD dataset includes constructive modeling command sequences and the corresponding textual descriptions, multi-view images, and point cloud data.
- We introduce four new evaluation metrics, namely, SegE, DangEL, SIR, and FluxEE, to evaluate the topological quality and enclosure of the generated CAD models, respectively.
- Extensive experiments show that our method demonstrates state-of-the-art performance over the baseline methods and high robustness under data flaws during inference.

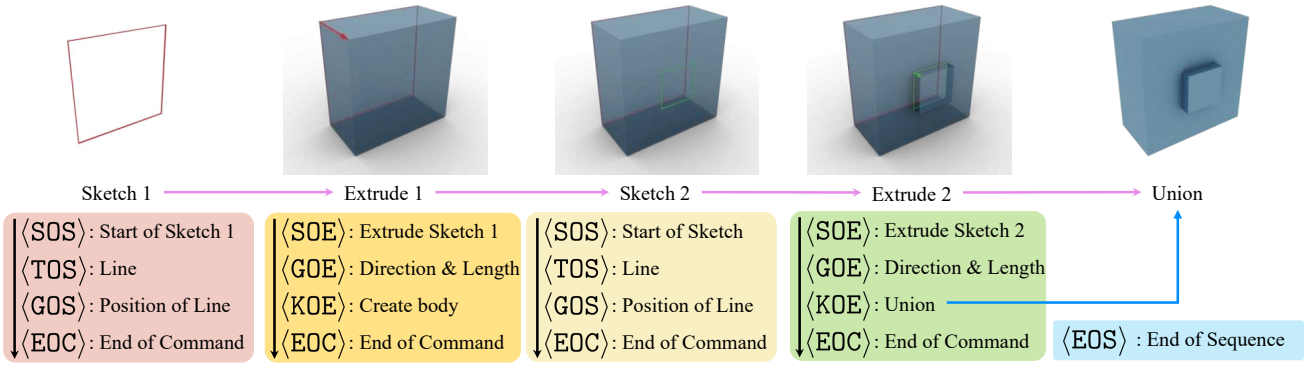


Fig. 1: A simple example about the construction process of a CAD model with command sequence representation. Starting with a *sketch* operation on a chosen 2D plane, the *extrusion* operation then "drags" this 2D *sketch* into a 3D solid volume. Further editing requires another *extruded* 3D solid volume. Subsequently, the *union* will "merge" these two 3D solids into a single integrated solid. Other boolean operators from Constructive Solid Geometry (CSG) support the construction of more complex geometries. As a result, this CAD model can be represented with these command sequences.

2 Related Work

In this section, we will first review existing CAD generation methods based on different CAD presentations, namely, Boundary Representation (B-rep) [22], Constructive Solid Geometry (CSG) [23], and Construction Command Sequence. Further, we will also review some MLLM related works.

2.1 B-rep Based CAD Generation

B-rep 3D models are depicted as graphs, incorporating both geometric primitives (e.g., parametric curves and surfaces) and topological primitives (e.g., vertices, edges, and faces) that trim and stitch surface patches to form solid models [22]. Works about B-rep classification and segmentation have used various methods around the graph property, including graph neural networks [11], [24], [25], custom convolutions [26], and hierarchical graph structures [27], [28], [29].

Some previous approaches have utilized predefined template curves and surfaces [18], [30], [31], [32], [33] for B-rep generation. Specifically, PolyGen [34] uses pointer networks [35] with Transformers [36] to generate *n-gon* meshes, a special case of B-rep models characterized by planar faces and linear edges. SolidGen [25] and BrepGen [37] can generate the entire B-rep models. SolidGen [25] first synthesizes the vertices and then constructs them with the edge topology. BrepGen [37] progressively denoises the faces, edges, and vertices utilizing Diffusion models [38]. Although B-rep is a direct representation of the boundary of the CAD model, it requires topological consistency, such as avoiding gaps and overlaps, inevitably introducing additional complexity to the CAD generation.

2.2 CSG Based CAD Generation

In CAD design, CSG is a widely-used technique for generating complex 3D shapes by combining solid primitives with boolean operators, like *union*, *intersection*, and *subtraction*, to form a CSG tree finally. Recent CSG-based methods have concentrated on reconstructing 3D shapes as assemblies of primitives without relying on the ground truth CSG tree [39], [40], [41], [42]. Meanwhile, CSG has been extensively leveraged in "shape programs" [43] with neural guidance [44], [45], [46] and without [47], [48], [49], [50]. Although the CSG tree can be converted into a B-rep model, the parametric CAD modeling [51] with a sequence of 2D *sketches* to be *extruded* to 3D is still the primary paradigm for CAD designing and portable parametric editing.

2.3 Command Sequence Based CAD Generation

Recent available large-scale datasets [1], [11] for parametric CAD modeling have facilitated the thriving of construction command sequence generation. Learning-based methods are investigated to utilize the history of the construction command sequence [1], [11], [52], [53], [54] and the constraints of sketches [12] for generating engineering sketches and solid models. The generated sequences can be parsed using a solid modeling kernel to obtain an editable parametric CAD file. Furthermore, some works can generate the sequences or conduct reverse engineering conditioned on the sketching data [55], [56], images [3], [57], voxel grids [58], point clouds [6] and target B-reps [59] or without sequence guidance [60]. However, there is a notable absence of generation methods conditioned on text inputs, as well as those that handle more complex multimodal conditions. Additionally, a multimodal command sequence dataset for supporting advanced generation methods is lacking.

2.4 Multimodal Alignment and Multimodal Large Language Models (MLLMs)

Prior to MLLMs, many works, such as CLIP [61], have explored multimodal alignment. With the remarkable progress of LLMs [62], [63], [64], [65], [66], many efforts [67], [68], [69], [70] empowered the LLMs to the vision tasks by bridging with the pretrained visual encoders, and then downstream tasks have reached significant milestones. On this basis, some specialized models [71], [72], [73] in various vertical domains are being progressively explored. These tailored models aim to address specific challenges and enhance capabilities within each domain.

Meanwhile, some works explore the application of generating CAD models. The concurrent work Img2CAD [2]

leverages VLMs to predict the global discrete structure and then conditioned on the structure, along with the semantics, to predict the continuous attributes. Another concurrent work Text2CAD [4] leverages both VLMs and LLMs for data annotation and uses Transformer [36] structure to generate the full CAD sequence in an auto-regressive way. Some other recent works [14], [74], [75] also investigate the utilization of VLM in CAD tasks. Compared to these works, our work supports not only image modality but also point and text modality simultaneously, and the MLLMs are directly empowered to predict the structure and attributes.

3 Command Sequence Based CAD Representation

From a user-interaction perspective, the popular industrial standard for the creation of CAD models can be described as the sequence of operations performed by CAD software (e.g. OpenCascade [76], Fusion360 [77], and Solidworks [78]). To create a solid shape, a user first needs to create a closed curve profile as a 2D *sketch*, and then *extrude* it into a 3D solid shape. To further create complex surfaces or objects, CSG [23] enables the user to combine simpler objects by applying boolean operators, such as *union*, *intersection*, and *subtraction*, which allows for the generation of visually intricate objects through the combination of a few primitive shapes.

Given a CAD command sequence, it can be automatically transformed into a B-rep representation of a CAD model through a CAD modeling library, like PythonOCC [79]. Following DeepCAD [1], we represent a sketch-and-extrude CAD model using a sequence with five types of tokens, **Start and End-command token**, **Topology token**, **Geometry token**, **Kind-of-extrusion token**, **End-of-sequence token**, which are aligned with notation in Tab. 2:

As an example, a CAD model can be constructed with the command sequence through the *union* of two *extruded* solid shapes, which is illustrated in Fig. 1.

Below, we define the *sketch* and *extrusion* operations in the format of commands consisting of various continuous attributes in a practical way.

Sketch. According to the CAD terminology, a *profile* is a closed region consisting of one or several loops, where the curve commands in each loop are concatenated. Thus, except the start point of the first curve is the origin of the plane, the remaining curves' start points are the endpoints of the predecessor curves typically. In practice, for the ⟨TOS⟩, we consider the most three types of curve commands, *lines* (L), *arcs* (A), and *circles* (R).

The corresponding ⟨GOS⟩ geometry of each type of ⟨TOS⟩ curve is defined as follows:

- $L : (x, y)$, where (x, y) defines the endpoint of a line.
- $A : (x, y, \alpha, f)$ which defines an arc with the endpoint (x, y) and sweep angle α. f refers to the counter-clockwise flag.
- $R : (x, y, r)$, where (x, y) is the center of an circle with a radius r.

Extrusion. As mentioned above, the extrusion command serves a dual purpose; it needs to provide the information not only on how to transform a sketch profile into a 3D shape by extending it along a specified path but also on the spatial relationship and merge operation of the newly formed 3D shape with other existing 3D shapes to form the final 3D shape. Therefore, the extrusion command can be defined as $E : (\theta, \phi, \gamma, x, y, z, s, e_p, e_n, b, u)$, where (θ, ϕ, γ) are the three Euler angles determining the extrusion orientation, (x, y, z) refers to the origin of the sketch plane, s represents the scale factor. Besides that, e_p, e_n denotes the extrusion distance towards the positive direction and negative direction respectively. The parameters related to the geometry of *extrusion* operation form ⟨GOE⟩. Additionally, b and u are two type arguments specifying the volume boolean type (e.g. *joining*, *intersecting*, *cutting*) and extrusion type (e.g. *one-sided*, *symmetric*, *two-sided*), which correspond to ⟨KOE⟩.

By integrating these two operations, each sequence can be vectorized as 16 distinct variables. To save the length of the command sequence without affecting the information of the command, instead of setting unused parameters in the command sequences to be -1 as DeepCAD [1], we use a particular Place Holder Token, combining with other tokens acting as the End-of-command token, ⟨EOC⟩. Specifically, when the last few variables of a sequence are all the placeholder tokens, these placeholder tokens will act as an ⟨EOS⟩ to indicate the end of the current command.

Notation	
Sketch Related Token:	
⟨SOS⟩	**Start-of-sketch token**: Denotes the start of a *sketch* operation.
⟨TOS⟩	**Topology-of-sketch token**: Specifies the type of curve used in the *sketch* operation. This token indicates whether the curve is a line, arc, or circle.
⟨GOS⟩	**Geometry-of-sketch token**: Contains the coordinates of points and geometric parameters that define the shape of the sketch. These tokens provide the necessary geometric information for constructing the curves. Note that every model is normalized within a cube range from $[-1, 1]^3$ before being quantized to 256 levels.
Extrude Related Token:	
⟨SOE⟩	**Start-of-extrusion token**: Denotes the start of a *extrusion* operation.
⟨GOE⟩	**Geometry-of-extrusion token**: Contains parameters related to the *extrusion* process. These parameters include the direction, type, and length of the extrusion.
⟨KOE⟩	**Kind-of-extrusion token**: Identify the associated volume boolean operations after creating the solid of this *extrusion* operation. The volume boolean operations include *union*, *intersection*, or *subtraction* with the current CAD design, which will generate a more complex solid.
Ending Related Tokens:	
⟨EOC⟩	**End-of-command token**: Denotes the end of a operation command.
⟨EOS⟩	**End-of-sequence token**: Denotes the end of the entire command sequence.

TABLE 2: Notation for sequential tokens. This table details the essential elements for constructing a sketch command, an extrusion command, and an entire command sequence.

4 Creation of Our Large-Scale Multi-modal CAD Dataset

Several datasets for CAD modeling are publicly accessible. The ABC dataset [8] comprises 1 million CAD designs

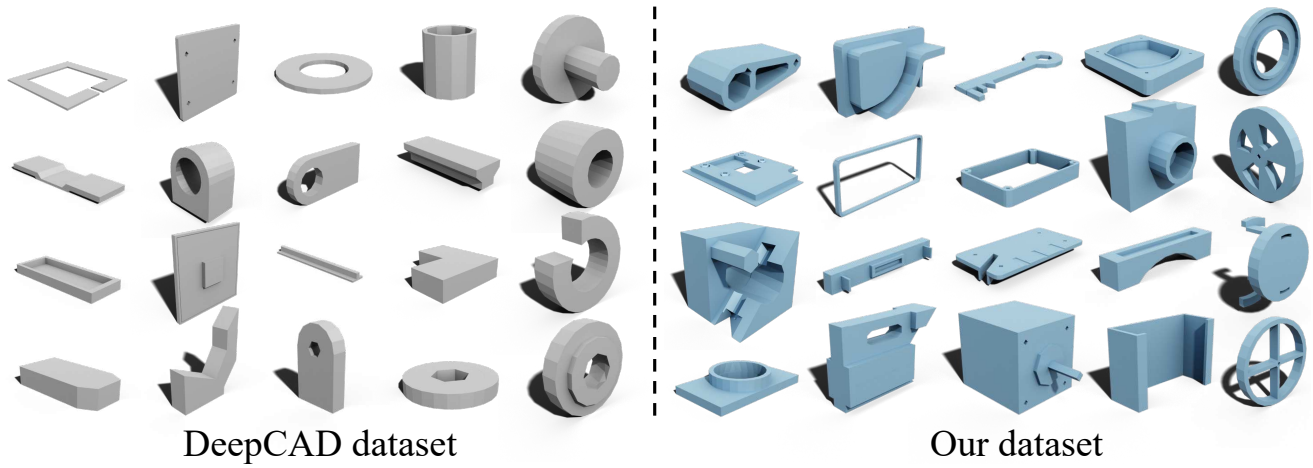


Fig. 2: Qualitative comparison between our CAD command sequence dataset and DeepCAD [1] dataset. According to Sec. 4, DeepCAD dataset is part of our created dataset. In the visualization of our dataset, we **exclude** the CAD models' IDs that have been included in the DeepCAD dataset. The extension part of our dataset contains more complex and realistic models with more details. **Best viewed zoomed in.**

sourced from Onshape [80], a cloud-based platform for product design. However, these CAD designs are initially provided in B-rep form, which lacks the detailed information needed to recover the construction operations. In contrast, the Fusion360 Reconstruction Dataset [11] offers CAD modeling sequences created by human designers. Despite this, the dataset contains only $8,625$ CAD designs, which is insufficient for training a generalized generative model. Apart from the insufficient scale of the datasets, current datasets only provide information related to CAD models. To lower users' barriers in creating CAD models and enable non-experts to bring their ideas to life through arbitrary multimodal conditions, a dataset with corresponding textual descriptions, multi-view images, and point cloud data alongside CAD models is essential. However, such a dataset does not currently exist.

Therefore, we create a new large-scale multimodal CAD dataset that simultaneously provides CAD command sequences and corresponding data in three modalities, which we hope will inspire and accelerate advancements in future research.

4.1 CAD Command Sequences Generation and Augmentation

Originating from the ABC dataset [8], we adopt DeepCAD's [1] approach to process CAD designs, utilizing Onshape's developer API [81] and parsing with Onshape's FeatureScript [82].

As shown in Sec. 3, the command sequence representation method focuses on the 2D plane and the process to transform into a 3D shape body, not including edge or face primitives that are required by some specific commands, such as *chamfer* and *fillet*. In CAD modeling, the *chamfer* and *fillet* operations are commonly employed to mitigate sharp edges and corners in engineering and design contexts. In detail, a *chamfer* replaces a sharp directional change with an angled slope, whereas a *fillet* introduces a smooth, curved transition between two surfaces. Due to the vectorized sequence representation limitation, this parsing method would remove all CAD designs containing *chamfer* or *fillet* operations.

However, unlike DeepCAD, which directly removes all the designs with any of these two operations, we individually remove each *chamfer* and *fillet* operation and retain the CAD design if it maintains a complete topology. As a result, we initially collect a dataset of $275,717$ models, nearly $1.54\times$ the $179,133$ designs reserved by DeepCAD.

We additionally augment our data by extracting intermediate CAD designs after each *extrusion* operation. For example, a CAD design with 7 *extrusion* operations can be augmented into 7 CAD designs. In the end, we collect a total of $453,220$ augmented CAD command sequence data. Note that to ensure fair testing and prevent the augmented data's interrelations from providing undue advantages, we divide the dataset into training and testing sets before we apply the data augmentation strategy exclusively to the training set.

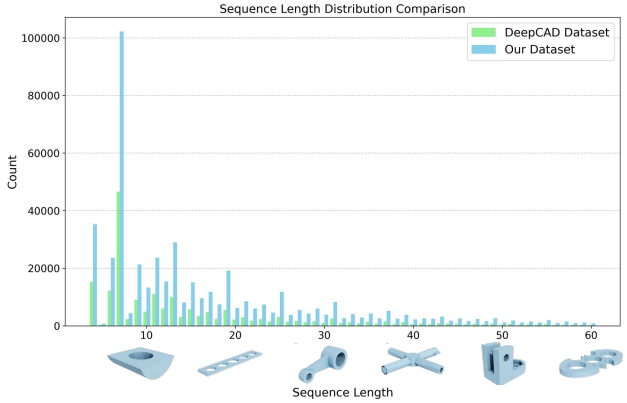
Fig. 2 visualizes the qualitative comparison of our dataset and DeepCAD [1] dataset. The statistics of our extension in both challenging sequence length and challenging *extrusion* operation count can be observed from Fig. 3.

4.2 Conditional Data Generation

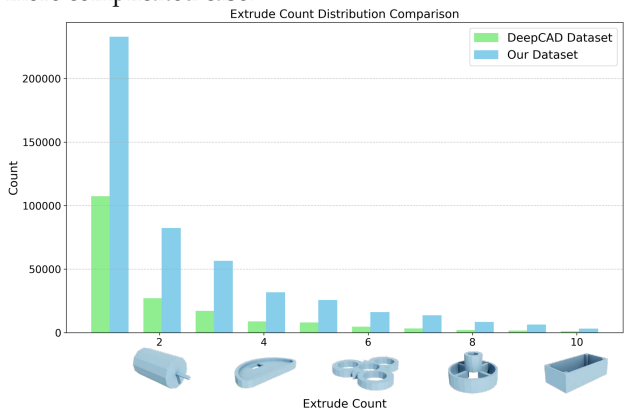
In addition to generating the vectorized command sequence representations for CAD models, our more important objective is to address the current gap in datasets that lack multimodal information corresponding to CAD models, such as images, point clouds, and textual descriptions.

For each CAD model, we render multi-view images from eight fixed perspectives. For the point cloud data, we randomly sample points and record their corresponding normal information.

For the textual description of each CAD design, since there is currently no effective method to directly input a CAD representation into an MLLM and achieve good caption results, we use previously rendered multi-view images as input for the MLLM. Due to budget constraints and considering the quality of generated textual descriptions, the inference speed of the MLLM, and whether the model



(a) The statistical comparison between our dataset and DeepCAD [1] dataset over the command sequence length per CAD model. The longer sequence length indicates the more complicated case.



(b) The statistical comparison between our dataset and DeepCAD [1] dataset over the *extrusion* operation counts per CAD model. The more *extrusion* operation count indicates the more challenging case. **Best viewed zoomed in.**

Fig. 3: The statistical comparison between our dataset and DeepCAD [1] dataset. The statistics are conducted **before data augmentation**. The charts indicate that our dataset extends the data over a wide range of sequence counts and *extrusion* operation counts with more challenging cases.

supports multiple images as input, we leverage the open-sourced MLLM InternVL2-26B [83], [84] and randomly select four view images as input for each CAD design to generate high-quality text captions. The prompt is as follows: "*These are the rendering images from 4 views of a CAD model. Please describe these images with one caption, and mainly focus on the shape and appearance of the foreground while ignoring the details of the background.*". To standardize the format of the textual descriptions, we include format constraints in the prompt, requiring all outputs to begin with "*Generate a CAD design with* ". Some examples of conditioned data are provided in the supplementary.

5 METHOD

Besides the general CAD model's representation B-rep, recent works [1], [52], [53], [56] show CAD command sequences are able to utilize the history of CAD modeling sequences and constraints on the sketches. We present our CAD-MLLM, an MLLM model tailored for CAD generation based on modeling sequences. CAD-MLLM supports cross-modal inputs, including text, image, and point cloud, as conditions for generating novel CAD models.

5.1 CAD-MLLM Architecture

As shown in Fig. 4, the proposed CAD-MLLM consists of three modules: visual data alignment, point data alignment, and the large language model. Notably, as text input data is directly fed into the LLM for embedding extraction, there is no need for an additional text alignment module.

- **Visual Data Alignment.**
  Given the input multi-view images $X_v = \{X_v^1, X_v^2, \ldots, X_v^k\}$ where k specifies the number of views, the vision encoder g_v extracts independent visual features from each image. These features are then concatenated into a unified representation $H_v \in \mathbb{R}^{k\times(1+L_s)\times d_s}$, where $1+L_s$ denotes the length of tokens, which includes the class head token as well as the patch tokens, and d_s is the dimension. Drawing inspiration from previous perceiver-based transformer architectures [85], we implement a cross-attention layer to integrate the information from the k multi-view images information contained in the input H_v into a learnable query token $Q \in \mathbb{R}^{1\times L_q\times d_q}$, where L_q and d_q is the length and dimension of token Q. Additionally, an image projection layer f_ϕ is employed to project the visual signals into the feature space of the pretrained LLM where ϕ denotes the parameters to be learned in the projection layer.

$$\begin{aligned} H_v &= \text{Con}(g_v(X_v)) \\ E_v &= f_\phi(\text{CA}(Q, H_v)) \end{aligned} \tag{1}$$

- **Point Data Alignment.**
  Similar to visual inputs, when provided with point cloud data X_p, a point encoder g_p is used to extract features. These features are then projected into a feature space comprehensible by the LLM through a linear layer f_γ where γ denotes the parameters to be learned in the point projection layer.

$$E_p = f_\gamma(g_p(X_p)) \tag{2}$$

- **LoRA based Large Language Model Finetuning.**
  Large language models serve a dual purpose in our approach. On one hand, for the textual description X_l of the input CAD model, we follow Vicuna's method [64], [86], [87], [88] by utilizing a BPE tokenizer [89] to obtain text embeddings E_l. On the other hand, we input the concatenated features of the conditioned modalities into the large language model, which is tasked with predicting the sequence of commands for the CAD model as the output. To optimize our model while minimizing the number of learnable parameters, we implement Low-Rank Adaptation (LoRA) [90] to finetune an open-sourced LLM (Vicuna-7B [86]), parameterized by δ.

5.2 Training Objective

We leverage the pretrained visual encoder g_v and point encoder g_p and keep them frozen. The overall objective,

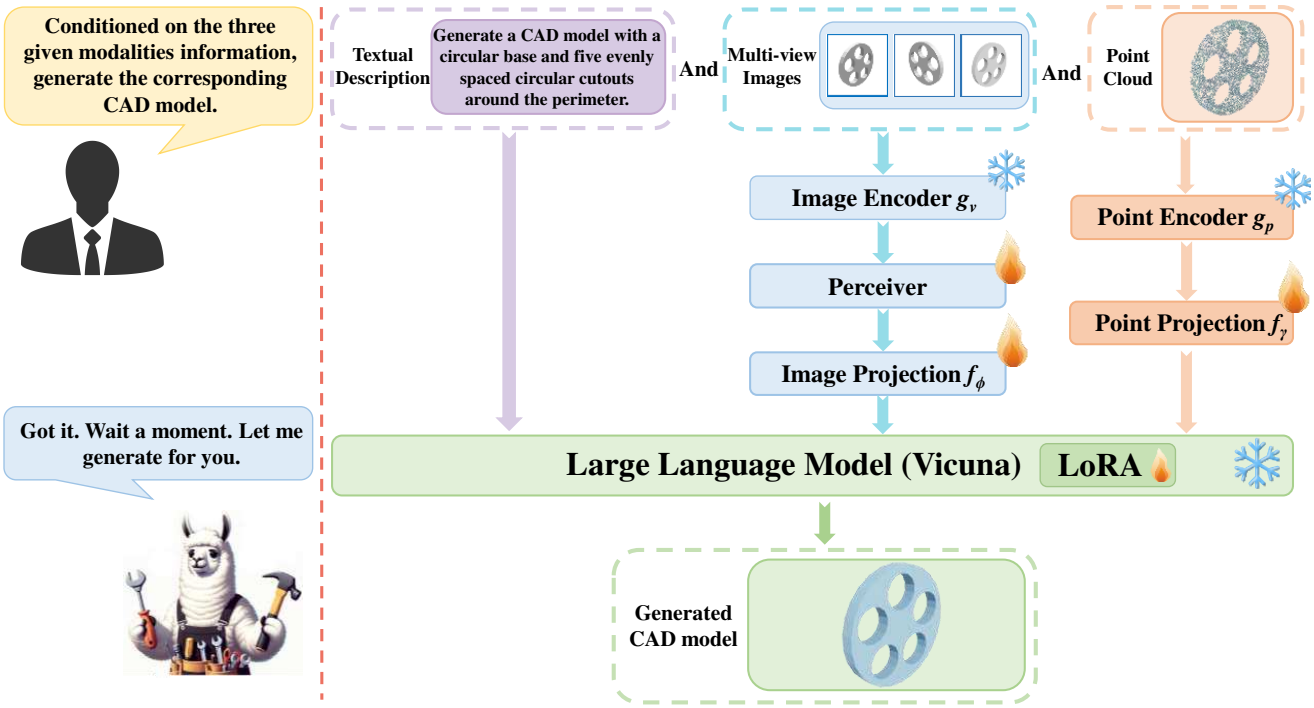


Fig. 4: **Our network architecture**. The network could process three single modalities of information of input or any combinations of them, each uniquely color-coded. We consider the most complex combination of modalities, where three different inputs are provided simultaneously. Except for the textual descriptions, each modality is first processed through its corresponding frozen encoder before being further integrated. Subsequently, they are passed through a trainable projection layer, aligning them within a unified language feature space. The fine-tuned Large Language Models (LLMs), augmented with Low-Rank Adaptation (LoRA), then process a combination of the prompt and the projected embeddings, enabling the accurate generation of CAD models.

as shown in Fig. 4, is to train the visual perceiver, image projection layer f_ϕ, point projection layer f_γ and the LoRA δ. We denote the trainable parameters as $\theta = \{\eta, \phi, \gamma, \delta\}$ where η is the parameter for the visual perceiver.

Inspired by [17], [91], we adopt a curriculum-based progressive training strategy, gradually introducing modalities in the following order: textual descriptions, point clouds, and multi-view images. The newly introduced modalities are randomly combined with existing ones to form various multimodal input configurations, allowing for comprehensive training across diverse input scenarios.

Considering the most complex case, when the user inputs the text description X_l, multi-view images X_v and point cloud data X_p as condition at the same time, as mentioned before, the visual data alignment module extracts the image feature E_v, point data alignment module extracts the point embedding E_p, and the textual embedding E_l is obtained by the BPE tokenizer. The language modeling (LM) loss is adopted to supervise the training of CAD-MLLM:

$$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{L} \log P_\theta(y_{i,t}|y_{i,<t}, E_v, E_p, E_l). \tag{3}$$

where $y_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,L}\}$ is the predicted response sequence with length L for the i^{th} input.

6 Experiments

6.1 Experimental Setup

6.1.1 Datasets

We use our multimodal CAD dataset for training and evaluation, which involves $453,220$ command sequences that are vectorized into the specific data flow we use. It also contains multimodal data (text/multi-view images/point clouds) for each vectorized augmented data for our multimodal training. We divide our Omni-CAD dataset into training and testing sets in a 9:1 ratio, with $425,726$ pairs of data used for training and $27,494$ for evaluation.

6.1.2 Training Details

We implement CAD-MLLM with PyTorch [92] and train it across 16 NVIDIA H800 80G GPUs for 20 epochs, taking approximately 47 hours. We employ an AdamW [93] optimizer with a learning rate 2e-5 and a linear decay. The dropout rate is set to 0.1, and the batch size is 8192, using a micro-batch size of 1 and 512 gradient accumulation steps. The maximum sequence length is 1024. For the large language model component, we utilize Vicuna-7B [86]. The DINO v2 [94], [95] is used as the visual encoder, and Michelangelo [96] is used as the point cloud encoder. Due to computational resource limitations, particularly when handling the most complex multimodal inputs, we limit the number of multi-view images to 2 in this work. As

mentioned in Section 5.2, a curriculum-based progressive training strategy is introduced during the training process. When given a batch of data, it is essential to pre-determine the modality information carried by each data sample. This modality selection is made randomly and with equal probability from the available combinations for the current phase. As a result, the chosen modalities for each data sample can vary, potentially consisting of either a single modality or a combination of multiple modalities. Notably, for the image inputs, since each CAD model is rendered from eight different viewpoints, consequently, two images are randomly sampled from these eight rendered views to serve as input.

6.1.3 Baselines

Our method can generate CAD command sequences with multimodal conditions. Therefore, we conduct our experiments on different tasks.

Point Clouds Conditioned CAD Generation: Since the point clouds condition offers a precise 3D reference for the target CAD model, we assess the reconstruction capability of our generation model against several baseline methods, including two different kinds of techniques: "B-rep"-based reconstruction and "command sequence"-based generation:

1) **"B-rep"-based reconstruction baselines**: We compare our method with "B-rep"-based point clouds CAD reconstruction baselines ParSeNet [18], ComplexGen [19], Point2CAD [20] and NVDNet [21]. Notice that these reconstruction methods target the B-rep reconstruction of the CAD models, which is different from the CAD command sequence. In the following comparisons, we can roughly consider the conditional generation task of our method based on point clouds as the point cloud reconstruction task.
2) **"Command Sequence"-based generation baseline**: We additionally compare our method with "Command Sequence"-based CAD generation baseline DeepCAD [1] on point clouds reconstruction. We conduct the point clouds conditioned DeepCAD with official implementation and with our dataset, using PointNet++ [97] to encode and embed the point cloud to the latent vector.

Image Conditioned CAD Generation: To the best of our knowledge, currently, there is no open-sourced "image-to-CAD" baseline to compare. Instead, we compare with the "image-to-mesh" baselines. As mentioned in Sec. 5.1, our multimodal model is able to generate a CAD model with multi-view images as a condition. We select the methods that support the multi-view images as a condition, including the InstantMesh [98] and SpaRP [99], for performance comparison.

Text Conditioned CAD Generation: To the best of our knowledge, currently, there is no open-sourced "text-to-CAD" baseline to compare. Instead, we compare with the "text-to-mesh" baselines, Michelangelo [96] and Tripo [100].

6.1.4 Evaluation Metrics

We follow the metrics in existing work for **CAD reconstruction**, and we additionally propose four new metrics covering the aspect of **CAD topology** and **model enclosure** to quantify the quality of the generated models better.

CAD reconstruction metrics:

Following previous reconstruction methods' evaluation [20], [21], we compare the Chamfer Distance (Chamfer) and F-score with a 0.05 threshold. Additionally, we compare the Normal Consistency (Normal C) between the ground truth model and the reconstructed/generated model following the evaluation of [101], [102]. Note that the objects are normalized to $[-0.5, 0.5]^3$ for reconstruction evaluation.

CAD topology metrics:

The reconstruction metrics mentioned above ignore the reconstructed model's topology quality and only focus on the point cloud. This oversight neglects critical topological information and fine-grained details inherent in CAD representations. Though Complexgen [19] proposed patch-to-patch topology accuracy metrics for measuring structural fidelity, it just evaluates patch-to-patch topology and ignores the overall topology quality as an assembled CAD model in mesh representation. To address this, we propose three additional metrics to better evaluate the generated CAD model's topology quality. We clarify that we treat edges in the mesh representation as connectivity descriptions. We define *"two nodes in the same segment"* as an edge connecting them. Additionally, we wish to prevent non-manifolds from being used for the reconstructed models. We call the edges that are only bounded by one face, as *"dangling edges"*, which will lead to a non-manifold structure. GeometryCentral [103] and CGAL [104] are used in our implementation.

1) **Segment Error (SegE)** measures the fidelity of the topology from the segment aspect. We denote $S(\cdot)$ as the segment number among all nodes in a mesh.
   The SegE of the CAD model is defined as follows:
$$\text{SegE}(\hat{G}) = \frac{|S(\hat{G}) - S(G)|}{S(G)} \tag{4}$$
   where G is the ground truth model and $\hat{G}$ is the generated model.
2) **Dangling Edge Length (DangEL)** measures the quantity of the non-manifold structure. For arbitrary mesh, the dangling edges are the edges that are only bounded by one face, which harms the manifold structure. We locate the dangling edges by executing a half-edge [105], [106] traversal over the whole mesh and then detecting the edges only accessed once. Finally, we sum up the length of all dangling edges in a mesh as the evaluation metric.
3) **Self-Intersection Ratio (SIR)** measures the ratio of the self-intersected faces among all faces. A mesh with self-intersections also does not meet the requirements of a manifold. We compute the number of self-intersected faces and then divide the total number of faces.

Model enclosure metric:

Additionally, the CAD models utilize boundary representation to form closed surfaces. Besides evaluating topology quality, the enclosure of the generated models is also an important aspect to consider. For a general continuous closed surface, according to the Gauss's Divergence Theorem [107], for a vector field $\mathbf{F}$, the flux through a closed surface is equal to the volume integral of the divergence of $\mathbf{F}$ over the region enclosed by the surface.

$$\oint_S \mathbf{F} \cdot \mathbf{n}\, dS = \int_V \nabla \cdot \mathbf{F}\, dV \tag{5}$$

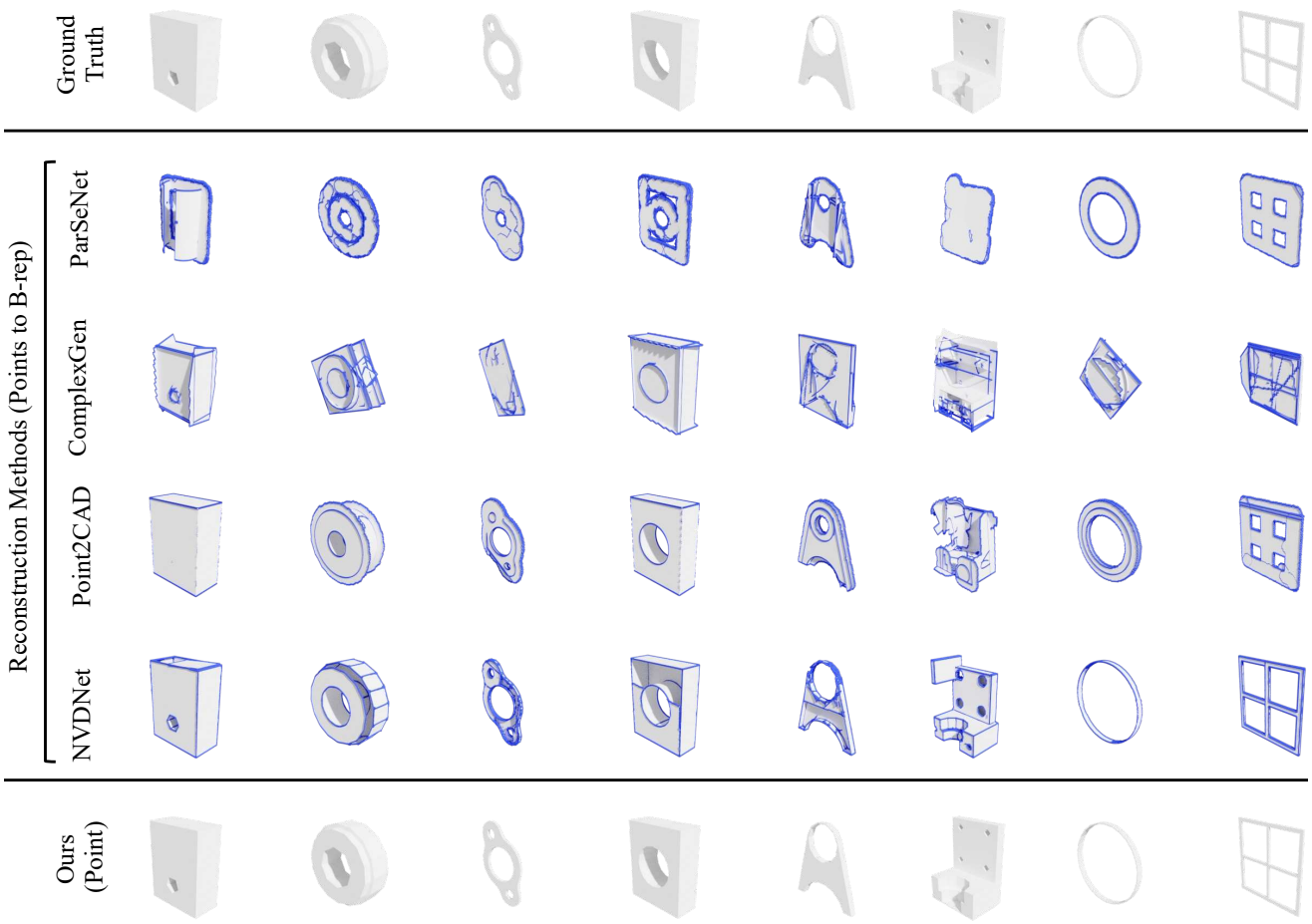


Fig. 5: We present qualitative point-based reconstruction results on our dataset and compare our generative method with the point-based B-rep reconstruction baseline. Blue lines highlight the dangling edges in the reconstructed model. Our method produces high-fidelity reconstructed results. Most of our reconstructed results are strict manifolds and do not have dangling edges (do not have blue lines). The results of the comparison of reconstruction baselines show that they have lots of dangling edges. This figure illustrates that our method outperforms from the topological aspect.

where S is the closed surface, V represents the volume enclosed by S, dS represents the surface differential, $\mathbf{n}$ represents the outward-pointing unit normal vector of the surface differential, dV represents the volume differential.

For simplicity, we define $\mathbf{F}$ as a constant vector field $(1, 1, 1)$, since the divergence of a constant vector field is always zero.

$$\oint_S \mathbf{F} \cdot \mathbf{n}\, dS = \oint_S (n_x + n_y + n_z) dS = \int_V \nabla \cdot \mathbf{F}\, dV = 0 \quad (6)$$

where n_x, n_y, and n_z represent the components along the x, y, z axes of the unit normal vector $\mathbf{n}$.

For the discrete computation, the discretization of an ideal closed surface suffices the following:

$$\oint_S (n_x + n_y + n_z)\, dS \approx \sum_{i=1}^{N} (n_{i,x} + n_{i,y} + n_{i,z})\, dS_i \quad (7)$$

where N is the total number of discrete meshes, and dS_i is the area of the i-th discrete mesh element. $n_{i,x}$ represents the x-component of $\mathbf{n}$ at the i-th discrete mesh element, with similar definitions for other axes.

For an ideal closed surface in discrete form, the discrete integral approximation becomes

$$\sum_{i=1}^{N} (n_{i,x} + n_{i,y} + n_{i,z})\, dS_i = 0 \quad (8)$$

The enormous flux of this constant vector field through the discrete mesh indicates that this mesh is far from the ideal closed surface, which is not expected for our generated model. We define additional metrics as:

4) **Flux Enclosure Error (FluxEE)** measures the degree of enclosure for the mesh. The FluxEE of the CAD model is defined as follows:

$$\text{FluxEE}(\hat{G}) = \left| \sum_{i=1}^{N} (n_{i,x} + n_{i,y} + n_{i,z})\, dS_i \right| \quad (9)$$

6.2 Results

6.2.1 Point Conditioned CAD Generation

We conduct our training for this point-based CAD reconstruction experiment using only point clouds as input, aligning our methodology with that of the selected baselines, to ensure a fair comparison.

Point-based CAD Reconstruction		Chamfer(×100)↓	F-score(×100)↑	Normal C(×100)↑	SegE↓	DangEL↓	SIR(%)↓	FluxEE(×100)↓
Reconstruction Methods	ParSeNet [18]	4.59	42.56	46.83	10.92	78.82	13.87	398.524
	ComplexGen [19]	1.65	86.12	64.82	17.72	55.32	22.57	115.918
	Point2CAD [20]	1.25	89.85	65.90	15.82	9.23	11.38	97.453
	NVDNet [21]	**0.82**	**98.94**	**93.94**	37.22	47.97	2.60	14.550
Generation Methods	DeepCAD(Point) [1]	4.51	71.83	63.68	8.98	1.26	5.73	0.347
	Ours(Point)	**1.85**	**90.88**	**79.71**	**1.66**	**0.46**	**1.31**	**0.044**

TABLE 3: The quantitative results on point-based reconstruction tasks. Our method's performance is comparable to some "B-rep"-based reconstruction methods in "point cloud"-based reconstruction metrics (Chamfer, F-score, Normal C). However, it significantly outperforms these methods in our proposed topological metrics (SegE, DangEL, and SIR) and the enclosure metric (FluxEE). Moreover, our method consistently surpasses the command sequence-based generation baseline across all evaluated metrics.

Tab. 3 presents a comparison of our method against the aforementioned baselines, focusing on reconstruction metrics as well as the new topology and enclosure metrics we proposed. Notably, our method demonstrates superior performance on reconstruction metrics, even outperforming some reconstruction methods and trailing only behind NVDNet [21]. This gap can be attributed to its Voronoi cells splitting and primitive fitting design. However, in terms of topology and enclosure metrics, our method significantly outperforms the baselines. From the visual comparison in Fig.5, we can also see that our method generates high-fidelity CAD models. While the baselines exhibit numerous dangling edges in their reconstructions, as indicated by the blue lines in the figure, most results of our approach exhibit strict manifold structures without any dangling edges, showcasing superior topological quality. The good topology of our results benefits from the accuracy of our generated command sequences.

When compared to the generative method DeepCAD [1], our approach shows clear advantages across all evaluated metrics. In the qualitative comparison with DeepCAD [1], as illustrated in Fig. 6, our method effectively generates the correct command sequence conditioned on the corresponding point cloud in the majority of cases. In contrast, DeepCAD struggles in several instances, particularly in the generation quality regarding fine details.

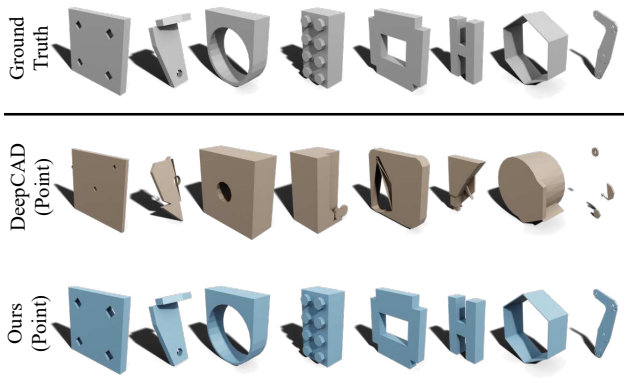


Fig. 6: We qualitatively compare point-based reconstruction results on our dataset with the baseline generative method. Our method successfully generates the correct command sequence with the corresponding point cloud condition.

Methods	Chamfer(×100)↓	F-score(×100)↑	Normal C(×100)↑
InstantMesh [98]	5.38	61.81	45.53
Ours(Image)	**3.77**	**76.70**	**59.62**

TABLE 4: The quantitative results on image-based reconstruction tasks. We observe that our method outperforms reconstruction metrics.

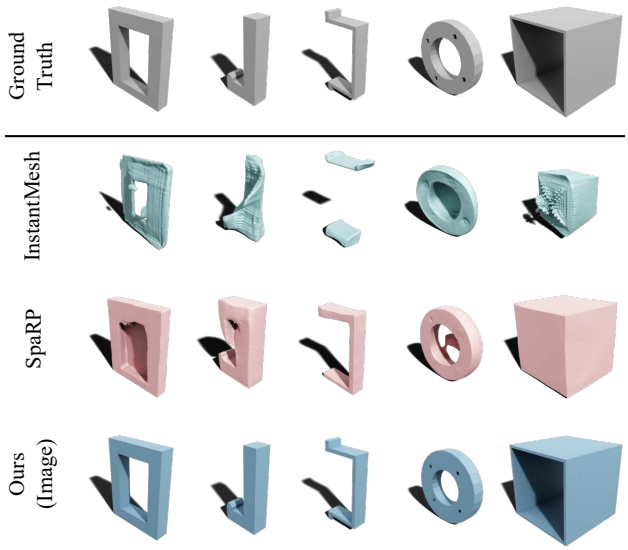


Fig. 7: The qualitative comparison with image-to-mesh baselines. InstantMesh [98] struggles to reconstruct the model's shape accurately. While SpaRP [99] manages to capture the rough shape, it falls short of producing a smooth and axis-aligned CAD model.

6.2.2 *Image Conditioned CAD Generation*

Similar to the training setting of point-based CAD reconstruction, we utilize images exclusively as inputs for training. To ensure a fair comparison, we configure the baseline inference to also process two-view images, aligning with our experimental setup. Given that SpaRP [99] has not been open-sourced and provides only a web demo for inference, our comparisons with it are qualitative only.

We quantitatively compare our method with InstantMesh [98] and show the result in Tab. 4. Note that InstantMesh [98] benefits from an iso-surface extraction module [108], which inherently produces meshes with ex-

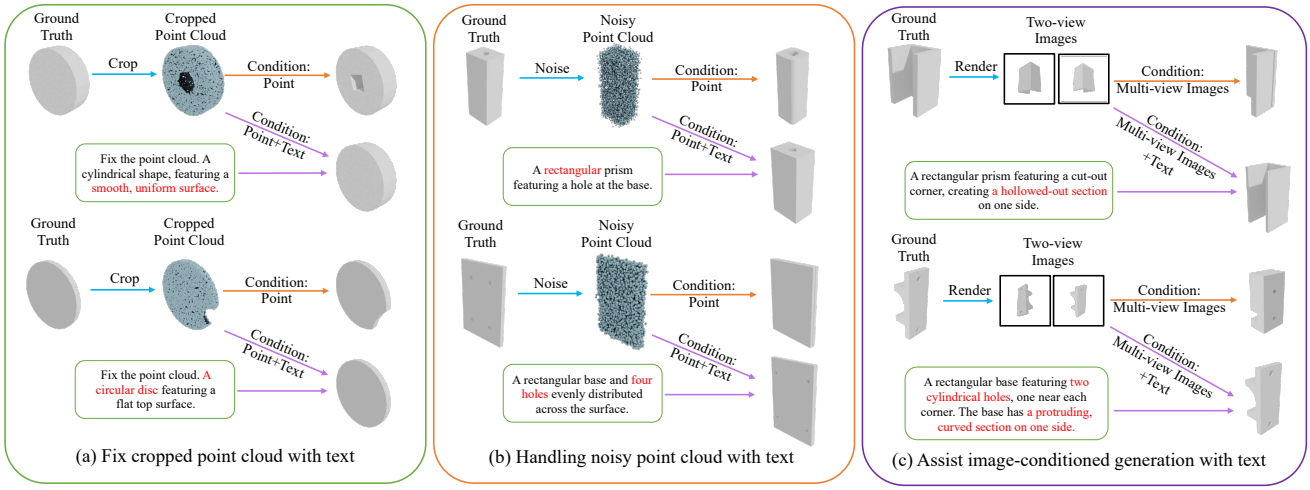


Fig. 8: We present three different applications of our multimodal model. (a) When generating CAD models conditioned solely on the cropped point cloud, our multimodal model will be influenced by the missing spatial information of the cropped data. After additionally prompted with the description of the complete CAD model, our multimodal model is able to fix the cropping. (b) When generating CAD models conditioned solely on the noisy point cloud, our multimodal model will be influenced by the missing spatial details. After additionally prompted with the description of the original CAD model, our multimodal model is able to retain the original feature. (c) When generating CAD models conditioned solely on the two-view images, the complete CAD models are not fully observed. Our multimodal model may fail in some cases. After additionally prompted with the description of the full CAD model, our multimodal model is able to fill the unobserved geometry.

cellent connectivity. This leads to notably low DangEL and SIR values in its extraction process. Furthermore, the use of Signed Distance Functions (SDF) ensures watertight geometries, resulting in exceptionally low FluxEE values. Given these methodological differences and distinct focus areas, our comparison just focuses on the reconstruction metrics. We can observe that our method performs better on reconstruction metrics.

For the qualitative comparison, we evaluate both SpaRP [99] and InstantMesh [98]. From Fig. 7, we observe that InstantMesh struggles with accurately reconstructing the model's shape, while SpaRP captures the general structure but fails to deliver a smooth, axis-aligned result. In contrast, our method successfully reconstructs a smooth and precise CAD representation.

6.2.3 Text Conditioned CAD Generation

Since there are currently no established metrics specifically designed to evaluate the generation of CAD models conditioned on textual descriptions, we compare our method with the open-source method Michelangelo [96] and the closed-source website Tripo [100] by conducting a user study. We invite 16 participants to evaluate 10 pairs of text descriptions and their corresponding generated 3D models. The participants are asked to rate the models based on two criteria: the alignment between the models and the conditioned textual descriptions and the overall quality of the generated models. Each data pair should be scored on a scale of 1 to 5, where 1 represents the lowest quality and 5 represents the highest. Tab. 5 presents the average scores. Our method achieves the highest scores in terms of text alignment and is comparable to that of Tripo in terms of model quality.

Methods	Text Alignment	Model Quality
Michelangelo [96]	1.16	2.04
Tripo [100]	3.30	**4.58**
Ours(Text)	**4.16**	4.45

TABLE 5: The user study results on text-conditioned generation tasks. We evaluate the generated models based on two criteria: text alignment and model quality. A higher score indicates better performance, with both criteria rated on a 5-point scale.

6.2.4 Multimodal-Input-Conditioned CAD Generation

In Fig. 8, we present three scenarios to demonstrate our multimodal model's adaptability in CAD generation across different input conditions.

(a) Cropped Point Cloud: When using only a cropped point cloud, our model will be influenced by the partial lack of spatial information. Supplementing this input with a complete CAD model description enables the model to compensate, reconstructing missing areas effectively.

(b) Noisy Point Cloud: Noisy point clouds reduce detail accuracy. By including a descriptive prompt of the original CAD model, the model produces a more accurate output.

(c) Two-View Images: With two-view images, incomplete viewpoints may lead to missing geometry. Adding a full model description helps the model fill in unobserved sections, achieving a more complete CAD generation.

These cases highlight our model's strength in leveraging multimodal inputs to address challenges from partial or noisy data, enhancing CAD generation fidelity and completeness.

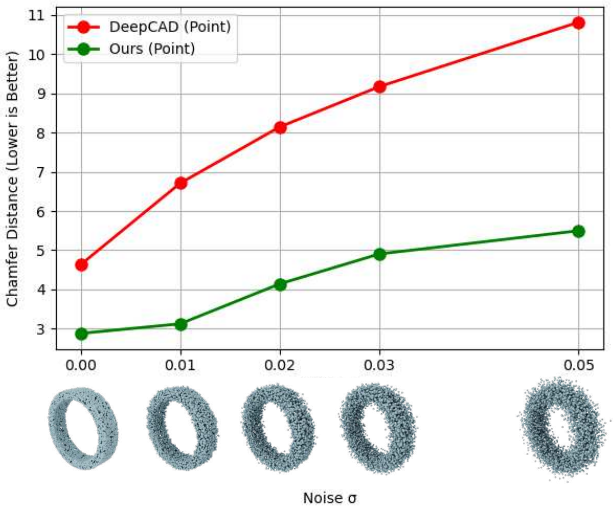


Fig. 9: The Chamfer Distance under varying noise levels. While both baseline [1] and our quality degrade with noises, our approach demonstrates stronger robustness in handling noisy point cloud data.

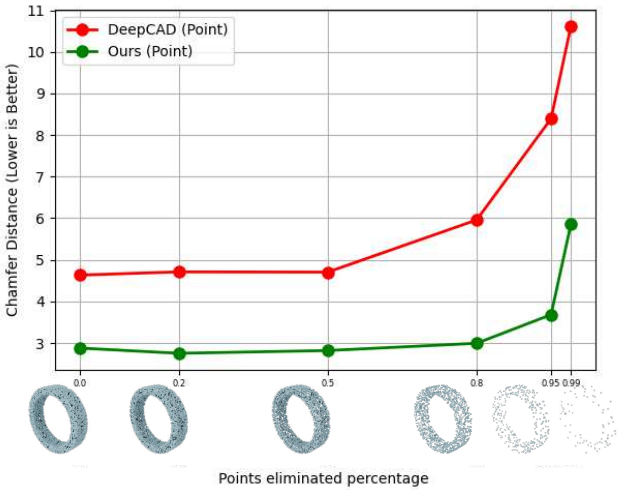


Fig. 11: The Chamfer Distance under different point elimination percentages. Our method demonstrates superior robustness across all point reduction levels. Even if 95% points are removed, our method can still robustly recover the correct CAD models.

6.3 Robustness Evaluation

To further assess the robustness of our approach, we conduct experiments on two challenging tasks: point cloud data with added noise and point cloud data with random point elimination. For each task, we randomly select 1,000 cases from the test set. The noisy point cloud tests examine how well our model can generate CAD models under varying levels of perturbation in the input data, while the partial point cloud tests evaluate the model's ability to reconstruct accurate shapes with less data.

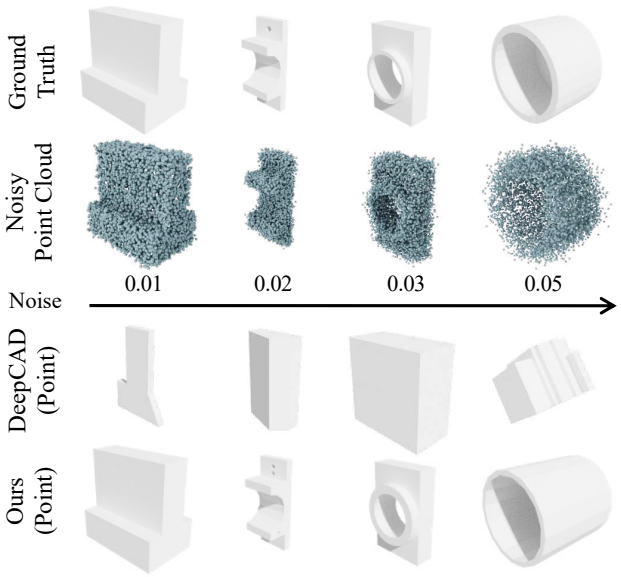


Fig. 10: The qualitative evaluation with different levels of noises added to the point clouds. While our method shows some errors due to the noises, the overall shape structure remains well-preserved. In contrast, DeepCAD [1] is more significantly affected by the noises.

6.3.1 *Performance on Noisy Point Cloud Inputs*

To simulate noisy data, we sample noises from a normal distribution, with zero mean and standard deviation of σ, as an offset of the points in the three positional dimensions.

We use Chamfer Distance as the primary metric for evaluating reconstruction performance and present the results in Fig.9, comparing our model with DeepCAD [1] under varying noise levels. As observed, although both methods experience a decline in performance as noise levels increase, our model demonstrates a slower degradation, indicating stronger robustness when handling noisy point cloud data. A similar trend is visible in Fig.10, where we compare the qualitative robustness of our approach against DeepCAD. While noise introduces some errors in the fine details of our model's reconstruction, the overall structural information remains largely consistent with the ground truth. In contrast, DeepCAD is more significantly affected by the presence of noise, resulting in poorer reconstruction performance. Comprehensive quantitative results for all metrics across all tested noise levels are provided in the supplementary material.

6.3.2 *Performance on Partial Point Cloud Inputs*

To evaluate the model's performance when only partial point cloud data is provided, we progressively eliminate different percentages of points randomly from the original point cloud.

In Fig. 11, we compare our method with DeepCAD [1] under various levels of point cloud reduction in terms of chamfer distance. Our method consistently outperforms DeepCAD across all reduction levels. Surprisingly, even when 95% of the point cloud data is removed, our model still maintains strong reconstruction performance, outperforming DeepCAD's results on complete point clouds.

In Fig. 12, we visualize the qualitative robustness evaluation with varying percentages of points eliminated from the original point clouds. As expected, the reduction in points

Methods	Chamfer(×100)↓	F-score(×100)↑	Normal C(×100)↑	SegE↓	DangEL↓	SIR(%)↓	FluxEE(×100)↓
DeepCAD [1](Point)	7.61	52.77	51.96	13.77	2.13	9.12	0.276
Ours(Point)	**3.39**	**79.27**	**66.78**	**2.05**	**0.63**	**1.68**	**0.194**

TABLE 6: Quantitative generalization test on Fusion360 reconstruction dataset [11]. Our method outperforms DeepCAD [1] across all reconstruction, topology, and enclosure metrics on the unseen data. Neither our method nor DeepCAD is trained using Fusion360 data.

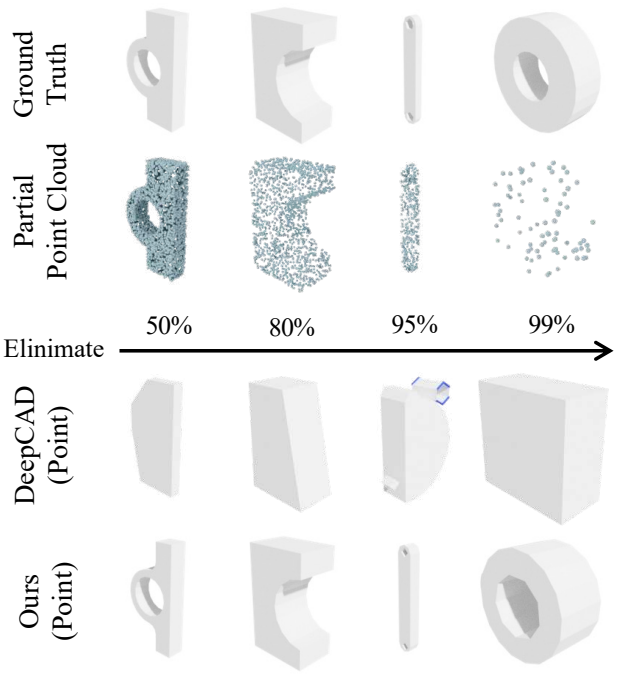


Fig. 12: The qualitative evaluation with different percentages of points eliminated. Our method reconstructs the overall shape well over different eliminated percentages, while DeepCAD [1] is more adversely affected by the elimination.

impacts the expressiveness of the shape, but remarkably, even under extreme conditions where 99% of the points are removed, our model still accurately reconstructs the overall layout and structure of the CAD model. Although there are slight deviations in fine details and dimensions, the core geometry is preserved. In contrast, DeepCAD [1] is significantly more affected by the reduction, leading to much poorer reconstruction results. This highlights the robustness of our method in handling sparse point clouds. Comprehensive quantitative results for all metrics across all tested elimination percentages are provided in the supplementary material.

6.4 Generalization Assessment

To validate the generalization performance of our method on unseen data, we conduct tests using the Fusion360 reconstruction dataset [11]. We randomly sample 1,512 models from the dataset to create a test set, utilizing point cloud data as input for evaluation. The results of our tests are presented quantitatively and qualitatively in Tab. 6 and Fig 13, respectively. Our method consistently outperforms the baseline across all evaluation metrics, demonstrating the effective generalization capabilities on unseen data.

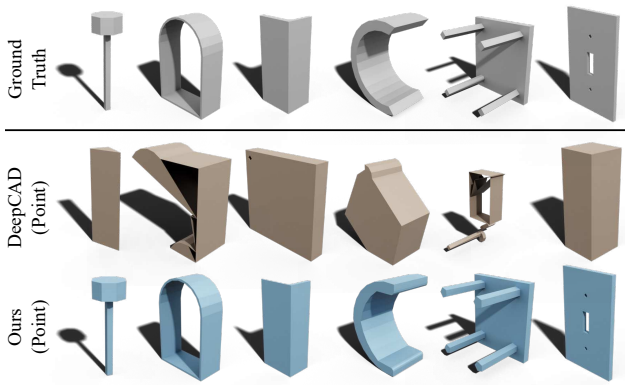


Fig. 13: We present qualitative comparisons for the generalization test on Fusion360 [11], demonstrating that our method achieves better reconstruction of models than DeepCAD [1]. Notably, neither our method nor DeepCAD is trained using Fusion360 data.

6.5 Influence of Multimodal Data Training

Additionally, we investigate the impact of training with multimodal data compared to using single-modal training data on the generated CAD models. As shown in Tab. 7, we compare the performance of models trained on multimodal datasets against those trained exclusively on image data and those trained solely on point cloud data. To ensure fairness in our testing, we utilize corresponding single-modal data for the evaluation phase.

We can observe from Tab. 7(a) that when testing on datasets where only images are used as input conditions, models trained on multimodal data outperform those trained solely on image data. We hypothesize that this improvement is due to the additional complementary information provided by other modalities, such as point clouds, which offer more detailed insights into the structure and geometry of the CAD models. This enriched information helps the model to generate higher-quality CAD models.

Tab. 7(b) shows that when conditioned solely on point data, our multimodal data trained model achieves comparable performance with the model trained exclusively on point cloud data from the topology and enclosure perspective. However, the point reconstruction accuracy of the only point data trained model surpasses that of the multimodal data trained model. A possible reason for this observation is that in unimodal training using only point cloud data, the model can fully focus on optimizing the point cloud representation. And the point cloud data possesses

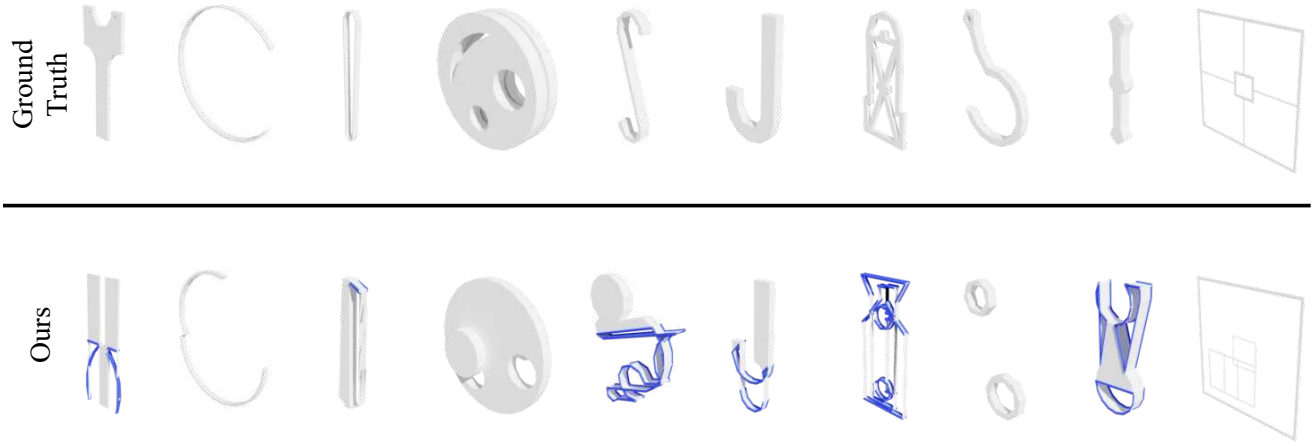


Fig. 14: We present the failure case of our method on some challenging cases, such as thin structures and complicated details. Dangling edges exist in these failure results.

(a) Image Reconstruction Comparison

Methods	Chamfer(×100)↓	F-score(×100)↑	Normal C(×100)↑	SegE↓	DangEL↓	SIR(%)↓	FluxEE(×100)↓
Ours(Image)	3.77	76.70	59.62	1.97	0.79	2.07	0.063
Ours(Multimodal)	**3.22**	**80.82**	**62.07**	**1.56**	**0.51**	**1.36**	**0.050**

(b) Point Reconstruction Comparison

Methods	Chamfer(×100)↓	F-score(×100)↑	Normal C(×100)↑	SegE↓	DangEL↓	SIR(%)↓	FluxEE(×100)↓
Ours(Point)	**1.85**	**90.88**	**79.71**	1.66	**0.46**	**1.31**	0.044
Our(Multimodal)	2.63	85.17	73.64	**1.53**	0.47	1.32	**0.035**

TABLE 7: The quantitative study on training with multimodal data. (a) Our multimodal model outperforms the image-only model across all reconstruction, topology, and enclosure metrics. (b) Our multimodal model is comparable with the point-only model in terms of topology and enclosure metrics. However, the point reconstruction performance of our multimodal model is slightly weaker than the point-only model.

sufficient CAD model's detailed information, resulting in higher point reconstruction accuracy. In contrast, models trained on multimodal data must integrate representations from various modalities and balance the optimization across them. The introduction of other modalities, particularly textual descriptions, which are inherently more coarse and less precise, may introduce noise into the training process. This noise can negatively impact the accuracy of point reconstruction.

7 Limitations

Despite the promising and robust performance of CAD-MLLM, several limitations exist. First, while InternVL2-26B is a commendable open-source multimodal large model, it is particularly sensitive to perspective distortions in multi-view images, especially when dealing with complex shapes, which can adversely affect the generation of textual descriptions. Additionally, current text descriptions often fail to accurately capture the precise geometry of complex shapes, primarily due to a lack of specific CAD dimension information. This results in relative descriptions of attributes such as edge lengths or apertures, rather than absolute size measurements. Consequently, the generated CAD models may exhibit similar shapes but differ significantly in size. This limitation may be addressed by leveraging other work, like [109], to extract CAD dimension attributes from the training data, thereby enabling more precise dimension information to be incorporated during model training. Some failure cases are illustrated in Fig.14.

8 Conclusions

In this work, we propose CAD-MLLM, a MLLM-assisted framework designed to generate parametric CAD models based on textual descriptions, multi-view images, point clouds, or any combination of these inputs, thus facilitating ease of use for non-expert users. To tackle this challenging task, we factor it into two sub-problems. First, we explore a vectorized representation of CAD command sequences to enhance LLM understanding, aligning the feature spaces of multi-view images and point clouds within the LLM's framework. Additionally, to address the gaps in existing datasets regarding multimodality information and to empower LLM capabilities, we propose a new dataset, Omni-CAD. We evaluate our method on this dataset, and beyond traditional reconstruction quality metrics, we introduce four novel evaluation criteria that focus on topology quality and surface enclosure extent. Extensive experimental results demonstrate that our approach outperforms the previous generation methods while exhibiting greater robustness.

9 ACKNOWLEDGEMENT

We sincerely thank Rundi Wu and Xiang Xu for clarifying certain questions during our code implementation process.

Additionally, we thank Ruihan Yu and Shangzhe Li for inspiring discussions on field theory in physics, which led to sparking inspiration for part of this paper.

REFERENCES

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

[2] Y. You, M. A. Uy, J. Han, R. Thomas, H. Zhang, S. You, and L. Guibas, "Img2cad: Reverse engineering 3d cad models from images through vlm-assisted conditional factorization," arXiv preprint arXiv:2408.01437, 2024.

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

[4] M. S. Khan, S. Sinha, T. U. Sheikh, D. Stricker, S. A. Ali, and M. Z. Afzal, "Text2cad: Generating sequential cad models from beginner-to-expert level text prompts," 2024. [Online]. Available: https://arxiv.org/abs/2409.17106

[5] A. Badagabettu, S. S. Yarlagadda, and A. B. Farimani, "Query2cad: Generating cad models using natural language queries," 2024. [Online]. Available: https://arxiv.org/abs/2406.00144

[6] M. A. Uy, Y.-Y. Chang, M. Sung, P. Goel, J. G. Lambourne, T. Birdal, and L. J. Guibas, "Point2cyl: Reverse engineering 3d objects from point clouds to extrusion cylinders," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11 850–11 860.

[7] E. Dupont, K. Cherenkova, D. Mallis, G. Gusev, A. Kacem, and D. Aouada, "Transcad: A hierarchical transformer for cad sequence inference from point clouds," 2024. [Online]. Available: https://arxiv.org/abs/2407.12702

[8] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo, "Abc: A big cad model dataset for geometric deep learning," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[9] E. Dupont, K. Cherenkova, A. Kacem, S. A. Ali, I. Arzhannikov, G. Gusev, and D. Aouada, "Cadops-net: Jointly learning cad operation types and steps from boundary-representations," in 2022 International Conference on 3D Vision (3DV). IEEE, 2022, pp. 114–123.

[10] S. Zhou, T. Tang, and B. Zhou, "Cadparser: A learning approach of sequence modeling for b-rep cad." in IJCAI, 2023, pp. 1804–1812.

[11] K. D. D. Willis, Y. Pu, J. Luo, H. Chu, T. Du, J. G. Lambourne, A. Solar-Lezama, and W. Matusik, "Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences," ACM Transactions on Graphics (TOG), vol. 40, no. 4, 2021.

[12] A. Seff, Y. Ovadia, W. Zhou, and R. P. Adams, "SketchGraphs: A large-scale dataset for modeling relational geometry in computer-aided design," in ICML 2020 Workshop on Object-Oriented Learning, 2020.

[13] C. Li, H. Pan, A. Bousseau, and N. J. Mitra, "Free2cad: Parsing freehand drawings into cad commands," ACM Transactions on Graphics (TOG), vol. 41, no. 4, pp. 1–16, 2022.

[14] Z. Yuan, J. Shi, and Y. Huang, "Openecad: An efficient visual language model for editable 3d-cad design," Computers &; Graphics, vol. 124, p. 104048, Nov. 2024. [Online]. Available: http://dx.doi.org/10.1016/j.cag.2024.104048

[15] T. Chen, C. Yu, Y. Hu, J. Li, T. Xu, R. Cao, L. Zhu, Y. Zang, Y. Zhang, Z. Li et al., "Img2cad: Conditioned 3d cad model generation from single image with structured visual geometry," arXiv preprint arXiv:2410.03417, 2024.

[16] Z. Guo, R. Zhang, X. Zhu, Y. Tang, X. Ma, J. Han, K. Chen, P. Gao, X. Li, H. Li, and P.-A. Heng, "Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following," 2023. [Online]. Available: https://arxiv.org/abs/2309.00615

[17] Z. Ge, H. Huang, M. Zhou, J. Li, G. Wang, S. Tang, and Y. Zhuang, "Worldgpt: Empowering llm as multimodal world model," in Proceedings of the ACM International Conference on Multimedia, 2024.

[18] G. Sharma, D. Liu, E. Kalogerakis, S. Maji, S. Chaudhuri, and R. Měch, "Parsenet: A parametric surface fitting network for 3d point clouds," in European Conference on Computer Vision, 2020.

[19] H. Guo, S. Liu, H. Pan, Y. Liu, X. Tong, and B. Guo, "Complexgen: Cad reconstruction by b-rep chain complex generation," ACM Trans. Graph. (SIGGRAPH), vol. 41, no. 4, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530078

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

[21] Y. Liu, J. Chen, S. Pan, D. Cohen-Or, H. Zhang, and H. Huang, "Split-and-fit: Learning b-reps via structure-aware voronoi partitioning," ACM Trans. Graph., vol. 43, no. 4, jul 2024. [Online]. Available: https://doi.org/10.1145/3658155

[22] S. Ansaldi, L. De Floriani, and B. Falcidieno, "Geometric modeling of solid objects by using a face adjacency graph representation," in Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, ser. SIGGRAPH '85. New York, NY, USA: Association for Computing Machinery, 1985, p. 131–139. [Online]. Available: https://doi.org/10.1145/325334.325218

[23] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, Computer graphics: principles and practice (2nd ed.). USA: Addison-Wesley Longman Publishing Co., Inc., 1990.

[24] W. Cao, T. Robinson, Y. Hua, F. Boussage, A. R. Colligan, and W. Pan, "Graph representation of 3d cad models for machining feature recognition with deep learning," Proceedings of the ASME 2020, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2020.

[25] P. K. Jayaraman, J. G. Lambourne, N. Desai, K. D. D. Willis, A. Sanghi, and N. J. W. Morris, "Solidgen: An autoregressive model for direct b-rep synthesis," in International Conference on Learning Representations, 2024.

[26] J. G. Lambourne, K. D. Willis, P. K. Jayaraman, A. Sanghi, P. Meltzer, and H. Shayani, "Brepnet: A topological message passing system for solid models," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 12 773–12 782.

[27] B. T. Jones, M. Hu, M. Kodnongbua, V. G. Kim, and A. Schulz, "Self-supervised representation learning for cad," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 21 327–21 336.

[28] B. Jones, D. Hildreth, D. Chen, I. Baran, V. G. Kim, and A. Schulz, "Automate: a dataset and learning approach for automatic mating of cad assemblies," ACM Trans. Graph., vol. 40, no. 6, dec 2021. [Online]. Available: https://doi.org/10.1145/3478513.3480562

[29] S. Bian, D. Grandi, T. Liu, P. K. Jayaraman, K. Willis, E. Sadler, B. Borijin, T. Lu, R. Otis, N. Ho, and B. Li, "HG-CAD: Hierarchical Graph Learning for Material Prediction and Recommendation in Computer-Aided Design," Journal of Computing and Information Science in Engineering, vol. 24, no. 1, p. 011007, 10 2023. [Online]. Available: https://doi.org/10.1115/1.4063226

[30] D. Smirnov, M. Bessmeltsev, and J. Solomon, "Learning manifold patch-based representations of man-made shapes," in International Conference on Learning Representations (ICLR), 2021.

[31] K. Wang, J. Zheng, and Z. Zhou, "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, 2022, pp. 1622–1631.

[32] X. Wang, Y. Xu, K. Xu, A. Tagliasacchi, B. Zhou, A. Mahdavi-Amiri, and H. Zhang, "Pie-net: Parametric inference of point cloud edges," Advances in neural information processing systems, vol. 33, pp. 20 167–20 178, 2020.

[33] L. Li, M. Sung, A. Dubrovina, L. Yi, and L. J. Guibas, "Supervised fitting of geometric primitives to 3d point clouds," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2652–2660.

[34] C. Nash, Y. Ganin, S. A. Eslami, and P. Battaglia, "Polygen: An autoregressive generative model of 3d meshes," in International conference on machine learning. PMLR, 2020, pp. 7220–7229.

[35] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in Advances in Neural Information Processing Systems, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[37] X. Xu, J. Lambourne, P. Jayaraman, Z. Wang, K. Willis, and Y. Furukawa, "Brepgen: A b-rep generative diffusion model with structured latent geometry," ACM Trans. Graph., vol. 43, no. 4, jul 2024. [Online]. Available: https://doi.org/10.1145/3658129

[38] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Advances in neural information processing systems, vol. 33, pp. 6840–6851, 2020.

[39] K. Kania, M. Zieba, and T. Kajdanowicz, "Ucsg-net-unsupervised discovering of constructive solid geometry tree," Advances in neural information processing systems, vol. 33, pp. 8776–8786, 2020.

[40] D. Ren, J. Zheng, J. Cai, J. Li, H. Jiang, Z. Cai, J. Zhang, L. Pan, M. Zhang, H. Zhao et al., "Csg-stump: A learning friendly csg-like representation for interpretable shape parsing," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 12 478–12 487.

[41] F. Yu, Z. Chen, M. Li, A. Sanghi, H. Shayani, A. Mahdavi-Amiri, and H. Zhang, "Capri-net: Learning compact cad shapes with adaptive primitive assembly," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 11 768–11 778.

[42] F. Yu, Q. Chen, M. Tanveer, A. M. Amiri, and H. Zhang, "D^2csg: Unsupervised learning of compact csg trees with dual complements and dropouts," 2023. [Online]. Available: https://arxiv.org/abs/2301.11497

[43] D. Ritchie, P. Guerrero, R. K. Jones, N. J. Mitra, A. Schulz, K. D. D. Willis, and J. Wu, "Neurosymbolic models for computer graphics," Computer Graphics Forum, vol. 42, no. 2, pp. 545–568, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14775

[44] Y. Tian, A. Luo, X. Sun, K. Ellis, W. T. Freeman, J. B. Tenenbaum, and J. Wu, "Learning to infer and execute 3d shape programs," in International Conference on Learning Representations, 2019.

[45] K. Ellis, M. Nye, Y. Pu, F. Sosa, J. Tenenbaum, and A. Solar-Lezama, "Write, execute, assess: Program synthesis with a repl," in Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[46] G. Sharma, R. Goyal, D. Liu, E. Kalogerakis, and S. Maji, "Csgnet: Neural shape parser for constructive solid geometry," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5515–5523.

[47] T. Du, J. P. Inala, Y. Pu, A. Spielberg, A. Schulz, D. Rus, A. Solar-Lezama, and W. Matusik, "Inversecsg: automatic conversion of 3d models to csg trees," ACM Trans. Graph., vol. 37, no. 6, dec 2018. [Online]. Available: https://doi.org/10.1145/3272127.3275006

[48] C. Nandi, A. Caspi, D. Grossman, and Z. Tatlock, "Programming language tools and techniques for 3d printing," in 2nd Summit on Advances in Programming Languages (SNAPL 2017). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[49] C. Nandi, J. R. Wilcox, P. Panchekha, T. Blau, D. Grossman, and Z. Tatlock, "Functional programming for compiling and decompiling computer-aided design," Proceedings of the ACM on Programming Languages, vol. 2, no. ICFP, pp. 1–31, 2018.

[50] J. F. Gonzalez, D. Kieken, T. Pietrzak, A. Girouard, and G. Casiez, "Introducing bidirectional programming in constructive solid geometry-based cad," in Proceedings of the 2023 ACM Symposium on Spatial User Interaction, ser. SUI '23. ACM, Oct. 2023, p. 1–12. [Online]. Available: http://dx.doi.org/10.1145/3607822.3614521

[51] J. D. Camba, M. Contero, and P. Company, "Parametric cad modeling," Comput. Aided Des., vol. 74, no. C, p. 18–31, may 2016. [Online]. Available: https://doi.org/10.1016/j.cad.2016.01.003

[52] X. Xu, K. D. Willis, J. G. Lambourne, C.-Y. Cheng, P. K. Jayaraman, and Y. Furukawa, "Skexgen: Autoregressive generation of cad construction sequences with disentangled codebooks," in International Conference on Machine Learning. PMLR, 2022, pp. 24 698–24 724.

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

[54] M. S. Khan, E. Dupont, S. A. Ali, K. Cherenkova, A. Kacem, and D. Aouada, "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 (CVPR), June 2024, pp. 4713–4722.

[55] C. Li, H. Pan, A. Bousseau, and N. J. Mitra, "Sketch2cad: sequential cad modeling by sketching in context," ACM Trans. Graph., vol. 39, no. 6, nov 2020. [Online]. Available: https://doi.org/10.1145/3414685.3417807

[56] A. Seff, W. Zhou, N. Richardson, and R. P. Adams, "Vitruvion: A generative model of parametric cad sketches," in International Conference on Learning Representations, 2022.

[57] Y. Ganin, S. Bartunov, Y. Li, E. Keller, and S. Saliceti, "Computer-aided design as language," Advances in Neural Information Processing Systems, vol. 34, pp. 5885–5897, 2021.

[58] J. G. Lambourne, K. Willis, P. K. Jayaraman, L. Zhang, A. Sanghi, and K. R. Malekshan, "Reconstructing editable prismatic cad from rounded voxel models," in SIGGRAPH Asia 2022 Conference Papers, ser. SA '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3550469.3555424

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

[60] D. Ren, J. Zheng, J. Cai, J. Li, and J. Zhang, "Extrudenet: Unsupervised inverse sketch-and-extrude for shape parsing," in European Conference on Computer Vision. Springer, 2022, pp. 482–498.

[61] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," in International conference on machine learning. PMLR, 2021, pp. 8748–8763.

[62] OpenAI, "Chatgpt," https://openai.com/blog/chatgpt/, 2023.

[63] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat et al., "Gpt-4 technical report," arXiv preprint arXiv:2303.08774, 2023.

[64] AI@Meta, "Llama 3 model card," 2024. [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[65] X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu et al., "Deepseek llm: Scaling open-source language models with longtermism," arXiv preprint arXiv:2401.02954, 2024.

[66] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang et al., "Qwen technical report," arXiv preprint arXiv:2309.16609, 2023.

[67] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," arXiv preprint arXiv:2304.10592, 2023.

[68] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua, "Next-gpt: Any-to-any multimodal llm," arXiv preprint arXiv:2309.05519, 2023.

[69] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," arXiv preprint arXiv:2304.08485, 2023.

[70] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," arXiv preprint arXiv:2301.12597, 2023.

[71] W. Wang, J. Xie, C. Hu, H. Zou, J. Fan, W. Tong, Y. Wen, S. Wu, H. Deng, Z. Li et al., "Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving," arXiv preprint arXiv:2312.09245, 2023.

[72] H. Shao, Y. Hu, L. Wang, G. Song, S. L. Waslander, Y. Liu, and H. Li, "Lmdrive: Closed-loop end-to-end driving with large language models," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 15 120–15 130.

[73] C. Wang, W. Luo, Q. Chen, H. Mai, J. Guo, S. Dong, X. Xuan, Z. Li, L. Ma, and S. Gao, "Mllm-tool: A multimodal large language

model for tool agent learning," arXiv preprint arXiv:2401.10727, vol. 4, 2024.
[74] K. Alrashedy, P. Tambwekar, Z. Zaidi, M. Langwasser, W. Xu, and M. Gombolay, "Generating cad code with vision-language models for 3d designs," 2024. [Online]. Available: https://arxiv.org/abs/2410.05340
[75] S. Wu, A. Khasahmadi, M. Katz, P. K. Jayaraman, Y. Pu, K. Willis, and B. Liu, "Cadvlm: Bridging language and vision in the generation of parametric cad sketches," 2024. [Online]. Available: https://arxiv.org/abs/2409.17457
[76] Capgemini, "Open cascade technology." [Online]. Available: https://www.opencascade.com/open-cascade-technology/
[77] Autodesk, "Fusion 360: 3d cad, cam, cae & pcb cloud-based software," Aug 2021. [Online]. Available: https://www.autodesk.co.uk/products/fusion-360/overview
[78] D. Systemes, "Solidworks. 2019," Dessault Systemes: Vélizy-Villacoublay, France, vol. 434, 2011.
[79] T. Paviot, "pythonocc," 2022. [Online]. Available: https://doi.org/10.5281/zenodo.3605364
[80] "Onshape," http://onshape.com, accessed: 2024-07-19.
[81] "Onshape developer documentation," https://onshape-public.github.io/docs/, accessed: 2024-07-19.
[82] "Onshape featurescript," https://cad.onshape.com/FsDoc/, accessed: 2024-07-19.
[83] Z. Chen, W. Wang, H. Tian, S. Ye, Z. Gao, E. Cui, W. Tong, K. Hu, J. Luo, Z. Ma et al., "How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites," arXiv preprint arXiv:2404.16821, 2024.
[84] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu et al., "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 24 185–24 198.
[85] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer et al., "Perceiver io: A general architecture for structured inputs & outputs," arXiv preprint arXiv:2107.14795, 2021.
[86] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," https://lmsys.org/blog/2023-03-30-vicuna/, 2023.
[87] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar et al., "Llama: Open and efficient foundation language models," arXiv preprint arXiv:2302.13971, 2023.
[88] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv preprint arXiv:2307.09288, 2023.
[89] R. Sennrich, "Neural machine translation of rare words with subword units," arXiv preprint arXiv:1508.07909, 2015.
[90] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," arXiv preprint arXiv:2106.09685, 2021.
[91] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in Proceedings of the 26th annual international conference on machine learning, 2009, pp. 41–48.
[92] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., "Pytorch: An imperative style, high-performance deep learning library," Advances in neural information processing systems, vol. 32, 2019.
[93] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in International Conference on Learning Representations, 2019.
[94] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.
[95] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, "Vision transformers need registers," 2023.
[96] Z. Zhao, W. Liu, X. Chen, X. Zeng, R. Wang, P. Cheng, B. FU, T. Chen, G. YU, and S. Gao, "Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation," in Thirty-seventh Conference on Neural Information Processing Systems, 2023. [Online]. Available: https://openreview.net/forum?id=xmxgMij3LY
[97] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," Advances in neural information processing systems, vol. 30, 2017.
[98] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan, "Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models," arXiv preprint arXiv:2404.07191, 2024.
[99] C. Xu, A. Li, L. Chen, Y. Liu, R. Shi, H. Su, and M. Liu, "Sparp: Fast 3d object reconstruction and pose estimation from sparse views," 18th European Conference on Computer Vision (ECCV), Milano, Italy., 2024.
[100] Tripo3D, "Tripo3d," 2024. [Online]. Available: https://www.tripo3d.ai/
[101] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction," Advances in Neural Information Processing Systems (NeurIPS), 2022.
[102] Y. Xiao, J. Xu, Z. Yu, and S. Gao, "Debsdf: Delving into the details and bias of neural indoor scene reconstruction," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2024.
[103] N. Sharp, K. Crane et al., "Geometrycentral: A modern c++ library of data structures and algorithms for geometry processing," 2019.
[104] The CGAL Project, CGAL User and Reference Manual, 5.6.1 ed. CGAL Editorial Board, 2024. [Online]. Available: https://doc.cgal.org/5.6.1/Manual/packages.html
[105] D. Muller and F. Preparata, "Finding the intersection of two convex polyhedra," Theoretical Computer Science, vol. 7, no. 2, pp. 217–236, 1978.
[106] K. Crane, "Discrete differential geometry: An applied introduction," 2023.
[107] C. F. Gauss, Theoria attractionis corporum sphaeroidicorum ellipticorum homogeneorum, methodo nova tractata. Berlin, Heidelberg: Springer Berlin Heidelberg, 1877, pp. 279–286. [Online]. Available: https://doi.org/10.1007/978-3-642-49319-5_8
[108] T. Shen, J. Munkberg, J. Hasselgren, K. Yin, Z. Wang, W. Chen, Z. Gojcic, S. Fidler, N. Sharp, and J. Gao, "Flexible isosurface extraction for gradient-based mesh optimization," ACM Trans. Graph., vol. 42, no. 4, jul 2023. [Online]. Available: https://doi.org/10.1145/3592430
[109] M. T. Khan, W. Feng, L. Chen, Y. H. Ng, N. Y. J. Tan, and S. K. Moon, "Automatic feature recognition and dimensional attributes extraction from cad models for hybrid additive-subtractive manufacturing," 2024. [Online]. Available: https://arxiv.org/abs/2408.06891

Supplementary material of CAD-MLLM: Unifying Multimodality-Conditioned CAD Generation With MLLM

Jingwei Xu*, Chenyu Wang*, Zibo Zhao, Wen Liu, Yi Ma, Shenghua Gao

1 Quantitative Results of Robustness Tests

We provide the complete quantitative evaluation results of both **Noisy Point Cloud Test** and **Partial Point Cloud Test** in Tab. 1 and Tab. 2. Our method outperforms DeepCAD [1] across all metrics at various kinds and levels of data flaws, which indicates the better robustness of our method.

(A) Clean data

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	4.63	71.47	64.47	9.47	1.32	6.35	0.375
Ours(Point)	**2.88**	**83.10**	**72.66**	**2.22**	**0.64**	**2.02**	**0.066**

(B_1) Noisy data with $\sigma_2 = 0.01$

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	6.71	55.97	53.34	9.27	1.38	8.97	0.227
Ours(Point)	**3.12**	**82.05**	**71.11**	**2.21**	**0.70**	**1.85**	**0.025**

(B_2) Noisy data with $\sigma_2 = 0.02$

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	8.15	46.67	49.64	16.99	1.94	7.63	0.511
Ours(Point)	**4.14**	**74.39**	**65.66**	**2.31**	**0.51**	**1.82**	**0.049**

(B_3) Noisy data with $\sigma_2 = 0.03$

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	9.17	40.84	45.83	16.75	2.01	10.10	0.363
Ours(Point)	**4.91**	**68.99**	**61.51**	**3.96**	**0.81**	**2.86**	**0.283**

(B_4) Noisy data with $\sigma_2 = 0.05$

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	10.82	32.69	44.02	14.70	2.44	13.54	1.230
Ours(Point)	**5.50**	**63.76**	**57.03**	**3.88**	**0.99**	**3.51**	**0.199**

TABLE 1: The quantitative experiment of the robustness tests with noisy data. We observe that over different noise levels, our method demonstrates greater robustness than DeepCAD [1] across all metrics.

- *Jingwei Xu and Chenyu Wang contributed equally to this work;*
- *Corresponding Author: Shenghua Gao; E-mail: gaosh@hku.hk*
- *Jingwei Xu and Zibo Zhao are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China. Email: xujw2023@shanghaitech.edu.cn, zhaozb@shanghaitech.edu.cn*
- *Chenyu Wang is with Transcengram. Email: wangchy@transcengram.com*
- *Wen Liu is with DeepSeek AI. Email: liuwen@deepseek.com*
- *Yi Ma, and Shenghua Gao are with the University of Hong Kong, Hong Kong SAR, China. E-mail: mayi@hku.hk, gaosh@hku.hk*

2 Multimodal Conditioned Input Dataset Visualization

We illustrate 5 pairs of data samples in Fig. 1. As mentioned in the main text, we construct corresponding multimodal data for each CAD model, including textual descriptions, images rendered from 8 fixed angles, and point cloud data. Here, we randomly selected images from 4 of these 8 angles for visualization purposes.

(A) Clean data

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	4.63	71.47	64.47	9.47	1.32	6.35	0.375
Ours(Point)	**2.88**	**83.10**	**72.66**	**2.22**	**0.64**	**2.02**	**0.066**

(C_1) Eliminate 20% points

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	4.71	71.47	64.63	7.64	1.34	6.03	0.281
Ours(Point)	**2.75**	**84.79**	**73.44**	**2.17**	**0.36**	**1.89**	**0.138**

(C_2) Eliminate 50% points

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	4.70	71.40	64.19	8.88	1.41	5.33	0.138
Ours(Point)	**2.82**	**83.37**	**72.69**	**2.14**	**0.45**	**1.67**	**0.025**

(C_3) Eliminate 80% points

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	5.96	62.32	58.40	12.51	1.44	7.54	0.462
Ours(Point)	**2.99**	**82.82**	**71.90**	**2.43**	**0.66**	**1.74**	**0.086**

(C_4) Eliminate 95% points

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	8.39	44.86	47.70	18.28	1.73	7.75	0.560
Ours(Point)	**3.68**	**76.73**	**65.43**	**2.44**	**0.71**	**1.92**	**0.040**

(C_5) Eliminate 99% points

Methods	Chamfer (×100)↓	F-score (×100)↑	Normal C (×100)↑	SegE↓	DangEL↓	SIR (%)↓	FluxEE (×100)↓
DeepCAD [1](Point)	10.62	34.02	44.14	7.71	1.32	7.54	0.323
Ours(Point)	**5.86**	**60.08**	**54.07**	**2.83**	**0.26**	**1.60**	**0.005**

TABLE 2: The quantitative experiment of the robustness tests with noisy data. We observe that over different percentages of eliminated point clouds, our method demonstrates greater robustness than DeepCAD [1] across all metrics.

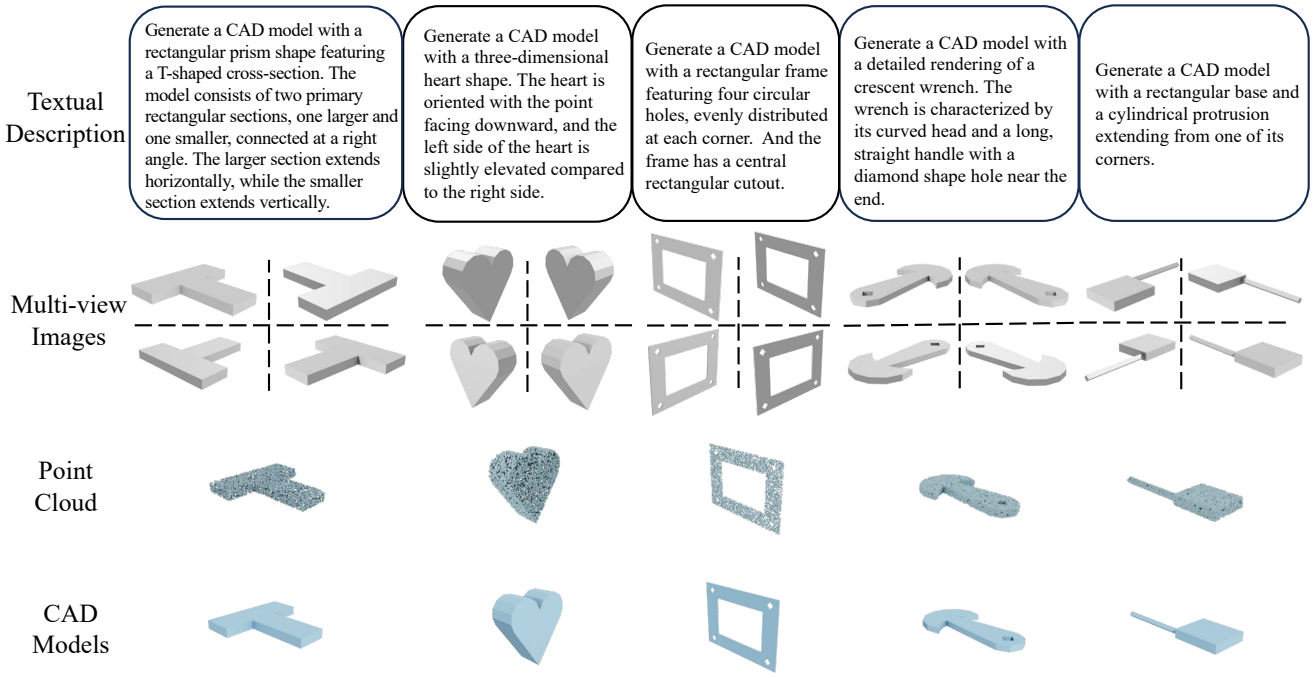


Fig. 1: **Dataset sample visualization.** We sample five cases from our proposed Omni-CAD dataset to illustrate the multimodal conditioned data and the corresponding ground truth CAD models. In the real dataset, each CAD model includes images of eight views; here, we randomly select four views for demonstration purposes.