BrepLLM: Enabling Large Language Models to Understand Boundary Representations

Liyuan Deng[1,2†], Hao Guo[1†], Yongkang Dai[1], Yunpeng Bai[3], Yifan Zhu[1], Yuanyuan Gao[4], Huaxi Huang[2*], and Yilei Shi[1*]

[1] Northwestern Polytechnical University
{dly,yilei_shi}@mail.nwpu.edu.cn
[2] Shanghai Artificial Intelligence Laboratory
huanghuaxi@pjlab.org.cn
[3] National University of Singapore
[4] Hong Kong University of Science and Technology

Abstract. Current token-sequence-based Large Language Models (LLMs) struggle to directly process 3D Boundary Representation (B-rep) models that contain complex geometric and topological information. To this end, we propose BrepLLM, the first multimodal framework that enables LLMs to directly parse and reason over raw B-rep data. BrepLLM adopts a two-stage training pipeline: cross-modal alignment pre-training and two-stage LLM fine-tuning. In the first stage, we design an adaptive UV sampling strategy to convert B-reps into graph representations that integrate geometric and topological information. Subsequently, we construct a hierarchical BrepEncoder to extract features from geometric elements (faces and edges) and topology, generating a global token and a sequence of node tokens. Then, via contrastive learning, we conduct an initial alignment between this global token and the text embeddings of a frozen CLIP text encoder (ViT-L/14). In the second stage, we integrate the pre-trained BrepEncoder into the LLM and employ a two-stage progressive strategy to align the sequence of node tokens: (1) training an MLP-based semantic mapping network that utilizes the prior knowledge of a 2D-VLM to align the B-rep representation to the 2D visual semantic space; (2) utilizing LoRA for parameter-efficient fine-tuning of the Q-Former and the LLM backbone network to achieve the final 3D-language generation capability. Furthermore, we construct the Brep2Text dataset, which contains 269,444 B-rep and text question-answer pairs. Experiments demonstrate that BrepLLM achieves SOTA performance on 3D object classification and captioning tasks.The project page is available at https://user-deng.github.io/BrepLLM/.

Keywords: B-rep · CAD · 3D Understanding

1 Introduction

Large language models (LLMs) have recently achieved remarkable progress in natural language processing [1,15,28], computer vision [3,23,25],and 3D under-

[†]Equal contribution. [*]Corresponding author.

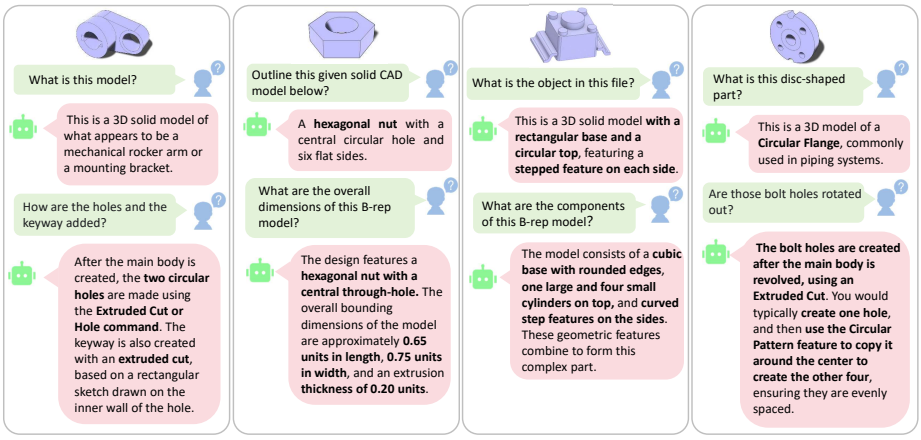


Fig. 1: We introduce BrepLLM, a multi-modal large language model capable of understanding B-rep models. By natively perceiving the topology and geometry of B-reps, it enables multi-turn interactions without occlusion or viewpoint dependency.

standing [34,43,51],establishing a unified foundation for multimodal intelligence. In industrial applications, Computer-Aided Design (CAD) plays an indispensable role, widely used in manufacturing, aerospace, and architecture [12]. CAD models are precisely defined by Boundary Representations (B-reps), which serve as the predominant format for expressing complex geometries in freeform surface modeling [46]. Unlike point clouds or triangle meshes, B-reps encode exact parametric surfaces together with a watertight topology and explicit topological adjacency, making them both information-rich and structurally constrained for downstream geometric reasoning and symbolic manipulation. Enabling LLMs to directly parse and reason over native B-rep models would drive significant advances in industrial model design and intelligent manufacturing.

However, existing large model frameworks still cannot directly process the native geometric and topological structures of CAD models. Instead of directly modeling B-reps, they typically convert CAD into executable intermediate representations—such as command sequences, sketch operation sequences, or program code—that are later executed by a CAD kernel. For example, CAD-MLLM [42] generates CAD command sequences to synthesize parametric models from text, images, or point clouds. CadVLM and CAD-LLM [38, 39] leverage pretrained models to generate sketch sequences for tasks such as sketch auto-completion and constraint reasoning, while cadrille and CAD-Coder [5, 18] use LLMs to generate CAD programs for indirect 3D construction. Although these methods have shown progress for generation and editing, their operation space is largely procedural rather than the underlying B-rep entities (faces/edges) and topology, which limits genuine geometric or topological reasoning. Moreover, construction sequence data is extremely scarce, whereas native B-rep models are an industry-standard format that is abundant and easier to acquire at scale.

To bridge this gap, we propose **BrepLLM**, the first large language model framework for directly understanding B-rep (Figure 1). BrepLLM employs a two-

stage training pipeline, consisting of **Cross-modal Alignment Pre-training** and **Two-stage LLM Fine-tuning** (Figure 2). In the cross-modal alignment stage, we introduce an adaptive UV sampling strategy that converts B-reps into graph representation. We design BrepEncoder, a hierarchical architecture that jointly extracts local geometric features of faces, edges, and topological features, which are then globally aggregated to produce a representation. We then employ a CLIP-style contrastive learning strategy [32] to align the global representation produced by BrepEncoder with the frozen CLIP text encoder (ViT-L/14) [31], thereby constructing a unified geometry–language embedding space.

In the LLM fine-tuning stage, we integrate the pretrained BrepEncoder into a large language model and adopt a two-stage progressive alignment strategy. We first leverage the alignment priors of 2D vision–language models (2D-VLMs) to establish an initial correspondence between B-reps and text. We then perform LoRA-based fine-tuning to incorporate B-rep representations into the LLM's semantic space. Furthermore, we construct the **Brep2Text** dataset, comprising 269,444 high-quality Brep–language pairs. This dataset serves as a large-scale benchmark for training and evaluating models on B-rep-centric language understanding and reasoning tasks.

Our key contributions are as follows:

- We propose BrepLLM, the first framework that enables large language models to directly parse and reason over native B-rep models.
- We introduce an adaptive UV sampling strategy and a hierarchical BrepEncoder to model geometry and topology, producing multi-granularity representations over faces, edges, and their connectivity.
- We adopt a progressive alignment strategy that transfers alignment priors from 2D-VLMs and gradually integrates B-rep geometric features into the semantic space of large language models via LoRA-based fine-tuning.
- We establish the first large-scale benchmark for B-rep-centric language understanding tasks, comprising 269,444 high-quality B-rep models paired with natural language question–answer examples.

2 Related Work

2.1 CAD in LLMs

Recent work on integrating LLMs into CAD has predominantly followed a procedural representation paradigm, converting parametric CAD into command sequences or domain-specific languages (DSLs) amenable to generation and editing. In the realm of sequence generation, CAD-LLM [38] and CadVLM [39] tackle sketch auto-completion, while Text2CAD [16], Text to CADQuery [41], and Query2CAD [2] pioneer text-conditioned CAD command sequence synthesis paradigm. Building on these foundations, CAD-LLama [20], CAD-coder [5], LLM4CAD [24], and FlexCAD [50] further enhance code construction capabilities via instruction tuning. Multimodal and interactive applications have likewise flourished: CAD-MLLM [42] and CAD-GPT [37] supported multimodal sequence

generation, whereas CAD-Editor [47], CAD Translator [22], BlenderLLM [6], and CAD-Assistant [27] explore local editing, code migration, and interactive assistance. Nevertheless, these methods primarily operate over procedural sequences rather than directly on the native B-rep entities — faces, edges, and topological structures — and thus fall short of direct geometric perception in 3D space. Furthermore, they depend on complete construction history data, which is difficult to obtain: the DeepCAD dataset [40], which includes build sequences, is only one-fifth the size of the ABC dataset [17], which provides only the final B-rep models. These constraints in data scale and modeling flexibility pose significant bottlenecks when scaling toward larger CAD foundation models.

2.2 Direct B-rep Learning

As the standard data format in industrial CAD, B-rep has attracted growing attention for 3D model classification, retrieval, feature recognition, and generation. Especially in the field of 3D generation, learning methods based directly on native B-rep are gradually becoming a research hotspot because they do not rely on construction history or predefined modeling operations. SolidGen [14] decomposes B-rep generation into joint prediction of vertices, edges, and faces, though it does not support freeform surface and curve types. To overcome these geometric expressiveness limitations, BrepGen [46] introduces a multi-stage diffusion model for B-rep generation in latent space, and NeuroNURBS [7] employs B-spline surface primitives in place of traditional UV grids for more compact geometric encoding. To improve the structural validity of generated outputs, DTGBrepGen [21] proposes disentangled generation of topology and geometry. Additionally, HoLa [26] constructs a holistic variational autoencoder that maps complete B-rep models into a unified latent space, substantially simplifying latent diffusion model training. Most recently, AutoBrep [45] adopts an autoregressive architecture for B-rep sequence generation, achieving notable improvements in both generation quality and computational efficiency. In a related direction, BrepARG [19] represents B-rep models as holistic token sequences that jointly encode geometric and topological information, further demonstrating the potential of sequence modeling for native B-rep generation. Despite these advances in native B-rep representation and generation, existing methods mainly focus on reconstruction or generative modeling, and no existing work has yet realized a unified multimodal large language model capable of natively parsing B-rep data and supporting complex engineering reasoning.

2.3 3D Large Models for Object Understanding

Recent efforts to integrate LLMs with 3D data have made substantial progress on object-level understanding tasks. Early approaches [13] rely on multi-view 2D renderings to leverage existing 2D vision-language models, but such methods are susceptible to spatial occlusion and lose fine-grained 3D geometric detail. The research paradigm has since shifted toward direct processing of 3D native data. Representative works including Point-Bind [11], PointLLM [44],

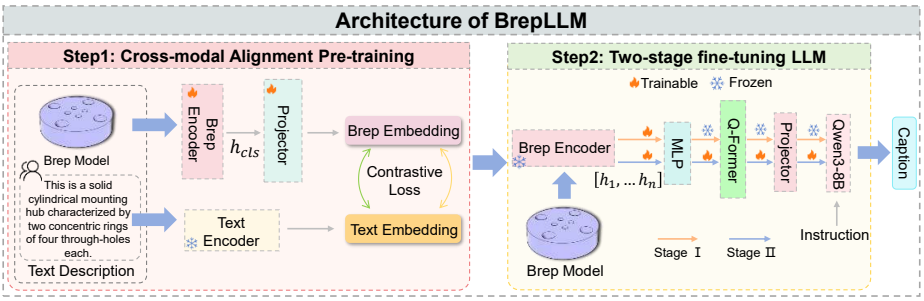


Fig. 2: Overview of the BrepLLM architecture. The framework consists of two steps. Step 1 (Left): Cross-modal Alignment Pre-training. BrepEncoder processes the B-rep model to produce a global feature. This feature is then aligned with text embeddings from a frozen CLIP Text Encoder (ViT-L/14) using a contrastive loss. Step 2 (Right): Two-stage LLM Fine-tuning. The frozen BrepEncoder's node tokens are progressively aligned with LLM. Stage I trains an MLP to map the node tokens; Stage II fine-tunes the Q-Former and LLM using LoRA.

and ShapeLLM [29] introduce pretrained 3D encoders to achieve end-to-end alignment between point cloud features and the representation space of language models, effectively bridging geometric and semantic information. Building on these, MiniGPT-3D [34] incorporates 2D priors and proposes a Mixture-of-Query-Experts (MQE) module with a cascaded training strategy, further improving the efficiency and fidelity of mapping point cloud features to the text space. GreenPLM [49] enables direct transfer of monolingual pretrained language models to other languages via bilingual lexicons. Beyond object-level understanding, recent studies have also extended 3D language-conditioned intelligence toward physically grounded world modeling and embodied action. Mirage2Matter [9] constructs a physically grounded Gaussian world model from video for scalable embodied training data generation, while HUGE-Bench [10] introduces a benchmark for high-level UAV vision-language-action tasks in 3D digital-twin environments. These efforts highlight the growing importance of connecting 3D perception, language, and action. Despite these advances, existing 3D large multimodal models (3D LMMs) still predominantly operate on discrete point clouds, polygonal meshes, 2D renderings, or scene-level Gaussian representations. Such representations inherently struggle to capture the continuous parametric surfaces and strict topological constraints in native CAD B-rep, leaving precise geometric understanding and reasoning in engineering scenarios an open challenge.

3 Method

3.1 Architecture Overview

BrepLLM adopts a two-stage training framework: **Cross-modal Alignment Pre-training** and **Two-stage LLM Fine-tuning**, as illustrated in Figure 2.

Cross-modal Alignment Pre-training aims to extract high-dimensional geometric and topological features from raw CAD data and align them with natural language descriptions, consisting of three steps. **(1) Graph Representation**: B-rep models are converted into graphs by representing face-adjacency topology as edge, and representing parameterized fine-grained geometry as node via adaptive UV sampling; **(2) Feature Encoding**: A hierarchical BrepEncoder processes the graph and extracts both a global token representing the entire B-rep and a sequence of node tokens corresponding to each face; **(3) Cross-modal Alignment**: The global B-rep feature is aligned with text embeddings from a frozen CLIP text encoder (ViT-L/14) via CLIP contrastive loss.

Two-stage LLM Fine-tuning freezes the BrepEncoder and fine-tunes the pretrained language model Qwen3-8B [36] in two stages. **Stage I (Geometry-to-Vision Bridging)**: A lightweight MLP is trained to project B-rep features into the Q-former input space; **Stage II (3D–Language Alignment)**: The Q-former and selected LLM layers are fine-tuned using LoRA.

3.2 Cross-modal Alignment Pre-training

B-rep Parameterization. To ensure compatibility with neural network inputs while preserving geometric and topological information, we propose a multi-scale adaptive UV sampling strategy. See Figure 3(a). This approach constructs a face-edge topological graph structure for joint discretization of faces and edges, where nodes represent parameterized face patches and edges encode adjacency relationships between faces. Face domains employ area-driven grid sampling, while edges adopt length-adaptive point sampling to establish multi-scale geometric representations across local details and global structures.

For face discretization, in the adaptive UV sampling, with given parameter domain $\Omega_{\mathcal{S}} = [u_{\min}, u_{\max}] \times [v_{\min}, v_{\max}]$, where $u_{\min}, u_{\max}$ and $v_{\min}, v_{\max}$ refer to the two directions minimum and maximum dimensions discretization respectively. The density $N_{\mathcal{S}}$ is computed as:

$$N_{\mathcal{S}} = N_{\min}^{\text{face}} + \frac{A_{\mathcal{S}} - A_{\min}}{A_{\max} - A_{\min}} \cdot (N_{\max}^{\text{face}} - N_{\min}^{\text{face}}) \tag{1}$$

where $A_{\min}/A_{\max}$ denote the minimum/maximum face areas in the model. Each sample point $(u_k, v_l) \in \Omega_{\mathcal{S}}$ yields a 10-dimensional feature tensor $\mathbf{X}_{\mathcal{S}} \in \mathbb{R}^{N_{\mathcal{S}} \times 10}$ by extracting 3D coordinates $\mathbf{P} \in \mathbb{R}^3$, unit normals $\mathbf{n} \in \mathbb{S}^2$, mean curvature $H \in \mathbb{R}$, visibility mask $\mathbf{V} \in \{0, 1\}$, face type $\mathbf{t} \in \mathbb{Z}$, and normalized area $a = A_{\mathcal{S}}/A_{\max}$.

For edges, the sampling count $M_{\mathcal{C}}$ is determined by:

$$M_{\mathcal{C}} = M_{\min}^{\text{edge}} + \frac{\ell_{\mathcal{C}} - \ell_{\min}}{\ell_{\max} - \ell_{\min}} \cdot (M_{\max}^{\text{edge}} - M_{\min}^{\text{edge}}) \tag{2}$$

with $\ell_{\min}/\ell_{\max}$ representing edge length range. Each sample point produces an 8-dimensional sequence feature $\mathbf{X}_{\mathcal{C}} \in \mathbb{R}^{M_{\mathcal{C}} \times 8}$ containing 3D coordinates $\mathbf{Q} \in \mathbb{R}^3$, unit tangents $\boldsymbol{\tau} \in \mathbb{S}^2$, edge type $\mathbf{c} \in \mathbb{Z}$, and normalized length $b = \ell_{\mathcal{C}}/\ell_{\max}$.

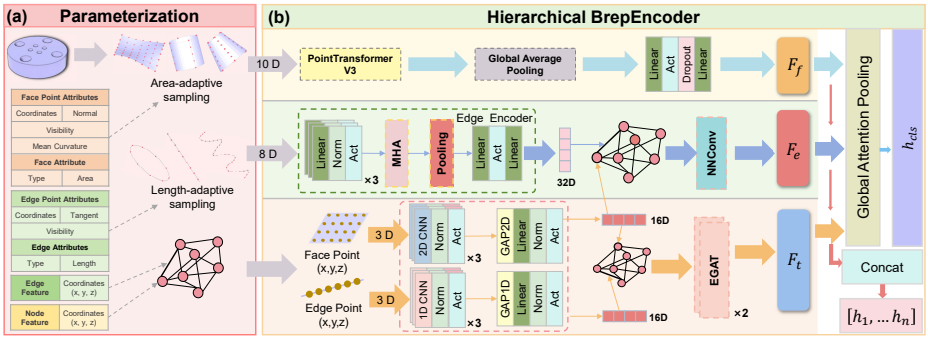


Fig. 3: The overview of BrepEncoder. (a) B-rep parameterization using area-adaptive UV sampling for faces and length-adaptive sampling for edges, producing the face attribute tensor $\mathbf{X}_{\mathcal{S}}$ and edge attribution tensor $\mathbf{X}_{\mathcal{C}}$. (b) Hierarchical BrepEncoder. Face features F_f, edge-conditioned features F_e, and global topology features F_t are extracted from per-node tokens $\mathbf{h}_i$. A global graph feature $\mathbf{h}_{\text{cls}}$ is obtained via global attention pooling.

This framework provides high-quality input for cross-modal alignment pre-training through $C_{\text{face}} = 10$ node features and $C_{\text{edge}} = 8$ edge features, with face sampling count $N_{\mathcal{S}}$ and edge sampling count $M_{\mathcal{C}}$ each constrained to the range $[32, 64]$, balancing geometric precision and topological integrity.

Hierarchical BrepEncoder. We design a novel Hierarchical BrepEncoder to encode geometry and topology of a B-rep model. The encoder produces a global graph feature $\mathbf{h}_{\text{cls}}$ and a sequence of node tokens obtained by concatenating three level-wise features. The global feature is used for contrastive learning, and the node tokens are used to fine-tune the LLM. It explicitly decomposes B-rep features into three disentangled, fine-to-coarse hierarchical representations: 1) the fine-grained face feature F_f, 2) edge-conditioned feature F_e, and 3) global topology feature F_t. See Figure 3(b).

We compute the face feature F_f for geometric details within each face. For this branch, we process the full attribute tensor of each face that contains geometric attributes (e.g., coordinates, normals, mean curvature, etc.). This tensor is fed into a PointTransformerV3 module [40], which applies self-attention over these sampled attribute points to produce a 32-dimensional feature vector $F_f \in \mathbb{R}^{32}$.

Then we compute the edge-conditioned feature F_e to encode local boundary and neighborhood information defined by adjacent edges. We introduce an Edge Encoder that transforms five attributes of each edge (e.g., coordinates, tangents, edge type, normalized length) into a feature representation capturing edge-level geometry. This representation is used as the kernel in an NNConv layer, enabling neighbor-to-center face message passing conditioned on the geometry of the shared boundary, resulting in a 32-dimensional feature $F_e \in \mathbb{R}^{32}$.

Finally, we extract the global topological feature F_t of the B-rep model. We first encode the 3D coordinates of faces and edges, using 2D and 1D CNNs, respectively. The resulting features are fed into two layers of EGATConv. Through

multi-hop attention propagation, information about the global topology is integrated into each face node, resulting in a 64-dimensional $F_t \in \mathbb{R}^{64}$.

These three features are concatenated to face node i:

$$h_i = [F_t^{(i)} \| F_e^{(i)} \| F_f^{(i)}] \in \mathbb{R}^{128} \tag{3}$$

For contrastive learning and LLM fine-tuning, the BrepEncoder outputs a global graph feature $\mathbf{h}_{\text{cls}}$ and a sequence of node tokens $[h_1, h_2, \ldots, h_n]$. The global feature $\mathbf{h}_{\text{cls}}$ serves as a compact representation of the entire B-rep model for contrastive pretraining, while the node tokens h_i encode hierarchical geometric and topological information and are used as inputs for aligning the LLM.

Brep–Text Embedding Alignment. Directly train a large language model with a complex, hierarchical graph representations as input is challenging. To bridge this gap, we first align the B-rep features with natural language descriptions through contrastive pretraining, mapping both modalities into a shared semantic embedding space. Then, we construct the Brep2Text dataset, which contains a large number of CAD models paired with corresponding textual descriptions. Based on this dataset, we adopt a CLIP-style dual-tower architecture for pretraining. The text tower is a frozen CLIP text encoder (ViT-L/14) [32], while the geometry tower comprises our BrepEncoder combined with a projector layer, which is jointly trained to align geometric representations with the text embedding space.

We take the global graph feature $\mathbf{h}_{\text{cls}}$ and feed it into a projector layer. This projects the feature to match the embedding dimension D of the text encoder, resulting in the final B-rep global embedding $\mathbf{z}_{\text{brep}} \in \mathbb{R}^D$. For the text modality, the descriptive caption is fed into the frozen CLIP text encoder, and the output is used as the semantic text representation $\mathbf{z}_{\text{text}} \in \mathbb{R}^D$. We employ a symmetric contrastive loss for training. Given a batch of N matched shape-text pairs, we first L_2-normalize their respective embeddings:

$$\hat{\mathbf{z}}_{\text{brep}} = \mathbf{z}_{\text{brep}} / \|\mathbf{z}_{\text{brep}}\|_2, \quad \hat{\mathbf{z}}_{\text{text}} = \mathbf{z}_{\text{text}} / \|\mathbf{z}_{\text{text}}\|_2 \tag{4}$$

We then compute a pairwise cosine similarity matrix $S \in \mathbb{R}^{N \times N}$, where $S_{ij} = \hat{\mathbf{z}}_{\text{brep},i} \cdot \hat{\mathbf{z}}_{\text{text},j}$. These similarity scores are scaled by a learnable temperature parameter τ within the softmax normalization to obtain the shape-to-text distribution P and the text-to-shape distribution Q:

$$P_{ij} = \frac{\exp(S_{ij}/\tau)}{\sum_{k=1}^{N} \exp(S_{ik}/\tau)} \tag{5}$$

$$Q_{ij} = \frac{\exp(S_{ij}/\tau)}{\sum_{k=1}^{N} \exp(S_{kj}/\tau)} \tag{6}$$

Finally, we adopt the symmetric InfoNCE loss, which maximizes the similarity of the N correctly matched pairs (i.e., the diagonal elements of the probability distributions):

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2N} \sum_{i=1}^{N} (\log P_{ii} + \log Q_{ii}) \tag{7}$$

This loss pulls matched CAD–text pairs closer together while pushing unmatched pairs farther apart in the shared space. During this stage, we optimize the BrepEncoder to produce these language-aligned representations, while the entire CLIP text encoder remains frozen. This process yields a language-aligned geometric embedding that serves as the foundation for the subsequent LLM fine-tuning stages.

3.3 Two-stage LLM Fine-tuning

After aligning geometric and textual representations, we integrate the pretrained BrepEncoder into a large language model [36] to form an end-to-end generation pipeline from geometry to text. The overall architecture comprises the B-rep encoding module, the geometry→Q-Former projection, the pretrained Q-Former, and the LLM backbone. The BrepEncoder outputs a sequence of node tokens, denoted as $[h_1, h_2, \ldots, h_n] \in \mathbb{R}^{N \times D_g}$. To match the pretrained BLIP-2 Q-Former embedding width ($D_{qf} = 1408$), we use a two-layer MLP to project the 128-dimensional geometric tokens to D_{qf}, yielding the Q-Former value input $\mathbf{X}_{\text{qf}} \in \mathbb{R}^{T \times D_{qf}}$.

Stage I: Geometry-to-Vision Bridging. We freeze the BrepEncoder, the Q-Former, and the LLM, and train only the geometry→Q-Former projection head (two-layer MLP). To transform the variable-length geometric tokens into a fixed-size representation for stable training, the pretrained Q-Former employs $Q = 32$ learnable query vectors. These queries cross-attend to the geometric embedding sequence $\mathbf{X}$, distilling the variable-length input into 32 fixed tokens. The resulting outputs are linearly mapped into the LLM hidden space and trained with an autoregressive cross-entropy objective. This establishes an initial semantic bridge from geometric tokens to the pretrained vision–language interface while preserving linguistic priors.

Stage II: 3D–Language Alignment Fine-tuning. Building on Stage I, we apply LoRA-based joint tuning to selected components: key sublayers and necessary normalization layers of the Q-Former, the geometry→Q-Former projection, and a small subset of LLM parameters. The BrepEncoder remains frozen. This stage aims to establish a stable and semantically coherent mapping between structured B-rep representations and the language space. By jointly adapting the cross-modal interface and the language backbone with a small learning rate, the model progressively acquires the capability to interpret Brep-specific geometric and topological patterns and express them in natural language.

4 Brep2Text Dataset

To train and evaluate our model, we introduce **Brep2Text**, the first large-scale instruction-tuning dataset and benchmark specifically designed for the B-rep data modality. Built upon the Text2CAD corpus, Brep2Text leverages two semantic tiers: the abstract level and the beginner level. The abstract level prompts high-level semantic descriptions of CAD objects (e.g., function or category),

while the beginner level focuses on outlining basic modeling steps. This dual-tier design explicitly encourages the model to jointly acquire categorical semantic understanding and procedural modeling logic—two core capabilities essential for real-world CAD interaction.

For each of the 134,722 unique B-rep models, we use Qwen-Max to automatically generate level-specific questions that are semantically aligned with the original human-written descriptions in Text2CAD, which serve as ground-truth answers. This yields a total of 269,444 high-quality question–answer pairs. Furthermore, we strictly reserve 200 CAD models as a held-out test set to ensure reliable evaluation. Brep2Text is not only the first instruction-following dataset tailored to B-rep representations but also establishes a reproducible benchmark for parametric CAD understanding tasks. See Appendix for details.

5 Experiment

5.1 Experimental Setup

Implementation Details. We adopt a two-step training scheme and conduct all experiments on 4 * 80GB A800 GPUs. We select the Qwen3-8B [36] model as the large language model backbone and initialize the Q-Former and projector using the pre-trained weights of TinyGPT-V [48]. In the **first stage**, we utilize the AdamW optimizer with an initial learning rate of 1×10^{-4} and a weight decay of 0.01. A linear warmup is applied for the first 5 epochs, followed by a cosine annealing strategy. The batch size is set to 256, and the model is trained for a total of 200 epochs with mixed-precision training and gradient clipping enabled. In the **second stage**, the weight decay is set to 0.05. The learning rate for the stage-1 is set to 3×10^{-5}, and 5×10^{-6} for the stage-2. Following the settings of previous works [29,35,44], we allocate 200 samples to the test set. For a fair comparison with point cloud-based baseline methods, each B-rep model is converted into a point cloud representation $P \in \mathbb{R}^{n \times d}$, where the number of points n is 8192 and the feature dimension d is 6 (missing color channels are set to black). For the VLM baselines, we render standard three-view images for each B-rep model. Additionally, we employ the "Qwen3-30B-A3B-Instruct-2507" model as the judge model during the evaluation phase. More implementation and training details can be found in the supplementary material.

Baselines. To investigate the performance gap among image-based, point cloud-based, and B-rep-based Multimodal Large Language Models, and to demonstrate the superiority of the B-rep data format in representing CAD geometry, we select several competitive 2D and 3D MLLMs as baselines. Under a consistent data split and evaluation protocol, we retrained all baselines on the Brep2Text dataset and uniformly use Qwen3-30B [36] for objective quantitative evaluation.

5.2 3D Object Captioning

To evaluate the model's fine-grained semantic understanding of 3D CAD objects, we conduct a 3D object captioning task on the Brep2Text dataset.

Table 1: 3D object captioning results on the Brep2Text dataset. Results are reported across human evaluation, LLM evaluation, and traditional metrics. The **bold** and underline denote the best and second best, respectively; green numbers show gains over the second best.

Model	LLM Size	Qwen3-30B	Sentence-BERT	SimCSE	Human Evaluation		
					Correctness	Hallucination↓	Precision
LLaVA-13B [25]	13B	36.73	45.67	47.16	2.16	1.58	63.15
Qwen3-VL-8B [4]	8B	55.48	67.65	68.85	3.97	0.86	78.76
PointLLM-7B [43]	7B	46.81	65.72	66.05	3.32	1.13	74.60
PointLLM-13B [43]	13B	49.65	66.78	67.32	3.46	1.28	72.99
ShapeLLM-7B [30]	7B	48.32	67.14	68.77	3.79	0.96	79.78
ShapeLLM-13B [30]	13B	51.36	68.36	70.12	3.74	1.35	73.47
Minigpt-3D [34]	**2.7B**	56.58	71.64	73.13	4.01	1.04	79.40
BrepLLM	8.4B	**61.36** (+4.78)	**75.42** (+3.78)	**77.23** (+4.10)	**4.63** (+0.62)	**0.83** (+0.03)	**83.62** (+3.84)

Settings. All models are prompted with a unified instruction: "How was this CAD model constructed? Please describe this CAD model in detail." To ensure a comprehensive evaluation, we adopt three complementary assessment protocols: (1) Large Language Model evaluation, where Qwen3-30B judges the semantic consistency between the generated captions and human annotations; (2) Traditional metric evaluation, following the methods of PointLLM [44] and MiniGPT-3D [35], which uses Sentence-BERT [33] and SimCSE [8] to compute embedding-based semantic similarity; (3) Human evaluation, where volunteers perform blind assessments using FreeCAD. Volunteers visually inspect each B-rep model and score the models' responses along two dimensions: *Correctness* (the number of accurately described attributes, such as shape, dimensions, and modeling steps) and *Hallucination* (the count of unsupported or fabricated details). *Precision* is defined as the ratio of the number of correct attributes to the total number of asserted attributes.

Results. As shown in Table 1, BrepLLM achieves state-of-the-art performance across all evaluation dimensions. It obtains a Qwen3-30B score of 61.36, outperforming the strongest baseline (MiniGPT-3D) by 4.78 points. Furthermore, it improves the Sentence-BERT and SimCSE similarity scores by 3.78 and 4.10, respectively, demonstrating its superior understanding of fine-grained geometric and parametric structures. In human evaluation, BrepLLM achieves a precision of 83.62 and the lowest hallucination rate of 0.83, surpassing all baselines in generating accurate and structurally faithful descriptions. It is worth mentioning that our 8.4B BrepLLM comprehensively outperforms larger 13B-parameter models in both automatic and human evaluations. These results validate BrepLLM's strong capability to capture CAD-specific semantics and faithfully reflect them, showcasing the fine-grained 3D understanding inherited from B-rep representation learning.

Table 2: Generative 3D object classification on the Brep2Text test split. We report accuracy under two prompts (I/C) and their arithmetic mean.

Model	LLM Size	Input	I (%)	C (%)	Average
LLaVA-13B [25]	13B	Single-V. Img.	46.5	44.5	45.5
Qwen3-VL-8B [4]	8B	Tri-V. Img.	54.0	55.0	54.5
PointLLM-7B [43]	7B	3D Point Cloud	52.5	51.0	51.75
PointLLM-13B [43]	13B	3D Point Cloud	53.0	51.5	52.25
ShapeLLM-7B [30]	7B	3D Point Cloud	53.5	52.5	53.0
ShapeLLM-13B [30]	13B	3D Point Cloud	54.5	53.0	53.75
Minigpt-3D [34]	**2.7B**	3D Point Cloud	<u>56.0</u>	<u>56.5</u>	<u>56.25</u>
BrepLLM	8.4B	**B-rep**	**60.5** (+4.5)	**59.5** (+3.0)	**60.0** (+3.75)

5.3 Generative 3D Object Classification

To evaluate BrepLLM's categorical understanding of parametric 3D CAD geometry, we formulate a generative object classification task based on free-form text generation on the Brep2Text test split.

Settings. Each sample is queried using two prompt styles: instructional (I) "What is this?", and completion (C) "This is an object of ". Given the B-rep input and a prompt, the model generates a text response. Following the evaluation protocol of the captioning task, we employ Qwen3-30B as an automatic judge to determine whether the generated response matches the ground-truth category. We report the classification accuracy (%) under the instructional (I) and completion (C) settings, as well as their arithmetic mean (*Average*).

Results. As shown in Table 2, BrepLLM achieves an accuracy of 60.5% under instructional prompting (I) and 59.5% under completion prompting (C), resulting in an average accuracy of 60.0%. This result comprehensively outperforms the baselines, exceeding them by margins of +4.5, +3.0, and +3.75 percentage points on the I, C, and *Average* metrics, respectively. Notably, BrepLLM exhibits highly consistent performance across different prompt formats—the gap between I and C is only 1.0 percentage point, reflecting its robustness to prompt variations. Furthermore, our 8.4B-parameter BrepLLM surpasses all larger 13B-parameter baselines. This superiority stems from its direct modeling of B-rep structures, which explicitly encodes geometric and topological information, thereby enabling reliable category inference even in open-ended generative scenarios.

5.4 Ablation Studies

Under the same settings and evaluation protocol as the main experiments, we conduct ablation studies on the core components to quantify their necessity and contribution to downstream task performance.

Adaptive UV sampling. As shown in Table 3, adaptive UV sampling brings a significant performance improvement of +3.24% in the first stage (Stage I). Although the relative gain slightly narrows after full training (Stage I+II), it still maintains a stable contribution of +2.15%. This suggests that its primary role

Table 3: Adaptive UV sampling.

Stage	w/o (%)	w/ (%)	Δ
I	40.59	43.83	+3.24
I+II	59.21	61.36	+2.15

Table 4: Hierarchical BrepEncoder.

Stage	w/o (%)	w/ (%)	Δ
I	39.64	43.83	+4.19
I+II	57.39	61.36	+3.97

Table 5: Training process.

I	II	Accuracy (%)
✓		43.83
	✓	59.12
✓	✓	61.36

is to enhance the model's early perception of fine-grained geometric structures. Even as the model's overall semantic understanding capabilities improve, adaptive UV sampling remains a crucial module for capturing precise surface details.

Hierarchical BrepEncoder. As shown in Table 4, introducing hierarchical feature extraction brings significant and consistent improvements across all training stages, increasing accuracy by +4.19% in Stage I and +3.97% in Stage I+II. This indicates that explicitly modeling the hierarchical structure of B-reps can effectively capture multi-granular geometric information, thereby significantly enhancing the model's deep semantic understanding of complex CAD objects.

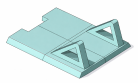
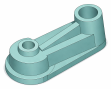
Training process. As shown in Table 5, we validate the proposed two-stage fine-tuning scheme. When only Stage I is enabled, the accuracy is 43.83%, indicating that while the initial geometry-to-vision mapping is helpful, it is insufficient to support high-quality language generation. When training with only Stage II, the accuracy reaches 59.12%, demonstrating that 3D-language alignment is the primary driver of performance improvement. However, skipping Stage I limits the model's geometric prior knowledge. Activating both stages simultaneously achieves the best performance with an accuracy of 61.36%. This demonstrates that adopting a progressive training strategy can more fully unleash the model's potential.

5.5 Qualitative Results

Figure 1 qualitatively demonstrates BrepLLM's strong capability in perceiving and reasoning about CAD data. Through natural language interaction, BrepLLM can generate technically accurate descriptions of CAD models. It not only identifies the overall shape but also extracts specific dimensional information (e.g., length, width, and extrusion thickness), and clearly explains complex sequences of modeling operations, such as Extruded Cut and Circular Pattern.

Furthermore, Table 12 presents a comparison between BrepLLM and the baseline on 3D object captioning. As shown in Sample 1, when faced with complex geometries, the baseline's description is relatively vague, whereas BrepLLM successfully identifies fine-grained part details, such as beveled side edges and internal triangular cutouts, providing a technically precise description that highly aligns with the ground truth. In Sample 2, which focuses on the model's construction method, BrepLLM accurately parses the topological structure of the model. It not only generates a comprehensive geometric description but also precisely localizes the spatial composition of various components, such as a rectangular

Table 6: Qualitative comparison for 3D object captioning on Brep2Text dataset against the next-best baseline. Green highlights indicate salient details that align with the ground truth.

Samples 1, 2		
Prompt	*What is this?*	*How this CAD model was constructed, describe this CAD model in detail.*
Ground Truth	This is a rectangular plate with two triangular braces on its front face and a flat back; both side edges are beveled and slope upward, and there is a centered rectangular notch along the bottom edge.	The design features a rounded rectangular base, a larger cylindrical body, a smaller cylindrical body, a cylindrical shape with a rectangular extension and a central hole, and a rectangular prism with a cylindrical base.
MiniGPT-3D	This is a rectangular CAD model with two triangular brackets on each side, integrated into the rectangular plate plane and with a cut at the bottom.	This model consists of a rectangular base, a large cylinder, and a small cylinder. It is made by stretching a rectangular base and adding circular features on both sides.
BrepLLM	**This is a CAD model based on a flat rectangular plate. It features a centered rectangular notch on the bottom edge and a flat back. The front face has two symmetrically positioned triangular braces with internal triangular cutouts,and both side edges are slanted faces.**	**The model is built on a rectangular base with rounded corners. It contains a larger cylinder and a smaller cylinder at each end. It combines a cylindrical body with a rectangular extension and a central hole, as well as a straight prism with a cylindrical base.**

base with rounded corners and large and small cylinders located at both ends. Its grasp of detail is significantly superior to that of the baseline. More qualitative results and comparisons with other models are provided in the Appendix.

6 Conclusions and Future Work

We introduced BrepLLM, the first multimodal framework that enables LLMs to directly parse and reason over raw B-rep data. The framework employs a two-stage training architecture comprising cross-modal alignment pre-training

and LLM fine-tuning. It generates compact tokens fused with geometric and topological information and seamlessly integrates them into the LLM, thereby achieving an in-depth understanding of 3D geometry.

Our main contributions include: (i) an adaptive UV sampling strategy to construct graph representations that fuse geometric and topological information; (ii) a hierarchical BrepEncoder (face/edge/global topology) for extracting global and node tokens; (iii) a CLIP-style contrastive learning approach to align B-rep features into a unified geometry-language space; (iv) a two-stage progressive 3D-language semantic alignment strategy; and (v) Brep2Text, a large-scale benchmark dataset specifically designed for B-rep understanding.

This study demonstrates that processing native B-reps directly enhances LLMs' geometric and topological understanding. Future work will scale up high-quality multimodal B-rep data to improve generalization. We also plan to extend this framework to B-rep generation, editing, and assembly understanding, exploring its potential in industrial intelligent CAD workflows.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62506302.

References

1. Anil, R., Dai, A.M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J.H., Shafey, L.E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G.H., Ahn, J., Austin, J., Barham, P., Botha, J., Bradbury, J., Brahma, S., Brooks, K., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C.A., Chowdhery, A., Crepy, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., Díaz, M., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., Garcia, X., Gehrmann, S., Gonzalez, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M., Li, W., Li, Y., Li, J., Lim, H., Lin, H., Liu, Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, A., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A.C., Roy, A., Saeta, B., Samuel, R., Shelby, R., Slone, A., Smilkov, D., So, D.R., Sohn, D., Tokumine, S., Valter, D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L., Yin, P., Yu, J., Zhang, Q., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S., Wu, Y.: Palm 2 technical report (2023), https://arxiv.org/abs/2305.10403
2. Badagabettu, A., Yarlagadda, S.S., Farimani, A.B.: Query2cad: Generating cad models using natural language queries. arXiv preprint arXiv:2406.00144 (2024)
3. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond (2023), https://arxiv.org/abs/2308.12966

4. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report (2025), https://arxiv.org/abs/2511.21631
5. Doris, A.C., Alam, M.F., Nobari, A.H., Ahmed, F.: Cad-coder: An open-source vision-language model for computer-aided design code generation (2025), https://arxiv.org/abs/2505.14646
6. Du, Y., Chen, S., Zan, W., Li, P., Wang, M., Song, D., Li, B., Hu, Y., Wang, B.: Blenderllm: Training large language models for computer-aided design with self-improvement (2024), https://arxiv.org/abs/2412.14203
7. Fan, J., Gholami, B., Bäck, T., Wang, H.: Neuronurbs: Learning efficient surface representations for 3d solids (2024), https://arxiv.org/abs/2411.10848
8. Gao, T., Yao, X., Chen, D.: Simcse: Simple contrastive learning of sentence embeddings (2022), https://arxiv.org/abs/2104.08821
9. Gao, Z., Li, Z., Wang, X., Huang, J., Ren, Z., Shao, M., Zhang, H., Huang, T., Cheng, Y., Guo, Y., Lin, R., Wang, Y., Liu, T., Zhang, K., Gong, M.: Mirage2matter: A physically grounded gaussian world model from video (2026), https://arxiv.org/abs/2602.00096
10. Guo, J., Chen, Z., Li, Z., Gao, Z., Huang, J., Zhang, H., Huang, F., Yao, Y., Liu, T., Gong, M.: Huge-bench: A benchmark for high-level uav vision-language-action tasks. arXiv preprint arXiv:2603.19822 (2026)
11. Guo, Z., Zhang, R., Zhu, X., Tang, Y., Ma, X., Han, J., Chen, K., Gao, P., Li, X., Li, H., Heng, P.A.: Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following (2023), https://arxiv.org/abs/2309.00615
12. Heidari, N., Iosifidis, A.: Geometric deep learning for computer-aided design: A survey (2025), https://arxiv.org/abs/2402.17695
13. Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., Gan, C.: 3d-llm: Injecting the 3d world into large language models (2023), https://arxiv.org/abs/2307.12981
14. Jayaraman, P.K., Lambourne, J.G., Desai, N., Willis, K.D.D., Sanghi, A., Morris, N.J.W.: Solidgen: An autoregressive model for direct b-rep synthesis (2023), https://arxiv.org/abs/2203.13944
15. Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., de las Casas, D., Hanna, E.B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L.R., Saulnier, L., Lachaux, M.A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T.L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mixtral of experts (2024), https://arxiv.org/abs/2401.04088
16. Khan, M.S., Sinha, S., Uddin, T., Stricker, D., Ali, S.A., Afzal, M.Z.: Text2cad: Generating sequential cad designs from beginner-to-expert level text prompts. Advances in Neural Information Processing Systems **37**, 7552–7579 (2024)
17. Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., Panozzo, D.: ABC: A big CAD model dataset for geometric deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9601–9611 (2019), https://arxiv.org/abs/1812.06216

18. Kolodiazhnyi, M., Tarasov, D., Zhemchuzhnikov, D., Nikulin, A., Zisman, I., Vorontsova, A., Konushin, A., Kurenkov, V., Rukhovich, D.: cadrille: Multi-modal cad reconstruction with reinforcement learning (2026), https://arxiv.org/abs/2505.22914
19. Li, J., Bai, Y., Dai, Y., Guo, H., Gan, H., Shi, Y.: Autoregressive generation with b-rep holistic token sequence representation (2026), https://arxiv.org/abs/2601.16771
20. Li, J., Ma, W., Li, X., Lou, Y., Zhou, G., Zhou, X.: Cad-llama: Leveraging large language models for computer-aided design parametric 3d model generation (2025), https://arxiv.org/abs/2505.04481
21. Li, J., Fu, Y., Chen, F.: Dtgbrepgen: A novel b-rep generative model through decoupling topology and geometry (2025), https://arxiv.org/abs/2503.13110
22. Li, J., et al.: CAD Translator: An effective drive for text to 3d parametric computer-aided design generative modeling. In: Proceedings of the 32nd ACM International Conference on Multimedia (2024). https://doi.org/10.1145/3664647.3681549, https://dl.acm.org/doi/10.1145/3664647.3681549
23. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models (2023), https://arxiv.org/abs/2301.12597
24. Li, X., Sun, Y., Sha, Z.: Llm4cad: Multi-modal large language models for 3d computer-aided design generation. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. vol. 88407, p. V006T06A015. American Society of Mechanical Engineers (2024)
25. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning (2023), https://arxiv.org/abs/2304.08485
26. Liu, Y., Xu, D., Yu, X., Xu, X., Cohen-Or, D., Zhang, H., Huang, H.: HoLa: B-rep generation using a holistic latent representation. ACM Transactions on Graphics **44**(4), 116 (2025). https://doi.org/10.1145/3730842, https://arxiv.org/abs/2504.14257
27. Mallis, D., Karadeniz, A.S., Cavada, S., Rukhovich, D., Foteinopoulou, N., Cherenkova, K., Kacem, A., Aouada, D.: Cad-assistant: Tool-augmented vllms as generic cad task solvers. arXiv preprint arXiv:2412.13810 (2024)
28. OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H.W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S.P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S.S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N.S., Khan, T., Kilpatrick, L., Kim, J.W., Kim, C., Kim, Y., Kirchner, J.H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk,

Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C.M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S.M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H.P., Michael, Pokorny, Pokrass, M., Pong, V.H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F.P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M.B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J.F.C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J.J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., Zoph, B.: Gpt-4 technical report (2024), https://arxiv.org/abs/2303.08774

29. Qi, Z., Dong, R., Zhang, S., Geng, H., Han, C., Ge, Z., Yi, L., Ma, K.: Shapellm: Universal 3d object understanding for embodied interaction (2024), https://arxiv.org/abs/2402.17766
30. Qi, Z., Dong, R., Zhang, S., Geng, H., Han, C., Ge, Z., Yi, L., Ma, K.: Shapellm: Universal 3d object understanding for embodied interaction (2024), https://arxiv.org/abs/2402.17766
31. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021), https://arxiv.org/abs/2103.00020
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
33. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks (2019), https://arxiv.org/abs/1908.10084
34. Tang, Y., Han, X., Li, X., Yu, Q., Hao, Y., Hu, L., Chen, M.: Minigpt-3d: Efficiently aligning 3d point clouds with large language models using 2d priors (2024), https://arxiv.org/abs/2405.01413
35. Tang, Y., Han, X., Li, X., Yu, Q., Hao, Y., Hu, L., Chen, M.: Minigpt-3d: Efficiently aligning 3d point clouds with large language models using 2d priors (2024), https://arxiv.org/abs/2405.01413
36. Team, Q.: Qwen3 technical report (2025), https://arxiv.org/abs/2505.09388
37. Wang, S., Chen, C., Le, X., Xu, Q., Xu, L., Zhang, Y., Yang, J.: Cad-gpt: Synthesising cad construction sequence with spatial reasoning-enhanced multimodal

llms. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 7880–7888 (2025)
38. Wu, S., Khasahmadi, A., Katz, M., Jayaraman, P.K., Pu, Y., Willis, K., Liu, B.: Cad-llm: Large language model for cad generation. In: Proceedings of the neural information processing systems conference. neurIPS. vol. 1 (2023)
39. Wu, S., Khasahmadi, A.H., Katz, M., Jayaraman, P.K., Pu, Y., Willis, K., Liu, B.: Cadvlm: Bridging language and vision in the generation of parametric cad sketches. In: European Conference on Computer Vision. pp. 368–384. Springer (2024)
40. Wu, X., Jiang, L., Wang, P.S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H.: Point transformer v3: Simpler, faster, stronger. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4840–4851 (2023), https://api.semanticscholar.org/CorpusID:266335894
41. Xie, H., Ju, F.: Text-to-cadquery: A new paradigm for cad generation with scalable large model capabilities (2025), https://arxiv.org/abs/2505.06507
42. Xu, J., Zhao, Z., Wang, C., Liu, W., Ma, Y., Gao, S.: Cad-mllm: Unifying multimodality-conditioned cad generation with mllm. arXiv preprint arXiv:2411.04954 (2024)
43. Xu, R., Wang, X., Wang, T., Chen, Y., Pang, J., Lin, D.: Pointllm: Empowering large language models to understand point clouds (2024), https://arxiv.org/abs/2308.16911
44. Xu, R., Wang, X., Wang, T., Chen, Y., Pang, J., Lin, D.: Pointllm: Empowering large language models to understand point clouds (2024), https://arxiv.org/abs/2308.16911
45. Xu, X., Jayaraman, P.K., Lambourne, J.G., Liu, Y., Malpure, D., Meltzer, P.: AutoBrep: Autoregressive b-rep generation with unified topology and geometry. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–12 (2025). https://doi.org/10.1145/3757377.3763814, https://arxiv.org/abs/2512.03018
46. Xu, X., Lambourne, J.G., Jayaraman, P.K., Wang, Z., Willis, K.D.D., Furukawa, Y.: Brepgen: A b-rep generative diffusion model with structured latent geometry (2024), https://arxiv.org/abs/2401.15563
47. Yuan, Y., Sun, S., Liu, Q., Bian, J.: Cad-editor: Text-based cad editing through adapting large language models with synthetic data
48. Yuan, Z., Li, Z., Huang, W., Ye, Y., Sun, L.: Tinygpt-v: Efficient multimodal large language model via small backbones (2024)
49. Zeng, Q., Garay, L., Zhou, P., Chong, D., Hua, Y., Wu, J., Pan, Y., Zhou, H., Voigt, R., Yang, J.: Greenplm: Cross-lingual transfer of monolingual pre-trained language models at almost no cost (2023), https://arxiv.org/abs/2211.06993
50. Zhang, Z., Sun, S., Wang, W., Cai, D., Bian, J.: Flexcad: Unified and versatile controllable cad generation with fine-tuned large language models (2025), https://arxiv.org/abs/2411.05823
51. Zhu, C., Wang, T., Zhang, W., Pang, J., Liu, X.: Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. arXiv preprint arXiv:2409.18125 (2024)