BREPASSEMBLYLLM: REASONING OVER BREPS FOR EXECUTABLE CAD ASSEMBLY

Anonymous authors
Paper under double-blind review

ABSTRACT

CAD assembly requires identifying mating entities between parts and estimating a rigid transformation that yields a valid relative pose. Extending large language models (LLMs) to this task is challenging: native Boundary Representations (BReps) exhibit heterogeneous face–edge graph structures, the candidate mating space grows rapidly with entity count, and assembly inherently couples discrete entity correspondence with continuous pose prediction. We present BRepAssemblyLLM, the first geometry-conditioned LLM framework for pairwise CAD assembly. The framework consists of a topology-aware BRep encoder, a lightweight projector, an LLM, and a CAD assembly executor. Given two CAD parts, the BRep encoder jointly encodes face geometry, edge geometry, and their topological relations, while the projector maps the resulting representations into the LLM embedding space. The LLM then predicts a structured assembly action comprising a pair of mating entities and a rigid transformation matrix specifying their relative pose. Subsequently, the CAD assembly executor executes the predicted action and provides contact and interpenetration feedback, which is used to jointly optimize mating-entity selection and pose prediction via reinforcement learning. On the Fusion 360 Gallery benchmark, BRepAssemblyLLM achieves state-of-the-art performance. These results demonstrate that LLMs can move beyond understanding and generating CAD descriptions or code to directly reason over native BReps and produce executable CAD assembly actions. The code will be released.

1 INTRODUCTION

Computer-aided design (CAD) intelligence (Deng et al., 2025; Shui et al., 2026; Zhang et al., 2025b; Tomiyama, 2007; Zhang et al., 2026; 2025a) should produce editable geometric actions, rather than merely descriptions, labels, or visually plausible shapes. Pairwise assembly is a representative test of this capability: two parts must be connected through precise topological entities and positioned by a rigid motion consistent with the intended joint. Existing CAD assembly systems (Jones et al., 2021; Thomas et al., 2018; Campi et al., 2022) have shown that native mating entities provide an effective and editable representation of design intent. We therefore study assembly as an executable reasoning problem for large language models (LLMs): given two parts in native boundary representation (BRep) form, predict the mating-entity indices and a 4×4 rigid transformation that realizes the assembly. This task couples discrete entity correspondence with continuous geometric reasoning (Ou & Xu, 2013). The model must select a compatible entity pair from the dense Cartesian product across two parts, infer geometric relations such as coincident axes, compatible radii, opposing normals, and planar or cylindrical contact, and use adjacent edges as local topological context. The predicted transformation must then bring the selected entities into contact without causing interpenetration. These objectives are inherently coupled: a plausible pose is not an editable CAD joint unless grounded in native entities, while a correct entity pair does not define a valid assembly without an executable pose. Native BReps are well suited to this setting because, unlike point clouds or meshes, they preserve parametric surfaces, curves, incidence relations, and stable entity identifiers used by CAD kernels. Large-scale CAD datasets (Koch et al., 2019; Willis et al., 2021) and native BRep encoders (Usama et al., 2026; Lambourne et al., 2021; Jones et al., 2023) have demonstrated the value of such structured representations for CAD learning. However, prior BRep models mainly predict per-part labels or embeddings, whereas assembly additionally requires cross-part entity correspondence and continuous pose estimation. Conversely, recent 3D-language models (Xu et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

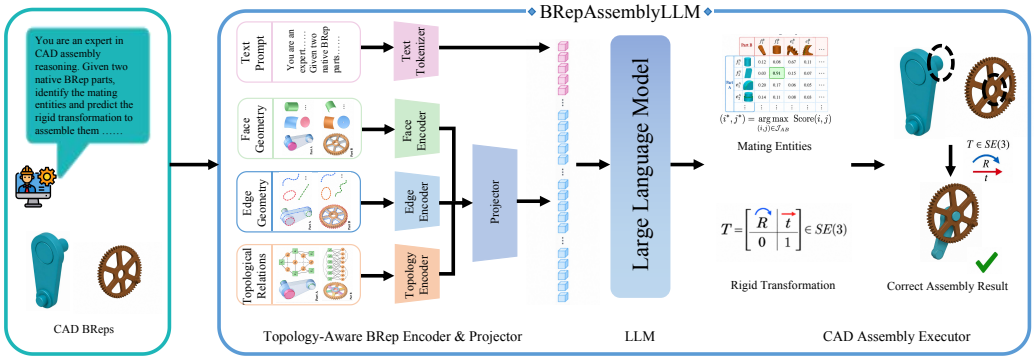


Figure 1: **Overview of BRepAssemblyLLM.** Given two native BRep CAD parts, a topology-aware BRep encoder extracts face and edge features and their topological relations, which are projected into the LLM embedding space. The LLM predicts a mating-entity pair and a rigid transformation that are executed in a CAD kernel to produce the assembly.

2024; Wang et al., 2023; Hong et al., 2023) connect geometric features to LLMs, but their tokens are generally not grounded in CAD-kernel entity identities, and their outputs cannot be directly executed as native assembly operations.

We introduce BRepAssemblyLLM, a geometry-conditioned LLM framework that reasons directly over native BReps, as illustrated in Fig. 1. We treat the LLM as a structured geometric policy over native BRep entities, with the BRep topology graph providing assembly context. A topology-aware BRep encoder extracts face and edge features, while topology-masked multi-head attention enables information exchange between incident entities. A lightweight projector maps the resulting geometry tokens into the LLM embedding space, while preserving native entity indices for direct entity grounding. Given two CAD parts, the LLM predicts a structured assembly action consisting of a pair of mating entities and a 4×4 rigid transformation, which is then executed by a CAD assembly executor. We train BRepAssemblyLLM in three stages. Cross-modal alignment pretraining first aligns BRep geometry with textual representations. Supervised fine-tuning then learns mating-entity selection. Finally, geometric feedback from the CAD executor, including entity correctness, surface contact, and interpenetration, is used to jointly optimize entity selection and rigid-pose prediction through reinforcement learning. On the Fusion 360 Gallery benchmark, BRepAssemblyLLM achieves state-of-the-art performance, with 89.61% joint connectivity accuracy and a pose Chamfer Distance of 0.0321. These results show that preserving native BRep entities and topology supports both accurate mating-entity selection and geometric pose estimation, enabling discrete correspondence and continuous pose reasoning within a unified LLM framework. More broadly, BRepAssemblyLLM demonstrates that LLMs can move beyond understanding or generating CAD descriptions and code to directly reason over native CAD representations and produce executable geometric actions. Our contributions are:

- We formulate pairwise CAD assembly as an executable BRep reasoning task for LLMs and introduce BRepAssemblyLLM, the first LLM framework for native CAD assembly, enabling joint prediction of native mating entities and rigid assembly poses.
- We introduce a topology-aware BRep-to-token representation that jointly encodes face geometry, edge geometry, and topological relations, effectively mapping native BRep structures into the LLM embedding space.
- We propose a three-stage training strategy that integrates BRep–text cross-modal alignment, supervised mating-entity learning, and geometric-feedback reinforcement learning to progressively improve executable assembly reasoning.
- We achieve state-of-the-art performance on the Fusion 360 Gallery assembly benchmark. The results validate the effectiveness of our framework and demonstrate the potential of LLMs to reason directly over native BReps and generate executable CAD actions.

2 Related Work

2.1 Native CAD Representations and CAD Assembly

Native CAD representations preserve the parametric geometry and topology used by CAD kernels, with Boundary Representations (BReps) explicitly encoding faces, edges, surfaces, curves, and their incidence relations. Large-scale datasets such as ABC (Koch et al., 2019) and Fusion 360 Gallery (Willis et al., 2021) have enabled learning directly from such structured representations. UV-Net (Jayaraman et al., 2021) combines UV-domain geometry with graph convolution, while BRep-Net (Lambourne et al., 2021) captures local topology through oriented coedges. Self-supervised learning (Jones et al., 2023) further demonstrates the effectiveness of unlabeled BReps, and Solid-Gen (Jayaraman et al., 2022) directly models indexed BRep topology for generation. Building on these native representations, CAD assembly requires reasoning across multiple parts to determine both entity correspondence and relative pose. AutoMate (Jones et al., 2021) learns reference-based CAD mates, while JoinABLe (Willis et al., 2022) predicts connectivity over dense cross-part entity pairs and recovers poses through geometric search. Beyond native CAD, prior work predicts 6-DoF part poses (Huang et al., 2020), learns fragment reassembly (Chen et al., 2022), and exploits $SE(3)$-equivariant representations for geometric assembly (Wu et al., 2023). Recent methods further model contacting joints (Li et al., 2024) and part–whole hierarchies (Du et al., 2024). However, most of these approaches rely on point clouds or meshes and do not preserve persistent CAD entity references. In contrast, BRepAssemblyLLM reasons directly over native BReps, combining entity grounding, LLM-based action prediction, and CAD-kernel execution for executable assembly.

2.2 CAD Construction and CAD Large Language Models

Many methods formulate CAD construction as structured generation. SketchGraphs (Seff et al., 2020), Vitruvion (Seff et al., 2022), DeepCAD (Wu et al., 2021), and SkexGen (Xu et al., 2022) model sketches or sketch–extrude sequences, while SECAD-Net (Li et al., 2023) reconstructs editable programs and SolidGen (Jayaraman et al., 2022) directly generates BReps. More recent language- and multimodal-based systems extend this paradigm to CAD and 3D reasoning. Text2CAD (Yavartanoo et al., 2024), CAD-GPT (Wang et al., 2025a), CAD-Llama (Li et al., 2025), CAD-Recode (Rukhovich et al., 2025), and CAD-Coder (Guan et al., 2025) connect language or multimodal inputs to parametric CAD generation, while 3D-LLM (Hong et al., 2023) integrates 3D representations with LLMs. CAD-CLIP (Usama et al., 2026) further aligns native BRep features with textual representations, and concurrent BRep language models explore native-shape understanding and code generation (Deng et al., 2025; Kim et al., 2026). In contrast, we focus on assembling two existing solids by directly predicting native mating entities and their relative rigid transformation.

3 Method

3.1 Overview

BRepAssemblyLLM formulates pairwise CAD assembly as structured generation over native BReps. Given two CAD parts, a topology-aware BRep encoder extracts geometry tokens, which are projected into the LLM embedding space. The LLM predicts mating entities and a rigid transformation, which are executed by a CAD kernel to provide geometric feedback for reinforcement learning. We next introduce the assembly action space (Sec. 3.2), model architecture (Sec. 3.3), and three-stage training strategy (Sec. 3.4).

3.2 Assembly Action Space

Let each part $k \in \{A, B\}$ be represented as a native BRep graph:

$$\mathcal{G}_k = (\mathcal{F}_k, \mathcal{E}_k, \mathcal{C}_k), \tag{1}$$

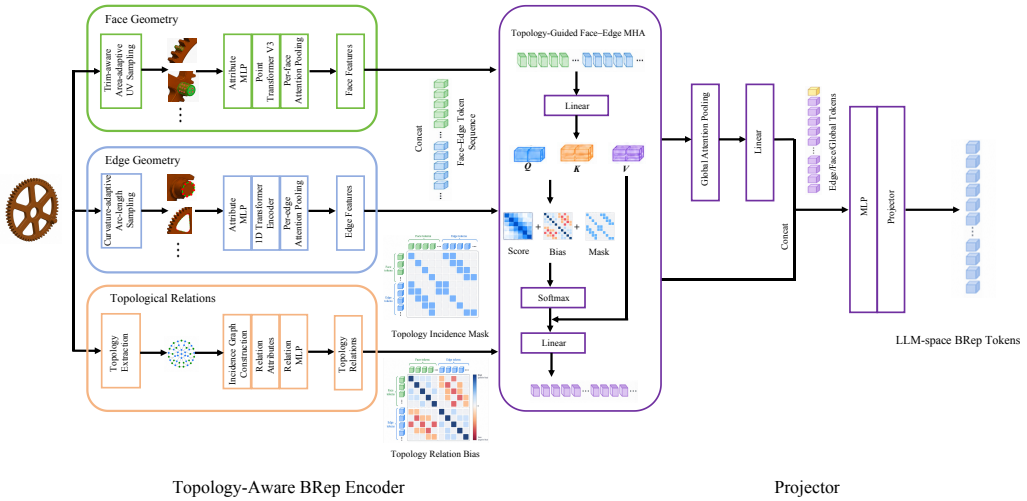


Figure 2: **Topology-Aware BRep encoder and projector.** Face and edge geometry, together with their topological relations, are encoded into tokens, refined by Topology-Guided Face–Edge MHA, and projected into the LLM embedding space by a lightweight projector.

where $\mathcal{F}_k$ and $\mathcal{E}_k$ denote its faces and edges, respectively, and $\mathcal{C}_k$ denotes the face–edge incidence relation. The candidate mating space is defined as:

$$\mathcal{J}_{AB} = \bigcup_{\tau \in \{F,E\}} \mathcal{S}_A^{\tau} \times \mathcal{S}_B^{\tau}. \tag{2}$$

BRepAssemblyLLM assigns a compatibility score to each candidate mating pair and selects the highest-scoring pair:

$$p^* = (i^*, j^*) = \underset{(i,j)\in\mathcal{J}_{AB}}{\arg\max}\ \mathrm{Score}(i,j), \tag{3}$$

$s_{i^*}^{A,\tau} \in \mathcal{S}_A^{\tau}$ and $s_{j^*}^{B,\tau} \in \mathcal{S}_B^{\tau}$ refer directly to the selected native entities in Parts A and B, respectively. Meanwhile, BRepAssemblyLLM predicts a continuous rigid transformation that maps Part B into the coordinate frame of Part A:

$$\mathbf{T}_{B\to A} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\top} & 1 \end{bmatrix} \in \mathrm{SE}(3), \qquad \mathbf{R} \in \mathrm{SO}(3),\ \mathbf{t} \in \mathbb{R}^3. \tag{4}$$

The complete assembly action is represented as:

$$y = \begin{pmatrix} \texttt{Mating Entity Indices} : [i^*, j^*], \\ \texttt{Rigid Transformation} : \mathbf{T}_{B\to A} \end{pmatrix}. \tag{5}$$

The native entity indices ground the action in the original CAD solids, while the rigid transformation specifies the executable relative pose.

3.3 BRepAssemblyLLM Architecture

3.3.1 Topology-Aware BRep Encoder and Projector

The overall pipeline is shown in Fig. 2. We represent each BRep as a heterogeneous face–edge graph $\mathcal{G} = (\mathcal{F}, \mathcal{E}, \mathcal{C})$. For faces, we adopt trim-aware, area-adaptive UV sampling (Jayaraman et al., 2021), followed by an attribute MLP (Lambourne et al., 2021), Point Transformer V3 (Wu et al., 2024), and attention pooling to obtain local face tokens. For edges, curvature-adaptive sampling (Miao et al., 2009) is applied along the 3D curve, followed by an attribute MLP, a 1D Transformer, and attention pooling to obtain local edge tokens.

To incorporate BRep topology, face and edge tokens are concatenated and processed by stacked Topology-Guided Face–Edge MHA layers. The face–edge incidence graph defines a topology mask,

while coedge orientation, loop membership, and edge convexity are encoded as head-wise relation biases. For attention head h,

$$\mathbf{A}^{(h)} = \mathrm{Softmax}\left(\frac{\mathbf{Q}^{(h)}\mathbf{K}^{(h)\top}}{\sqrt{d_h}} + \mathbf{B}_{\mathrm{rel}}^{(h)} + \mathbf{M}_{\mathrm{inc}}\right), \tag{6}$$

where $\mathbf{B}_{\mathrm{rel}}^{(h)}$ is the topology relation bias and $\mathbf{M}_{\mathrm{inc}}$ restricts attention to self-connections and incident face–edge pairs. Stacking these layers enables bidirectional information exchange between topologically related entities while preserving their native identities.

The topology-aware entity tokens are then aggregated by global attention pooling to obtain a part-level token $\mathbf{h}_{\mathrm{global}}$. The complete BRep token sequence is

$$\mathbf{H}_B = [\mathbf{h}_{\mathrm{global}}; \mathbf{h}_1^F; \ldots; \mathbf{h}_{|\mathcal{F}|}^F; \mathbf{h}_1^E; \ldots; \mathbf{h}_{|\mathcal{E}|}^E]. \tag{7}$$

We further add entity-type, native-index, and part-ID embeddings to preserve CAD-specific identities. Finally, a lightweight MLP projector maps the BRep tokens into the LLM embedding space. The two input parts share the same encoder and projector, producing entity-grounded tokens for mating-entity selection and rigid-transformation prediction.

3.3.2 Backbone Instantiation and CAD Executor

We adopt Qwen2.5-VL-3B (Bai et al., 2025) as the LLM backbone, although no image inputs are used in the current experiments. The projected BRep features are provided as continuous non-text token embeddings and processed jointly with the textual prompt. The LLM predicts a structured assembly action comprising a pair of native mating entities and a rigid transformation matrix $\mathbf{T}_{B\to A}$. A deterministic, non-learnable CAD executor then resolves the predicted entity indices, keeps Part A fixed, and transforms Part B into the predicted pose, thereby producing the final CAD assembly.

3.4 Training Strategy

As shown in Fig. 3, we train BRepAssemblyLLM using a three-stage curriculum that progressively aligns BRep geometry with language, learns mating-entity selection, and optimizes the complete executable assembly action.

3.4.1 Stage I: Cross-Modal Alignment Pretraining

Stage I aligns native BRep geometry with language before assembly supervision. Given paired BReps and descriptions $\{(\mathcal{G}_k, d_k)\}_{k=1}^N$, the BRep encoder and projector produce a normalized geometry embedding:

$$\mathbf{h}_{b,k} = \mathrm{norm}\Big(\mathcal{P}\big(\mathcal{E}_{\mathrm{BRep}}^{\mathrm{global}}(\mathcal{G}_k)\big)\Big), \tag{8}$$

while a frozen text encoder produces:

$$\mathbf{h}_{t,k} = \mathrm{norm}(\mathcal{E}_{\mathrm{text}}(d_k)). \tag{9}$$

Their similarity is:

$$s_{k\ell} = \frac{\mathbf{h}_{b,k}^\top \mathbf{h}_{t,\ell}}{\tau}, \tag{10}$$

where τ is the temperature. We optimize the BRep encoder and projector using symmetric InfoNCE:

$$\mathcal{L}_{b\to t} = -\frac{1}{N}\sum_{k=1}^{N} \log \frac{\exp(s_{kk})}{\sum_{\ell=1}^{N}\exp(s_{k\ell})}, \tag{11}$$

$$\mathcal{L}_{t\to b} = -\frac{1}{N}\sum_{k=1}^{N} \log \frac{\exp(s_{kk})}{\sum_{\ell=1}^{N}\exp(s_{\ell k})}, \tag{12}$$

$$\mathcal{L}_{\mathrm{align}} = \frac{1}{2}\left(\mathcal{L}_{b\to t} + \mathcal{L}_{t\to b}\right). \tag{13}$$

This stage initializes geometry representations aligned with the language space.

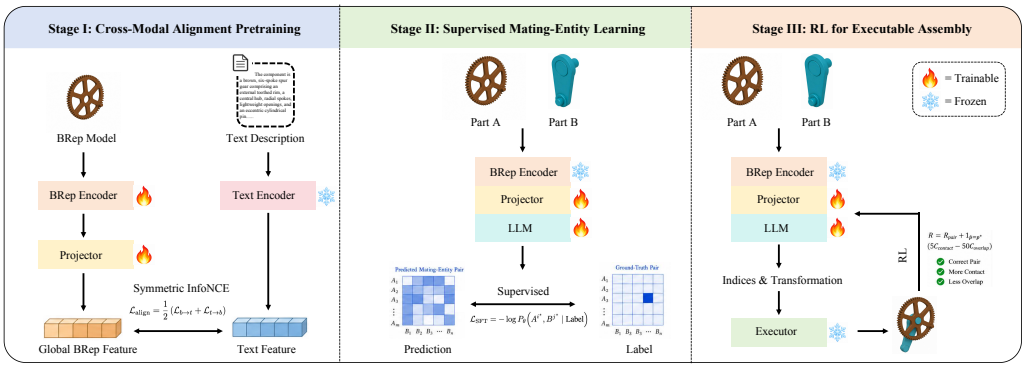


Figure 3: **Three-stage training curriculum of BRepAssemblyLLM.** Stage I aligns BRep and text representations, Stage II learns mating-entity selection, and Stage III optimizes executable assembly using CAD-executor feedback. Flame and snowflake symbols denote trainable and frozen modules, respectively.

3.4.2 Stage II: Supervised Mating-Entity Learning

Stage II trains the LLM to select native mating entities. For Part $X \in \{A, B\}$, let $\mathcal{F}_X$ and $\mathcal{E}_X$ denote its face and edge sets. The candidate space is

$$\mathcal{J}_{AB} = (\mathcal{F}_A \times \mathcal{F}_B) \cup (\mathcal{E}_A \times \mathcal{E}_B). \tag{14}$$

The BRep encoder is frozen, while the projector and LLM are optimized. Given a ground-truth pair $p^\star = (i^\star, j^\star)$, its entity indices are serialized as $\mathbf{y}^\star$. Supervised fine-tuning minimizes:

$$\mathcal{L}_{\text{SFT}} = -\sum_{r=1}^{|\mathbf{y}^\star|} \log P_\theta(y_r^\star \mid \mathbf{y}_{<r}^\star, \mathbf{H}_A, \mathbf{H}_B, \mathbf{q}), \tag{15}$$

where $\mathbf{H}_A$ and $\mathbf{H}_B$ are projected BRep tokens and $\mathbf{q}$ is the assembly prompt. At inference,

$$\hat{p} = \arg\max_{p \in \mathcal{J}_{AB}} P_\theta(\mathbf{y}(p) \mid \mathbf{H}_A, \mathbf{H}_B, \mathbf{q}). \tag{16}$$

Stage II learns discrete entity selection, while pose prediction is optimized in Stage III.

3.4.3 Stage III: RL for Executable Assembly

Stage III optimizes the complete assembly action. Starting from Stage II, the projector and LLM define a policy π_θ that predicts:

$$\hat{a} = \left(\hat{p}, \hat{\mathbf{T}}_{B\to A}\right), \tag{17}$$

where $\hat{p}$ is the mating pair and $\hat{\mathbf{T}}_{B\to A} \in \mathrm{SE}(3)$ is:

$$\hat{\mathbf{T}}_{B\to A} = \begin{bmatrix} \hat{\mathbf{R}} & \hat{\mathbf{t}} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{18}$$

A deterministic CAD executor applies the action and evaluates contact and interpenetration. The reward is:

$$R(\hat{a}) = R_{\text{pair}} + \mathbf{1}_{\hat{p}=p^\star}\left(\lambda_c C_{\text{contact}} - \lambda_o C_{\text{overlap}}\right), \tag{19}$$

where

$$R_{\text{pair}} = \begin{cases} 1, & \hat{p} = p^\star, \\ -1, & \text{otherwise}, \end{cases} \tag{20}$$

and $\lambda_c = 5$, $\lambda_o = 50$. Geometric rewards are applied only when the mating pair is correct.

We maximize:

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{\hat{a}\sim\pi_\theta}\left[R(\hat{a})\right], \tag{21}$$

while keeping the BRep encoder frozen and updating the projector and LLM. The CAD executor provides geometric feedback during training and executes the predicted action at inference.

4 EXPERIMENTS

4.1 BENCHMARK AND PROTOCOL

We evaluate BRepAssemblyLLM on the Fusion 360 Gallery assembly benchmark (Willis et al., 2021; 2022). Each example contains two native BRep CAD parts and their assembled model. Given the two BReps, the model predicts a mating pair and a rigid transformation.

4.2 IMPLEMENTATION DETAILS

We use Qwen2.5-VL-3B (Bai et al., 2025) as the LLM backbone without visual inputs. The LLM is frozen, with LoRA adapters (Hu et al., 2021) ($r = 16$, $\alpha = 32$, dropout $= 0.05$). The BRep encoder, MLP projector, and LoRA adapters are trainable. All experiments are conducted on four NVIDIA A800 80GB GPUs. Training follows three stages. In Stage I, we use the frozen CLIP text encoder (Radford et al., 2021) and train the BRep encoder and projector with the BRep–text contrastive objective for 200 epochs using AdamW, a learning rate of 1×10^{-4}, a weight decay of 0.01, and a batch size of 128. In Stage II, the projector and LoRA adapters are jointly trained for 100 epochs with learning rates of 1×10^{-4} and 2×10^{-5}, respectively, and a batch size of 32. In Stage III, the projector and LoRA adapters are optimized with GRPO for 50 epochs using learning rates of 1×10^{-5} and 5×10^{-6} with reward weights $\lambda_c = 5$ and $\lambda_o = 50$.

4.3 BASELINES AND METRICS

Connectivity Prediction. We evaluate connectivity prediction using Top-1 accuracy over the cross-part candidate graph $\mathcal{F}_A \times \mathcal{F}_B$. A prediction is considered correct only when the predicted native entity pair exactly matches the ground-truth mating relation. We compare against four categories of baselines: (1) **Human CAD Expert**, where an expert manually selects the mating pair; (2) **general-purpose multimodal LLMs** (Bai et al., 2025; Wang et al., 2025b; Lu et al., 2024; Liu et al., 2023), which take rendered CAD views and textual descriptions as input and predict the mating relation; (3) **point-cloud-based methods**, including ANCSH (Li et al., 2020), Shape2Motion (Wang et al., 2019), PointLLM-7B (Xu et al., 2024), and ShapeLLM (Qi et al., 2024), which take sampled point clouds as input and predict the mating faces; and (4) **native BRep methods**, which directly take BRep representations as input and predict the mating entity pair. For generative LLM baselines and the Human CAD Expert, we evaluate on a randomly sampled subset of 100 test examples. A prediction is counted as correct only when the generated response unambiguously identifies the ground-truth mating entities.

Pose Prediction. We evaluate the predicted rigid transformation $\mathbf{T}_{B \to A}$ using the Chamfer Distance (CD) between the predicted and ground-truth assembled geometries, where a lower CD indicates a more accurate assembly pose. We compare against B-Pose (Zhan et al., 2020), JoinABLe (Willis et al., 2022), and JoinABLe+Search (Willis et al., 2022). General-purpose image- or point-cloud-based language models are excluded from the pose comparison because they do not directly produce an executable full rigid transformation.

4.4 EXPERIMENTAL RESULTS

Assembly Connectivity Accuracy. Table 1 reports the Top-1 assembly connectivity accuracy. BRepAssemblyLLM achieves the best accuracy of 89.61%, outperforming JoinABLe by 10.08 percentage points and substantially surpassing the Human CAD Expert, image-based multimodal LLMs, and point-cloud-based methods. These results indicate that directly leveraging native BRep representations while preserving entity-level geometric and topological information is critical for accurate assembly reasoning. By jointly modeling face geometry, edge geometry, and their topological relations, BRepAssemblyLLM enables more accurate and reliable mating-entity prediction.

Rigid Pose Prediction. Table 2 evaluates rigid pose prediction using Chamfer Distance (CD). BRepAssemblyLLM achieves the lowest CD of 0.0321, substantially outperforming all baselines. Compared with B-Pose, JoinABLe, and JoinABLe+Search, it reduces CD by 54.14%, 48.80%, and 44.66%, respectively. These results show that BRep representations provide effective geometric context for pose prediction, while reinforcement learning further improves the accuracy and geo-

Table 1: Top-1 joint connectivity accuracy on the Fusion 360 Gallery assembly benchmark. Higher is better.

Method	Input	Acc. (%) ↑
Human CAD Expert	Image	80.00
Qwen2.5-VL-7B (Bai et al., 2025)	Image + Text	36.00
InternVL3.5-8B (Wang et al., 2025b)	Image + Text	30.00
DeepSeek-VL-7B (Lu et al., 2024)	Image + Text	41.00
LLaVA-7B (Liu et al., 2023)	Image + Text	26.00
ANCSH (Li et al., 2020)	Point Cloud	10.59
Shape2Motion (Wang et al., 2019)	Point Cloud	4.28
PointLLM-7B (Xu et al., 2024)	Point Cloud	45.00
ShapeLLM-7B (Qi et al., 2024)	Point Cloud	51.00
UV-Net (Jayaraman et al., 2021)	BRep	65.21
B-Heuristic (Willis et al., 2022)	BRep	71.39
B-Random (Willis et al., 2022)	BRep	21.55
JoinABLe (Willis et al., 2022)	BRep	79.53
BRepAssemblyLLM	BRep	**89.61**

Table 2: Rigid pose prediction measured by Chamfer Distance(CD) on the Fusion 360 Gallery benchmark. Lower is better.

Method	Input	CD ↓
B-Pose (Zhan et al., 2020)	BRep	0.0700
JoinABLe (Willis et al., 2022)	BRep	0.0627
JoinABLe (Willis et al., 2022) + Search	BRep	0.0580
BRepAssemblyLLM	BRep	**0.0321**

metric validity of the predicted transformations. More importantly, BRepAssemblyLLM addresses the limitation of existing LLMs in directly understanding native BRep representations and predicting executable rigid transformations, enabling direct prediction from native CAD geometry to assembly poses.

4.5 Ablation Studies

Topology-aware BRep encoder. Table 3 compares our topology-aware BRep encoder with UV-Net under identical training settings. Without RL, our encoder improves connectivity accuracy from 85.87% (#4) to 87.48% (#6). Under the complete three-stage curriculum, it further increases accuracy from 86.98% (#5) to 89.61% (#7), while reducing CD from 0.0395 to 0.0321. By jointly modeling face geometry, edge geometry, and face–edge incidence relations, our encoder captures both local geometric cues and native topology, producing more discriminative BRep tokens for mating-entity selection and pose estimation. These results show that our topology-aware BRep encoder achieves clearly better assembly reasoning performance than existing BRep encoders.

Three-stage training curriculum. Table 3 demonstrates the effectiveness of our three-stage training curriculum. Starting from the zero-shot model, SFT dramatically improves connectivity accuracy from 30.00% (#1) to 85.29% (#2), establishing reliable native-entity grounding. Alignment pretraining further improves representation quality, while RL introduces CAD-executor feedback to jointly optimize mating-entity selection and executable pose prediction. With our topology-aware BRep encoder, adding RL on top of alignment and SFT increases accuracy from 87.48% (#6) to 89.61% (#7) and achieves a CD of 0.0321, yielding the best overall performance. The same trend is also observed with UV-Net, where the complete three-stage strategy improves accuracy from 85.29% (#2) to 86.98% (#5) and achieves a CD of 0.0395. These results show that the three stages play complementary roles and together provide substantial gains in executable CAD assembly.

Table 3: Ablation study of BRepAssemblyLLM on the Fusion 360 Gallery benchmark.

ID	Configuration	BRep Encoder	Align.	SFT	RL	Acc. (%) ↑	CD ↓
#1	Zero-shot	UV-Net	–	–	–	30.00	–
#2	SFT only	UV-Net	–	✓	–	85.29	–
#3	SFT + RL	UV-Net	–	✓	✓	86.80	0.0401
#4	Align. + SFT	UV-Net	✓	✓	–	85.87	–
#5	Three-stage training	UV-Net	✓	✓	✓	86.98	0.0395
#6	Align. + SFT	Ours	✓	✓	–	87.48	–
#7	Full model	Ours	✓	✓	✓	**89.61**	**0.0321**

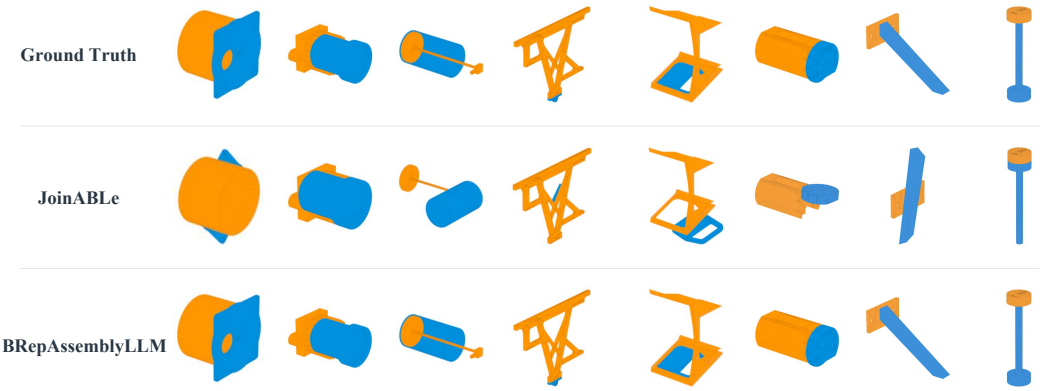


Figure 4: Qualitative comparison of ground truth, BRepAssemblyLLM, and JoinABLe on CAD assemblies from the Fusion 360 Gallery benchmark. Yellow and blue denote the BReps of Part A and Part B, respectively.

4.6 Qualitative Results

Fig. 4 presents qualitative comparisons between BRepAssemblyLLM and JoinABLe on the Fusion 360 Gallery. Across diverse part geometries and mating configurations, the assembled poses predicted by BRepAssemblyLLM closely match the ground truth in both relative orientation and contact position. In particular, our method maintains accurate mating relations and relative poses for challenging cases involving cylindrical surfaces, planar contacts, and slender structures, where precise geometric correspondence is required. In contrast, JoinABLe exhibits several failure cases, including incorrect orientations, misaligned mating regions, and separated parts.

5 Conclusion

This paper formulates CAD assembly as executable BRep reasoning for LLMs and presents BRepAssemblyLLM, a geometry-conditioned LLM framework for BRep models. The framework uses a topology-aware BRep encoder and a lightweight projector to model native geometric structures, and predicts mating-entity pairs together with rigid transformations. The model is trained with a three-stage curriculum comprising BRep–text alignment, supervised entity selection, and CAD-kernel-guided reinforcement learning. On the Fusion 360 Gallery benchmark, BRepAssemblyLLM achieves the best overall performance. The results demonstrate that LLMs can not only interact with CAD systems through CAD views, text, and code, but also directly understand native CAD entities (BReps) and serve as structured geometric reasoning policies that reason over these entities to generate executable assembly actions.

Limitations and future work. The current study focuses on pairwise rigid assembly and has not yet been extended to multi-part assembly or richer joint semantics. Moreover, the current pose reward mainly considers contact and overlap, which may not fully capture CAD design intent when multiple physically plausible assemblies exist. Future work will address more complex cases.

6 AI USE STATEMENT

Generative AI tools were used only for language polishing and visual refinement of figures. They were not used to generate experimental results, perform data analysis, or make scientific conclusions. All AI-assisted content was carefully reviewed by the authors.

7 REPRODUCIBILITY STATEMENT

We provide detailed model, training, and evaluation settings in the paper. All experiments are conducted on the publicly available Fusion 360 Gallery benchmark. Model training is performed using four NVIDIA A800 GPUs, each with 80 GB of memory. We will release the source code, training configurations, and evaluation scripts to facilitate reproducibility.

REFERENCES

Shuai Bai et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Federico Campi, Claudio Favi, Michele Germani, and Marco Mandolini. Cad-integrated design for manufacturing and assembly in mechanical design. *International Journal of computer integrated manufacturing*, 35(3):282–325, 2022.

Yun-Chun Chen, Haoda Li, Dylan Turpin, Alec Jacobson, and Animesh Garg. Neural shape mating: Self-supervised object assembly with adversarial shape priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Liyuan Deng, Hao Guo, Yunpeng Bai, Yongkang Dai, Huaxi Huang, and Yilei Shi. BrepLLM: Native boundary representation understanding with large language models. *arXiv preprint arXiv:2512.16413*, 2025.

Bi'an Du, Xiang Gao, Wei Hu, and Renjie Liao. Generative 3d part assembly via part-whole-hierarchy message passing. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20850–20859. IEEE, 2024.

Yandong Guan, Xilin Wang, Ximing Xing, Jing Zhang, Dong Xu, and Qian Yu. CAD-Coder: Text-to-CAD generation with chain-of-thought and geometric reward. *arXiv preprint arXiv:2505.19713*, 2025.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jialei Huang, Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas Guibas, and Hao Dong. Generative 3d part assembly via dynamic graph learning. In *Advances in Neural Information Processing Systems*, 2020.

Pradeep Kumar Jayaraman, Aditya Sanghi, Joseph G. Lambourne, Karl D. D. Willis, Thomas Davies, Hooman Shayani, and Nigel Morris. UV-Net: Learning from boundary representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Pradeep Kumar Jayaraman, Joseph G Lambourne, Nishkrit Desai, Karl DD Willis, Aditya Sanghi, and Nigel JW Morris. Solidgen: An autoregressive model for direct b-rep synthesis. *arXiv preprint arXiv:2203.13944*, 2022.

Benjamin Jones, Dalton Hildreth, Duowen Chen, Ilya Baran, Vladimir G Kim, and Adriana Schulz. Automate: A dataset and learning approach for automatic mating of cad assemblies. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021.

Benjamin T. Jones, Michael Hu, Milin Kodnongbua, Vladimir G. Kim, and Adriana Schulz. Self-supervised representation learning for CAD. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21327–21336, 2023.

Mingi Kim, Yongjun Kim, Jungwoo Kang, and Hyungki Kim. BrepCoder: A unified multimodal large language model for multi-task B-rep reasoning. *arXiv preprint arXiv:2602.22284*, 2026.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. ABC: A big CAD model dataset for geometric deep learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Joseph G. Lambourne, Karl D. D. Willis, Pradeep Kumar Jayaraman, Aditya Sanghi, Peter Meltzer, and Hooman Shayani. BRepNet: A topological message passing system for solid models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Jiahao Li, Weijian Ma, Xueyang Li, Yunzhong Lou, Guichun Zhou, and Xiangdong Zhou. Cad-llama: leveraging large language models for computer-aided design parametric 3d model generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18563–18573. IEEE, 2025.

Pu Li, Jianwei Guo, Xiaopeng Zhang, and Dong-Ming Yan. Secad-net: Self-supervised cad reconstruction by learning sketch-extrude operations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16816–16826. IEEE, 2023.

Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3706–3715, 2020.

Yichen Li, Kaichun Mo, Yueqi Duan, He Wang, Jiequan Zhang, Lin Shao, Wojciech Matusik, and Leonidas Guibas. Category-level multi-part multi-joint 3d shape assembly. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. Deepseek-vl: Towards real-world vision-language understanding, 2024.

Yongwei Miao, Renato Pajarola, and Jieqing Feng. Curvature-aware adaptive re-sampling for point-sampled geometry. *Computer-Aided Design*, 41(6):395–403, 2009.

Li-Ming Ou and Xun Xu. Relationship matrix based automatic assembly sequence generation from a cad model. *Computer-Aided Design*, 45(7):1053–1067, 2013.

Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *European Conference on Computer Vision*, pp. 214–238. Springer, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021.

Danila Rukhovich, Elona Dupont, Dimitrios Mallis, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada. CAD-Recode: Reverse engineering CAD code from point clouds. *arXiv preprint arXiv:2412.14042*, 2025.

Ari Seff, Yaniv Ovadia, Wenda Zhou, and Ryan P. Adams. SketchGraphs: A large-scale dataset for modeling relational geometry in computer-aided design. *arXiv preprint arXiv:2007.08506*, 2020.

Ari Seff, Wenda Zhou, Nick Richardson, and Ryan P. Adams. Vitruvion: A generative model of parametric CAD sketches. *arXiv preprint arXiv:2109.14124*, 2022.

Yuan Shui, Yandong Guan, Zhanwei Zhang, Juncheng Hu, Jing Zhang, Dong Xu, and Qian Yu. Articad: Articulated cad assembly design via multi-agent code generation, 2026. URL https://arxiv.org/abs/2604.10992.

Garrett Thomas, Melissa Chien, Aviv Tamar, Juan Aparicio Ojea, and Pieter Abbeel. Learning robotic assembly from cad. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3524–3531. IEEE, 2018.

Tetsuo Tomiyama. Intelligent computer-aided design systems: Past 20 years and future 20 years. *Ai Edam*, 21(1):27–29, 2007.

Muhammad Usama, Didier Stricker, Mohammad Sadil Khan, and Muhammad Zeshan Afzal. Brepclip: Contrastive multimodal pretraining on brep primitives for cad understanding. *arXiv preprint arXiv:2606.05515*, 2026.

Siyu Wang, Cailian Chen, Xinyi Le, Qimin Xu, Lei Xu, Yanzhou Zhang, and Jie Yang. CAD-GPT: Synthesising CAD construction sequence with spatial reasoning-enhanced multimodal LLMs. In *AAAI Conference on Artificial Intelligence*, 2025a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8868–8876. IEEE, 2019.

Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023.

Karl D. D. Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Fusion 360 gallery: A dataset and environment for programmatic CAD construction from human design sequences. *ACM Transactions on Graphics*, 40 (4), 2021.

Karl D. D. Willis, Pradeep Kumar Jayaraman, Hang Chu, Yunsheng Tian, Yifei Li, Daniele Grandi, Aditya Sanghi, Linh Tran, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. JoinABLe: Learning bottom-up assembly of parametric CAD joints. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Ruihai Wu, Chenrui Tie, Yushi Du, Yan Zhao, and Hao Dong. Leveraging $SE(3)$ equivariance for learning 3d geometric shape assembly. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Rundi Wu, Chang Xiao, and Changxi Zheng. DeepCAD: A deep generative network for computer-aided design models. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851. IEEE, 2024.

Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pp. 131–147. Springer, 2024.

Xiang Xu, Karl D. D. Willis, Joseph G. Lambourne, Chin-Yi Cheng, Pradeep Kumar Jayaraman, and Yasutaka Furukawa. SkexGen: Autoregressive generation of CAD construction sequences with disentangled codebooks. In *International Conference on Machine Learning*, 2022.

Mohsen Yavartanoo, Sangmin Hong, Reyhaneh Neshatavar, and Kyoung Mu Lee. Text2CAD: Text to 3d CAD generation via technical drawings. *arXiv preprint arXiv:2411.06206*, 2024.

Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas Guibas, Hao Dong, et al. Generative 3d part assembly via dynamic graph learning. *Advances in Neural Information Processing Systems*, 33:6315–6326, 2020.

Zhanwei Zhang, Minghao Chen, Zhihong Gu, Xinkui Zhao, Zheng Yang, Binbin Lin, Deng Cai, and Wenxiao Wang. Straj: Self-training for bridging the cross-geography gap in trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22723–22731, 2025a.

Zhanwei Zhang, Shizhao Sun, Wenxiao Wang, Deng Cai, and Jiang Bian. Flexcad: Unified and versatile controllable cad generation with fine-tuned large language models. In *International Conference on Learning Representations*, volume 2025, pp. 3204–3227, 2025b.

Zhanwei Zhang, Junjie Liu, Wenxiao Wang, Binbin Lin, Liang Xie, Chen Shen, Deng Cai, et al. Geocad: Local geometry-controllable cad generation with large language models. *Advances in Neural Information Processing Systems*, 38:130989–131014, 2026.