MUSE: BENCHMARKING ASSEMBLY STRUCTURE AND INTERFACE GEOMETRY IN MULTI-COMPONENT TEXT-TO-CAD GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have recently advanced text-driven 3D generation, yet generating CAD models that satisfy practical design requirements remains challenging. Existing benchmarks focus primarily on single-part CAD models and geometric similarity, providing limited assessment of assembly structure and interface geometry. To address this gap, we introduce MUSE, a Text-to-CAD benchmark for generating multi-component, editable boundary representation (B-Rep) assemblies. MUSE pairs practical design instances with structured *Design Specifications* and evaluates generated outputs across four dimensions: code executability, geometric validity, structural and interface fidelity, and functional consistency. We further introduce Structure and Interface Evaluation (SIE), which combines reference Physical Assembly Graphs with direct B-Rep analysis to assess component correspondence, connectivity, interface types, and interface clearances. Experiments on proprietary and open-weight LLMs show that high execution success does not necessarily translate into correct assembly structure or interface geometry. Together, MUSE provides a benchmark and evaluation framework for studying multi-component Text-to-CAD generation beyond shape similarity. Our dataset and code are available at `https://anonymous.4open.science/status/muse_-8C3B`.

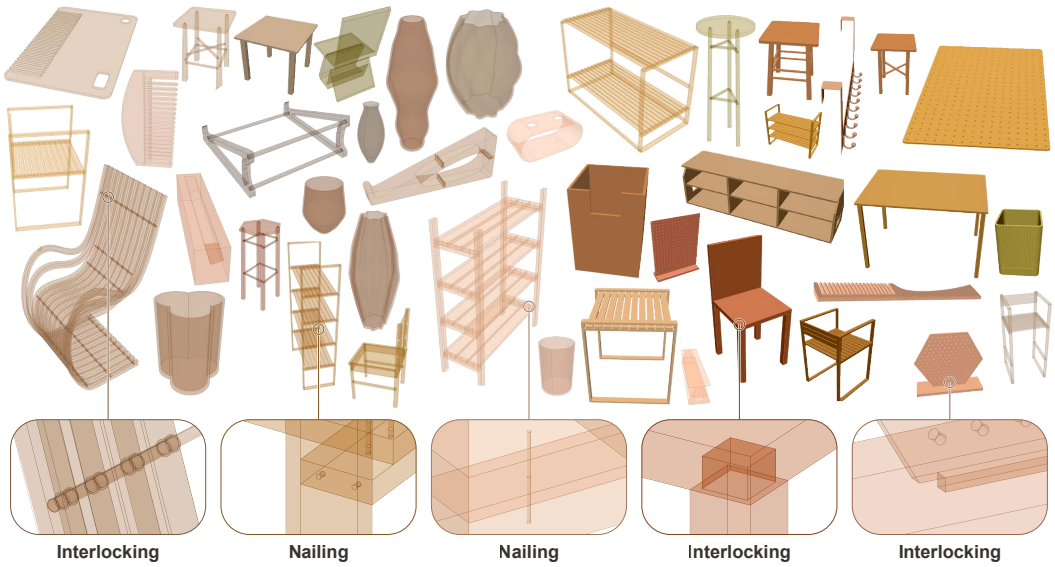


Figure 1: Examples of multi-component designs in MUSE, illustrating explicit assembly structure and interface geometry.

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.

1 INTRODUCTION

Text-to-CAD, the task of generating computer-aided design (CAD) models from natural-language descriptions, is emerging as a promising direction for AI-assisted 3D modeling Khan et al. (2024a). Beyond visual plausibility, Text-to-CAD aims to produce **design instances** grounded in practical use and engineering requirements. For multi-component designs, valid individual geometries alone are insufficient: the components must also connect and fit together as intended. This shifts the focus toward engineering usability, requiring the generated model to capture the intended shape while satisfying usage-level constraints.

Although Text-to-CAD datasets Khan et al. (2024a) have grown substantially in recent years, many still focus on single-part CAD models, providing limited coverage of the interdependent constraints in practical multi-component designs. Such designs require not only valid part geometry, but also correct connectivity, interface types, and interface clearances. Constructing design instances that explicitly capture these requirements demands substantial input from experienced designers, making them costly to curate.

Beyond dataset complexity, evaluation is another key challenge for Text-to-CAD. Existing evaluation methods can be broadly grouped into three categories: *geometric similarity*, *visual alignment*, and *programmatic STEP checks*. Geometric similarity metrics, such as Chamfer Distance (CD) Wang et al. (2025), measure shape resemblance but may overlook errors in connectivity, interface types, and interface clearances in multi-component designs. For example, a functionally valid chair may receive a poor CD score, while a geometrically similar one may fail to satisfy these requirements (Figure 2). Visual alignment evaluates rendered CAD models against textual or visual references using VLMs Shui et al. (2026). However, it requires additional model inference and may produce unreliable judgments, particularly when details such as interface types and interface clearances are obscured in rendered views. Programmatic STEP checks, as used in CADTestBench (Mallis et al., 2026a), directly inspect generated STEP models but remain focused on single-part designs.

To address these limitations, we propose MUSE, a benchmark for practical Text-to-CAD generation emphasizing **assembly structure** and **interface geometry** in complex multi-component designs. Here, assembly structure denotes component configuration, while interface geometry denotes the geometric features through which components connect and fit. MUSE is constructed through collaboration between human designers and LLMs. Each design instance is defined by a structured *Design Specification* that describes the intended use, manufacturing methods, admissible parameter ranges, and requirements for assembly structure and interface geometry. We further introduce an attributed graph representation, the Assembly Graph, that explicitly records components, connectivity, interface types, and interface clearances. Each design instance is also paired with program-generated CAD drawings for visual review. All design instances are reviewed by human designers.

Our evaluation framework assesses generated CAD models along five complementary dimensions: code executability, geometric validity, structural and interface fidelity, and functional consistency. To evaluate assembly structure and interface geometry, we introduce **Structure and Interface Evaluation (SIE)**, which uses programmatic STEP checks guided by the reference Assembly Graph. SIE evaluates four aspects: Structure Correspondence, connectivity, interface types, and interface clearances. It first matches generated components to the design reference, then evaluates the remaining three aspects on the matched components. By grounding these assessments in direct B-Rep measurements, SIE provides explicit geometric evidence for the fidelity of assembly structure and interface geometry. For these dimensions, its deterministic programmatic STEP checks avoid the inference latency and token costs associated with VLM-based visual alignment. Functional consistency, which is less amenable to direct geometric verification, is evaluated separately using a VLM-as-judge protocol.

Experiments on MUSE reveal that both proprietary and open-weight LLMs remain far from reliable end-to-end CAD generation. Even the strongest evaluated model, GPT-6 Astra, achieves only 57.55% joint success, with substantial residual errors in geometric validity, component recovery, assembly connectivity, and interface typing. Our analysis further shows that successful code execution is not equivalent to correct CAD modeling, and that correctly identified interface types do not necessarily imply accurate interface clearances. Together, these results identify geometric validity, assembly structure, interface correctness, and clearance accuracy as key remaining bottlenecks for robust multi-component Text-to-CAD generation.

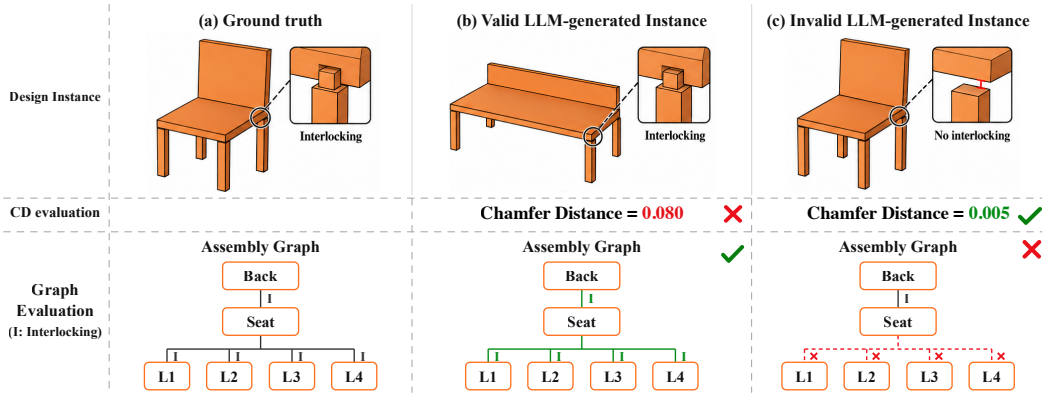


Figure 2: Chamfer Distance can misjudge multi-component CAD designs, while SIE evaluates assembly structure and interface geometry through programmatic STEP checks.

Table 1: Comparison of Representative Existing Text-to-CAD Benchmarks.

Benchmark	Input Type	Dataset Complexity	Evaluation Focus
Text2CAD (Khan et al., 2024c)	Modeling sequence	Single-part	Geometric similarity
CADPrompt (Alrashedy et al., 2024)	Text description	Single-part	Geometric similarity
Text2CAD-Bench (Wang et al., 2026)	Geo. / Seq. description	Single-part	Geometric similarity
CADTestBench (Mallis et al., 2026a)	Text requirements	Single-part	Programmatic STEP Checks
ArtiCAD (Shui et al., 2026)	Text / image description	Multi-part assembly	Visual Alignment
MUSE (Ours)	**Design description & assembly graph**	**Multi-part assembly**	**Programmatic STEP Checks**

2 Related Work

Text-to-CAD Datasets. Text-to-CAD translates natural language into editable CAD models through command-sequence prediction (Khan et al., 2024b; Li et al., 2024) or executable code generation (Shui et al., 2026). Existing datasets commonly augment CAD construction sequences with textual descriptions. For example, Text2CAD enriches DeepCAD with automatically generated descriptions at different levels of detail (Wu et al., 2021; Khan et al., 2024b), while CADFusion incorporates visual feedback to improve text-to-shape alignment (Wang et al., 2025). As summarized in Table 1, these benchmarks largely emphasize single-part geometry, with evaluation centered on geometric similarity and visual alignment (Khan et al., 2024b; Wang et al., 2025).

Although recent work such as ArtiCAD (Shui et al., 2026) explores assembly-aware generation, shape similarity alone does not establish whether components are correctly separated, connected, or dimensioned at mating interfaces. A visually similar model may fuse distinct components or introduce incorrect interface clearances. In contrast, MUSE pairs multi-component B-Rep designs with structured Design Specifications and evaluates component correspondence, connectivity, interface types, and interface clearances through direct geometric analysis. These assessments are complemented by execution, geometric-validity, and VLM-based functional checks.

Text-to-CAD Evaluation. Existing Text-to-CAD evaluation mainly relies on geometric similarity, visual alignment, and programmatic STEP checks. Geometric similarity metrics such as Chamfer Distance compare generated and reference shapes Wu et al. (2021); Khan et al. (2024b), while recent work also considers topology-aware measures Preintner et al. (2025). However, these metrics do not explicitly evaluate assembly structure and interface geometry. Visual alignment using VLMs has also been adopted for CAD verification and refinement Khan et al. (2024b); Alrashedy et al. (2024); Barkley et al. (2026), but is limited by occlusion and viewpoint coverage, especially for internal interface geometry and small interface clearances. More recently, executable tests directly on CAD geometry have emerged as a deterministic alternative Mallis et al. (2026b). Our evaluation also operates directly on CAD geometry, but targets a different aspect: component correspondence, typed interface connectivity, and local interface clearances in multi-component CAD designs.

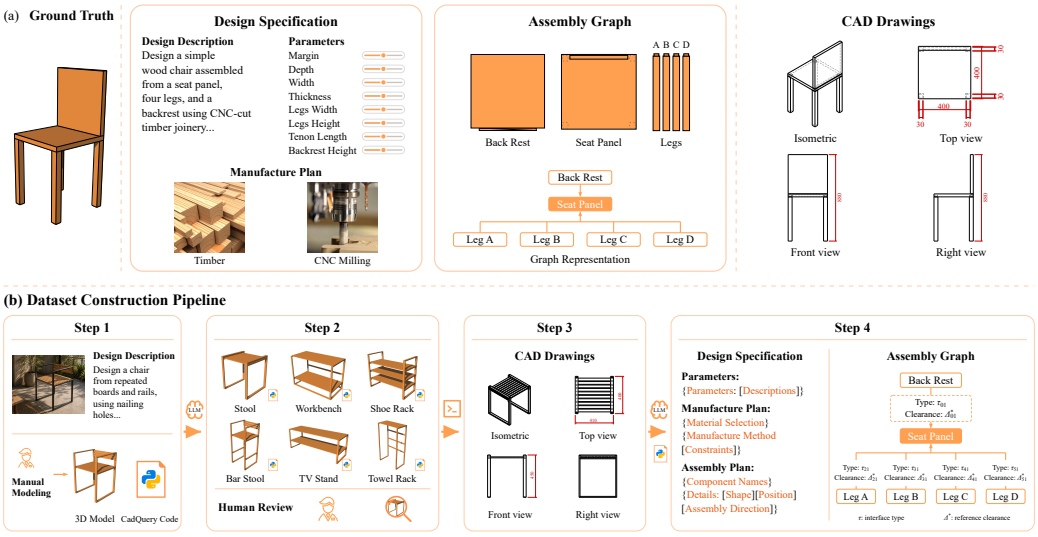


Figure 3: Overview of our dataset construction. (a) Each benchmark instance provides a Design Specification, assembly graph and CAD drawings. (b) The dataset is curated through expert seed modeling, LLM-based augmentation, CAD drawing generation, and Design Specification synthesis.

3 MUSE: Benchmark Design

3.1 Preliminaries: Design Description and Assembly Graph

Existing Text-to-CAD benchmarks primarily use text descriptions or images as inputs, as summarized in Table 1, which do not explicitly represent assembly structure and interface geometry. We therefore introduce a Physical Assembly Graph to make this information explicit for multi-component CAD designs. Each benchmark instance provides two inputs, organized as a structured *Design Specification*:

$$\mathcal{S} = \langle \mathcal{D}, \mathcal{G} \rangle, \tag{1}$$

where $\mathcal{D}$ is a natural-language design description and $\mathcal{G}$ is a Physical Assembly Graph. The description $\mathcal{D}$ specifies the intended use, manufacturing methods, admissible parameter ranges. The graph $\mathcal{G}$ explicitly represents assembly structure and interface geometry.

Definition 1 (Physical Assembly Graph)**.** *A Physical Assembly Graph is an attributed undirected graph:*

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}), \qquad \mathbf{a}(v_i) = \ell_i, \qquad \mathbf{a}(e_{ij}) = \langle \tau_{ij}, \Delta^*_{ij} \rangle, \tag{2}$$

where each node $v_i \in \mathcal{V}$ represents a physical component with a semantic label ℓ_i, such as seat panel *or* support leg*. Each edge $e_{ij} \in \mathcal{E}$ represents a physical interface between components v_i and v_j, with interface type τ_{ij} and, where applicable, a reference interface clearance Δ^*_{ij}. Each reference interface clearance Δ^*_{ij} is accompanied by an allowable range $[L_{ij}, U_{ij}]$, which specifies the lower and upper bounds for the generated interface clearance.*

3.2 Dataset Construction Pipeline

We introduce a dataset of strictly functional, manufacturable and assemblable CAD models. We employ a "Human-in-the-loop" pipeline that synergizes expert domain knowledge with LLM-driven scalable augmentation. The dataset curation pipeline consists of four stages, as illustrated in Figure 3. We first create high-quality seed 3D models with expert designers, then expand them into diverse 3D model variants using LLM-assisted CAD script augmentation. The resulting models are converted into standardized engineering views, and finally paired with structured Design Specifications.

Step 1: Expert-Guided Construction of Seed Models. We first collect the design description and rendered reference examples of established designs across diverse design categories (e.g., chairs, tables, and business card holders). Following these instructions, professional designers manually construct high-quality 3D models in STEP format as the seed models. We then convert the designers' modeling procedures into executable *CadQuery* scripts, which serve as the basis for subsequent augmentation.

Step 2: LLM-Powered Data Augmentation. To maximize dataset diversity, we use Claude Opus 4.7 to systematically expand the initial *CadQuery* scripts along two primary dimensions: *stylistic transformations* (e.g., adapting a traditional wooden chair into modern industrial aesthetics) and *functional repurposing* (e.g., transforming a chair's into a shoe rack or coat stand).

Step 3: Engineering View and Physical Assembly Graph Generation. For each seed and augmented model, we generate engineering views and a Physical Assembly Graph $\mathcal{G}$. To mitigate occlusion and limited viewpoint coverage in conventional rendered images Qi et al. (2026), we use the geometry kernel to extract boundaries, contours, and hidden lines, with hidden lines shown as dashed paths. The views are arranged in a standardized 2×2 grid (top, front, right, and isometric) for visual inspection (Figure 3(a)). We also provide each STEP model to an LLM to generate $\mathcal{G}$, recording components, connectivity, interface types, and reference interface clearances with their allowable ranges.

Step 4: Design Specification Synthesis and Expert Review. Using the engineering views, *CadQuery* scripts, and Physical Assembly Graphs, we prompt GPT-5.5 with expert-written examples (Appendix C) to generate design descriptions $\mathcal{D}$ covering intended use, manufacturing methods, admissible parameter ranges, and interface clearance requirements. Each description is paired with $\mathcal{G}$ to form $\mathcal{S} = \langle \mathcal{D}, \mathcal{G} \rangle$. Human experts review and correct both $\mathcal{D}$ and $\mathcal{G}$ for consistency with the CAD model and design requirements.

3.3 Evaluation Protocol

Given a Design Specification $\mathcal{S} = \langle \mathcal{D}, \mathcal{G} \rangle$, each model generates a *CadQuery* program, which is executed to produce a STEP model C and standardized engineering drawings. We evaluate the outputs through five checks: code, geometry, structure, interface, and function, as illustrated in Figure 4 and summarized in Table 2. The code check assesses program executability; the geometry, structure, and interface checks operate on the STEP model; and the function check uses a VLM to assess the engineering drawings.

Code Check. A sample passes **Sandbox Success** if its *CadQuery* program terminates without execution errors, constructs a non-empty CAD model, and successfully exports a STEP file. This check measures whether the model produces an executable CAD artifact.

Geometry Check. For exported STEP models, we evaluate four independent binary criteria. *Watertight* requires closed solids without open boundaries; *Manifold* requires valid manifold topology; *Self-Intersection Free* excludes self-intersecting faces or invalid internal intersections within individual solids; and *Overlap Free* excludes physical interpenetration between distinct components. *Geom. Valid* requires successful execution and passes on all four checks.

Structure Check. Geometric validity alone does not establish whether a model contains the intended physical components. Let the reference Physical Assembly Graph be $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$. SIE associates generated solids with labeled reference components using normalized relative-position, dimension, and face-count descriptors. An augmented Hungarian assignment permits unmatched components; mutual-nearest and ambiguity-margin checks reject uncertain correspondences. The full procedure appears in Algorithm 1 in Appendix A. We compare the resulting labeled node set $\widehat{\mathcal{V}}$ with $\mathcal{V}^*$ using *Node Precision*, *Node Recall*, and *Node F1*. A correct node requires both component presence and correct semantic correspondence.

Interface Check. SIE further evaluates connectivity, interface types, and interface clearances through the following three checks.

(1) Connectivity and interface type checks. We detect connections between solids and identify their interface types, e.g., interlocking (Algorithm 2 in Appendix A). As an example of interlocking detection, rectangular mortise-and-tenon interfaces are identified from nearby, oppositely oriented

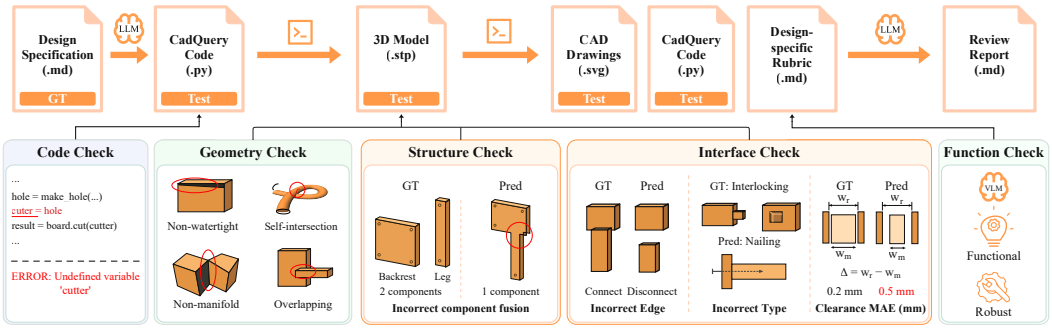


Figure 4: Illustration of the proposed evaluation pipeline. Given a Design Specification, each evaluated model generates a *CadQuery* script, which is executed and exported as a STEP file. The evaluation then performs code, geometry, structure, interface, and function checks.

planar faces with overlapping projections. Four-wall candidates are verified using orthogonality, common-section, and solid-containment checks, and retained when all four male walls match inward-facing receiver walls. For connectivity, the detected component pairs define the predicted edge set $\widehat{\mathcal{E}}$, which is compared with the reference edge set $\mathcal{E}^*$ using *Edge F1*. For interface type checks, we evaluate whether the detected connections have the correct interface types using *Type F1*.

(2) interface clearances. For each localized rectangular interface e, we recover paired male and receiver side faces from the STEP geometry. Their single-side clearance is

$$g_{ei} = (\mathbf{c}^r_{ei} - \mathbf{c}^m_{ei})^\top \mathbf{n}^m_{ei}, \tag{3}$$

where $\mathbf{c}^m_{ei}$ and $\mathbf{c}^r_{ei}$ are face centroids and $\mathbf{n}^m_{ei}$ is the male face's outward normal. The four side clearances form two opposing pairs, giving total clearances along two cross-sectional directions:

$$\Delta_{e1} = g_{e1} + g_{e2}, \qquad \Delta_{e2} = g_{e3} + g_{e4}. \tag{4}$$

Equivalently, $\Delta_{ek} = w^r_{ek} - w^m_{ek}$, where $k \in \{1, 2\}$ indexes the cross-sectional directions, and w^r_{ek} and w^m_{ek} are receiver and male widths. Agreement between these calculations provides an internal consistency check. We measure deviation from reference total clearances Δ^*_{ek} using

$$\text{ClearanceMAE} = \frac{1}{N_\Delta} \sum_{(e,k)\in\mathcal{Q}} \left|\Delta_{ek} - \Delta^*_{ek}\right|, \tag{5}$$

where $\mathcal{Q}$ contains interface–direction pairs with both predicted and reference measurements, and $N_\Delta = |\mathcal{Q}|$. Clearance MAE is reported in millimeters and measures reference-clearance deviation, not tolerance compliance.[1] Compliance with the specified allowable range $[L_{ek}, U_{ek}]$ requires $L_{ek} \le \Delta_{ek} \le U_{ek}$.

Function Check. Even a geometrically valid model with correct component structure may fail to satisfy its intended function. A VLM therefore evaluates the generated SVG engineering views together with the Design Specification $\mathcal{S}$. **Functional** assesses whether the design contains the structures needed for its intended use. **Robust** assesses whether the visible component arrangement, supports, interfaces, and apparent load-transfer structure suggest a mechanically plausible and stable design. These criteria are scored independently and represent visual assessments rather than physical validation. The evaluation prompt and scoring protocol are provided in Appendix C.

4 Experiments

We first provide an overview of MUSE. We then use our five-stage evaluation pipeline to assess the CAD models generated by proprietary and open-source LLMs.

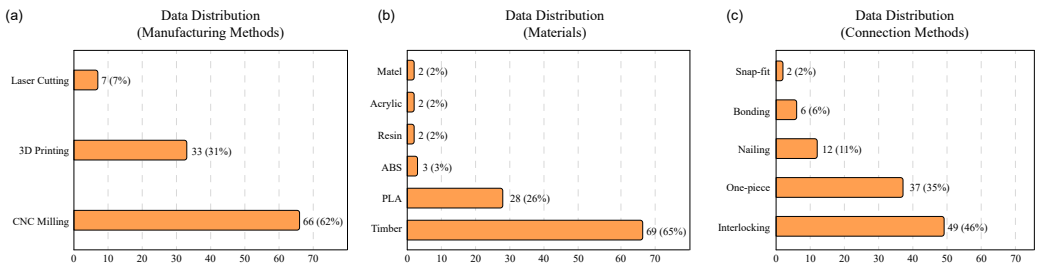


Figure 5: Dataset statistics of MUSE, showing the distributions of (a) manufacturing methods, (b) materials, and (c) connection methods across all design instances.

Table 2: Performance across five evaluation checks. Bold and italics indicate the best and second-best available values, respectively, including ties.

	Code Check	Geometry Check					Structure Check	Interface Check			Function Check		Overall
Model	Sandbox Success ↑	Watertight ↑	Manifold ↑	Self-Int. Free ↑	Overlap Free ↑	Geom. Valid ↑	Node F1 ↑	Edge F1 ↑	Type F1 ↑	Clearance MAE↓	Functional ↑	Robust ↑	Joint Success ↑
Closed-Source Models													
gpt-6-astra	**99.06**	**97.17**	**97.17**	**89.62**	**97.17**	**89.62**	**89.77**	**86.08**	**74.22**	0.3341	**86.79**	**84.91**	**57.55**
claude-opus-4.7	76.42	*76.42*	*76.42*	*73.58*	60.38	58.49	63.77	48.39	33.02	0.3082	43.40	42.45	23.58
claude-3.7-sonnet	46.23	46.23	46.23	46.23	23.58	23.58	30.18	11.96	8.20	*0.2353*	11.32	13.21	8.49
gemini-3.1-pro	65.09	59.43	63.21	63.21	62.26	56.60	53.52	53.66	41.24	0.3066	47.17	45.28	22.64
gpt-5.5	*77.36*	72.64	74.53	*73.58*	*70.75*	*66.04*	*69.05*	*59.19*	*48.64*	0.4750	*51.89*	*52.83*	*34.91*
gpt-4o	36.79	36.79	36.79	36.79	13.21	13.21	17.78	4.58	1.41	–	1.89	5.66	1.89
Open-Source Models													
glm-5.1	31.13	31.13	31.13	31.13	27.36	27.36	29.12	17.93	14.64	0.3641	17.92	19.81	15.09
glm-4.7-flash	8.49	7.55	7.55	7.55	5.66	4.72	5.25	0.00	0.00	–	1.89	1.89	0.94
minimax-m2.7	18.87	18.87	18.87	18.87	10.38	10.38	13.26	2.22	2.22	**0.2000**	5.66	4.72	4.72
minimax-m2.5	25.47	22.64	22.64	22.64	10.38	9.43	11.23	1.84	0.21	–	1.89	1.89	1.89
qwen-3.5-122b-a10b	27.36	25.47	25.47	25.47	13.21	11.32	15.56	2.42	2.42	–	6.60	8.49	4.72
qwen-2.5-72b	50.00	49.06	49.06	49.06	19.81	19.81	20.58	2.61	0.00	–	1.89	3.77	0.94
llama-3.1-70b	31.13	30.19	30.19	30.19	22.64	22.64	8.78	0.00	0.00	–	0.00	1.89	0.00
qwen-3.6-35b-a3b	8.49	8.49	8.49	8.49	5.66	5.66	6.57	0.18	0.00	–	2.83	4.72	2.83
qwen-3.6-coder-next	14.15	14.15	14.15	14.15	10.38	10.38	9.37	0.00	0.00	–	2.83	4.72	1.89
llama-3.1-8b	2.83	1.89	1.89	1.89	0.94	0.94	0.63	0.00	0.00	–	0.00	0.00	0.00

4.1 Data Distribution of MUSE

MUSE comprises 106 design instances spanning a wide range of manufacturing processes, materials, and connection methods, as summarized in Figure 5. The benchmark includes representative manufacturing processes such as computer numerical control (CNC) milling, 3D printing, and laser cutting, with CNC milling constituting the largest category. In terms of material usage, timber and polylactic acid filaments (PLA) are the most prevalent. Regarding connection methods, one-piece designs represent single-body objects, whereas interlocking, nailing, and snap-fit designs correspond to multi-component assemblies. Overall, these distributions indicate that MUSE places particular emphasis on assemblable CAD models while maintaining broad coverage of real-world manufacturable design scenarios.

4.2 Evaluation of Closed-Source and Open-Source LLMs

We evaluate a broad set of closed-source and open-source LLMs using the proposed five-stage evaluation pipeline. Given a Design Specification $\mathcal{S}$, each model generates a *CadQuery* script, evaluated through (1) code, (2) geometry, (3) structure, (4) interface, and (5) function checks. The code check assesses program executability; the geometry, structure, and interface checks assess the resulting STEP model; and the function check evaluates standardized engineering drawings using a VLM. The structure and interface checks jointly constitute SIE, assessing assembly structure and interface geometry. The closed-source models evaluated include GPT-6 Astra[2], GPT-5.5, GPT-

[1]Clearance MAE evaluates only matched, measurable interface clearances, excluding unavailable measurements rather than assigning them zero.

[2]GPT-6 Astra is accessed programmatically through Codex, whereas all other models are accessed through OpenRouter. All models receive the same task inputs under a single-pass generation setting.

4o, Claude Opus 4.7, Claude 3.7 Sonnet, and Gemini 3.1 Pro. The open-source models include GLM-5.1, GLM-4.7-Flash, MiniMax-M2.7, MiniMax-M2.5, Qwen3.5-122B-A10B, Qwen2.5-72B, Llama-3.1-70B, Qwen3.6-35B-A3B, Qwen3.6-Coder, and Llama-3.1-8B.

Inapplicability of Existing Text-to-CAD Baselines. We do not directly evaluate existing Text-to-CAD methods, as their task settings differ fundamentally from ours and are not applicable to our benchmark. Prior work primarily uses command-sequence-style inputs to generate isolated primitive or single-part CAD models (Xu et al., 2024; Liao et al., 2025; Yavartanoo et al., 2024; Alrashedy et al., 2024; Qi et al., 2026). These methods are architecturally and procedurally tailored to such data and interaction paradigms, and thus cannot be directly applied to our design instances, which are manufacturable multi-component assemblies with explicit component relations and manufacturing constraints. ArtiCAD (Shui et al., 2026) addresses assembly-level CAD generation, but we exclude it from our experiments because its implementation was unavailable at the time of evaluation.

4.3 Experimental Results and Analysis

Table 2 summarizes performance across the five checks defined in our evaluation protocol. For cases that fail code execution, the per-case precision, recall, and F1 scores for components, connectivity, and interface types are set to zero. These scores are averaged over all 106 cases. We additionally report **Joint Success**: a case receives a score of one only if it passes all five checks under the criteria defined in the evaluation protocol, and zero otherwise. Table 5 provides a more detailed breakdown of assembly-graph evaluation, reporting precision and recall for components, connectivity, and interface types. The STEP column counts cases with available STEP outputs. We also report *Graph Edit Distance (GED)*, the minimum node and edge edit cost required to transform the predicted assembly graph into the reference graph.

RQ1: What are the main bottlenecks? Geometry, assembly structure, and interface correctness remain major bottlenecks, with substantial performance gaps relative to code execution success (Table 2). For example, Claude Opus 4.7 achieves 76.42% execution success, but only 58.49% geometric validity, 63.77% Node F1, and 33.02% Type F1. The gaps are even larger for Qwen-2.5-72B: despite 50.00% execution success, it reaches only 19.81% geometric validity, 20.58% Node F1, and 0.00% Type F1. **Even GPT-6 Astra still falls short.** Although it leads these metrics, its 99.06% execution success is accompanied by 89.62% geometric validity, 89.77% Node F1, 86.08% Edge F1, and 74.22% Type F1, leaving room for improvement in component recovery, connectivity, and interface typing.

RQ2: Does strong code generation guarantee correct CAD models? High execution success does not guarantee end-to-end CAD correctness. GPT-6 Astra executes successfully on 99.06% of designs, yet achieves only 57.55% joint success, meaning that only slightly more than half of all designs satisfy the combined evaluation requirements. The same gap appears in other models: GPT-5.5 achieves 77.36% execution success but 34.91% joint success, while Claude Opus 4.7 achieves 76.42% and 23.58%, respectively. Thus, producing executable code remains distinct from generating models that jointly satisfy geometric, structural, interface, and functional requirements.

RQ3: Are correctly typed interfaces properly fitted? Clearance MAE captures dimensional interface clearance errors that interface type scores alone cannot reveal, and the two metrics do not consistently rank models in the same order. For example, GPT-5.5 achieves the second-highest Type F1 (48.64%) but the largest reported clearance MAE (0.4750 mm). This illustrates that correctly typed interfaces may still fit poorly for practical use. Conversely, MiniMax-M2.7 has the lowest clearance MAE (0.2000 mm). Since this metric is computed only on detected, measurable connections, its low error may be attributed to the simpler interface types it generates, which are easier to fit.

Local Geometric Feature Analysis. We analyze local geometric features to characterize how LLM-generated CAD models deviate from reference designs, considering both feature counts (holes, fillets, pockets, and slots) and feature dimensions (diameters, depths, and widths). Definitions and computation details are provided in Appendix D. For each eligible design, we normalize each generated descriptor by its ground-truth counterpart. Figures 6 and 7 summarize these ratios, with dots indicating medians and light bars showing the 25th–75th percentile ranges. A ratio of 1 indicates numerical agreement with the reference, while ratios below or above 1 indicate smaller or larger

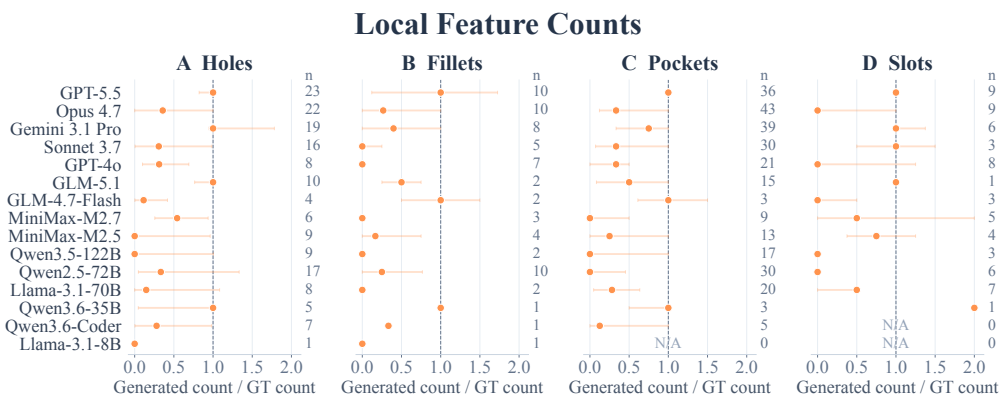


Figure 6: **Local feature counts.** Dots and light bars show the median and 25th–75th percentiles of per-design generated-to-GT count ratios. n denotes eligible designs. Details are in Appendix D.

Table 3: VLM–human agreement on Functional and Robust scores.

	Gemini 3.1 Pro			GPT-5.5			GPT-4o		
Level	Pearson r	Spearman ρ	Kendall τ	Pearson r	Spearman ρ	Kendall τ	Pearson r	Spearman ρ	Kendall τ
Sub-criteria	0.718	0.699	0.648	0.653	0.661	0.601	0.626	0.603	0.572
Criteria	0.755	0.756	0.674	0.719	0.698	0.632	0.696	0.692	0.604
Design instance	0.842	0.824	0.701	0.768	0.781	0.647	0.819	0.807	0.670

values, respectively. Most models have median hole-count ratios below 1, indicating a tendency to generate fewer holes than the references. Agreement in feature counts, however, does not necessarily extend to feature dimensions, and dimensional agreement can vary across descriptors. For GPT-5.5, Opus, and Gemini, hole-diameter ratios concentrate near 1, whereas the interquartile ranges of hole-depth ratios extend toward lower values. GPT-5.5 illustrates this distinction particularly clearly: its median count ratio is 1 across all four feature categories, yet it produces shallower holes in some designs despite close agreement in hole diameters.

4.4 Human Alignment of the VLM Judge

We assess VLM–human agreement for the function check using rendered outputs from 20 sampled design instances, annotated by four raters across two rounds. Human scores are obtained by averaging available annotations for the Functional and Robust sub-criteria. We compare these scores with VLM judgments using Pearson, Spearman, and Kendall correlations.

5 Conclusion

We introduced MUSE, a benchmark for multi-component Text-to-CAD generation with explicit structural and fit requirements. Its Structure and Interface Evaluation (SIE) combines reference assembly graphs with direct B-Rep analysis to assess Structure Correspondence, connectivity, interface types, and interface clearances, while functional consistency is evaluated separately using a VLM. Our results show that generating executable CAD models does not guarantee correct assembly structure or interface geometry, highlighting the need for evaluation beyond global shape similarity.

Limitations. Our evaluation focuses on static assembly structure and interface geometry, but does not evaluate the feasibility of assembly sequences. Such evaluation requires reasoning about component motion, collision-free insertion paths, and intermediate assembly states, which are beyond the scope of our current geometric checks. Moreover, the reference designs have not been physically manufactured or assembled, and VLM-based functional assessments cannot verify real-world mechanical performance.

AI USE STATEMENT

We used generative AI tools for synthetic dataset generation and benchmark evaluation. Claude Opus 4.7 assisted in augmenting expert-written CadQuery scripts to create design variants. An LLM analyzed STEP models to generate draft Physical Assembly Graphs, and GPT-5.5 generated natural-language design descriptions. Human experts reviewed and corrected both the descriptions and the graphs comprising the final Design Specifications. We also used a VLM to evaluate functional consistency from standardized engineering drawings, as described in our evaluation protocol. Additionally, generative AI assisted with manuscript editing, terminology standardization, and LaTeX formatting.

ETHICS STATEMENT

This work develops a benchmark for evaluating multi-component CAD generation. Its evaluation results should not be interpreted as certification of manufacturing safety or physical assembly feasibility. In particular, agreement in assembly structure and interface geometry does not establish load-bearing capacity, material suitability, or safe performance in real-world use. Generated designs require application-specific engineering validation before fabrication or deployment. LLM-generated annotations and VLM-based assessments may also contain errors; we describe expert review of the Design Specifications and evaluate VLM–human agreement to characterize these limitations.

REPRODUCIBILITY STATEMENT

The dataset construction process is described in Section 3.2, with prompting examples provided in Appendix C. Section 3.3 defines the evaluation protocol and metrics, while Appendix A provides detailed notation and implementation procedures. Together, these descriptions document the construction of Design Specifications and the evaluation of generated CAD models, including assembly structure and interface geometry.

REFERENCES

Kamel Alrashedy, Pradyumna Tambwekar, Zulfiqar Zaidi, Megan Langwasser, Wei Xu, and Matthew Gombolay. Generating CAD code with vision-language models for 3D designs. *arXiv preprint arXiv:2410.05340*, 2024.

Jesse Barkley, Rumi Loghmani, and Amir Barati Farimani. Cadsmith: Multi-agent cad generation with programmatic geometric validation. *arXiv preprint arXiv:2603.26512*, 2026.

Mohammad S Khan, Sankalp Sinha, Talha U Sheikh, Didier Stricker, Sk A Ali, and Muhammad Z Afzal. Text2CAD: generating sequential cad designs from beginner-to-expert level text prompts. *Advances in Neural Information Processing Systems*, 37:7552–7579, 2024a.

Mohammad Sadil Khan, Sankalp Sinha, Talha Uddin Sheikh, Didier Stricker, Sk Aziz Ali, and Muhammad Zeshan Afzal. Text2CAD: generating sequential CAD models from beginner-to-expert level text prompts. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 7552–7579, 2024b.

Mohammad Sadil Khan, Sankalp Sinha, Talha Uddin Sheikh, Didier Stricker, Sk Aziz Ali, and Muhammad Zeshan Afzal. Text2cad: Generating sequential CAD designs from beginner-to-expert level text prompts. In *Advances in Neural Information Processing Systems*, volume 37, pp. 7552–7579. Curran Associates, Inc., 2024c.

Xueyang Li, Yu Song, Yunzhong Lou, and Xiangdong Zhou. CAD translator: an effective drive for text to 3D parametric computer-aided design generative modeling. In *Proceedings of the ACM International Conference on Multimedia (ACM MM 2024), Poster*, 2024.

Jianxing Liao, Junyan Xu, Yatao Sun, Maowen Tang, Sicheng He, Jingxian Liao, Shui Yu, Yun Li, and Xiaohong Guan. Automated CAD modeling sequence generation from text descriptions via transformer-based large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21720–21748, 2025.

Dimitrios Mallis, Marco Wang, Ahmet Serdar Karadeniz, Elisa Ricci, Anis Kacem, and Djamila Aouada. Text-to-cad evaluation with cadtests. *arXiv preprint arXiv:2605.07807*, 2026a.

Dimitrios Mallis, Marco Wang, Ahmet Serdar Karadeniz, Elisa Ricci, Anis Kacem, and Djamila Aouada. Text-to-cad evaluation with cadtests. *arXiv preprint arXiv:2605.07807*, 2026b.

Tobias Preintner, Weixuan Yuan, Adrian König, Thomas Bäck, Elena Raponi, and Niki Van Stein. EvoCAD: evolutionary CAD code generation with vision language models. In *2025 IEEE 37th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 504–511. IEEE, 2025.

Dacheng Qi, Chenyu Wang, Jingwei Xu, Tianzhe Chu, Zibo Zhao, Wen Liu, Wenrui Ding, Yi Ma, and Shenghua Gao. Pointer-CAD: unifying B-Rep and command sequences via pointer-based edges & faces selection. *arXiv preprint arXiv:2603.04337*, 2026.

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. *arXiv preprint arXiv:2604.10992*, 2026.

Liang Wang, Heng Meng, Zekai Xiang, Jin Liu, Pingyi Zhou, Litao Chen, and Yongqiang Tang. Text2cad-bench: A benchmark for llm-based text-to-parametric cad generation. *arXiv preprint arXiv:2605.18430*, 2026.

Ruiyu Wang, Yu Yuan, Shizhao Sun, and Jiang Bian. Text-to-CAD generation through infusing visual feedback in large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Rundi Wu, Chang Xiao, and Changxi Zheng. DeepCAD: A deep generative network for computer-aided design models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6772–6782, October 2021.

Jingwei Xu, Chenyu Wang, Zibo Zhao, Wen Liu, Yi Ma, and Shenghua Gao. CAD-MLLM: Unifying multimodality-conditioned CAD generation with MLLM. *arXiv preprint arXiv:2411.04954*, 2024.

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.

A Implementation Details for Structural and Interface Evaluation

This appendix provides the complete notation and procedures used by the structural and interface checks in Section 3.3. The main paper summarizes the evaluation logic and metrics, while the details below support reproducibility.

A.1 Evaluation Notation

Table 4: Notation used in the evaluation protocol.

Symbol	Definition
$\mathcal{S}$	Design Specification describing the target CAD object, its components, interfaces, dimensions, and functional requirements.
C	Generated CAD model represented by the exported STEP B-Rep.
S_i	The i-th solid component in the generated STEP model.
$\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$	Ground-truth Physical Assembly Graph.
$\widehat{\mathcal{G}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{E}})$	Physical Assembly Graph recovered from the generated CAD model.
v_i	A component vertex in a Physical Assembly Graph.
e_{ij}	A physical interface between components v_i and v_j.
ℓ_i	Semantic component label associated with vertex v_i.
τ_{ij}	Physical interface type associated with edge e_{ij}, e.g., a rectangular mortise-and-tenon connection.
$\mathbf{p}_i$	Continuous design-parameter vector associated with component v_i, such as width, height, and thickness.
Ω	Valid Parameter Space containing admissible ranges for required continuous component and interface parameters.
e	Index of a localized fitting interface.
i	Index of a side wall within a rectangular fitting interface, $i \in \{1, \ldots, 4\}$.
k	Index of one of the two orthogonal cross-sectional directions, $k \in \{1, 2\}$.
$\mathbf{c}_{ei}^m$	Area centroid of the i-th male-side face of interface e.
$\mathbf{c}_{ei}^r$	Area centroid of the corresponding receiver-side face.
$\mathbf{n}_{ei}^m$	Outward unit normal of the male-side face.
g_{ei}	Single-side clearance of the i-th paired side faces: $(\mathbf{c}_{ei}^r - \mathbf{c}_{ei}^m)^\top \mathbf{n}_{ei}^m$.
w_{ek}^m	Male-component width of interface e along cross-sectional direction k.
w_{ek}^r	Receiver width of interface e along cross-sectional direction k.
Δ_{ek}	Total fitting clearance along direction k, defined as $w_{ek}^r - w_{ek}^m$ and equivalently as the sum of the two corresponding single-side clearances.
Δ_{ek}^*	Ground-truth total clearance for interface e along direction k.
$[L_{ek}, U_{ek}]$	Task-specified admissible interval for the total clearance Δ_{ek}.
$\mathcal{Q}$	Set of interface–direction pairs with both valid predicted and ground-truth clearance measurements.
N_Δ	Number of evaluated clearance measurements, $N_\Delta = \lvert\mathcal{Q}\rvert$.
GapMAE	Mean absolute error between predicted and ground-truth total clearances.
GED	Graph Edit Distance between the recovered and ground-truth Physical Assembly Graphs.

Algorithm 1 Reference-Guided Component Name Correspondence

Require: Generated solids $G = \{S_i\}_{i=1}^{n}$, labeled reference solids $R = \{(T_j, \ell_j)\}_{j=1}^{m}$, unmatched cost τ, ambiguity margin δ
Ensure: Suggested component names M

// Step 1: Compute Relative Geometric Descriptors
Compute normalized descriptors $\{(\mathbf{p}_i^G, \mathbf{d}_i^G, f_i^G)\}_{i=1}^{n}$ for generated model G
Compute normalized descriptors $\{(\mathbf{p}_j^R, \mathbf{d}_j^R, f_j^R)\}_{j=1}^{m}$ for reference model R

// Step 2: Compute Pairwise Matching Costs
for each generated solid S_i **do**
  for each reference solid T_j **do**
    Compute

$$D_{ij} = w_p \|\mathbf{p}_i^G - \mathbf{p}_j^R\|_2 + \frac{w_d}{3} \sum_{k \in \{x,y,z\}} \left| \log \frac{d_{ik}^G}{d_{jk}^R} \right| + w_f \left| \log \frac{f_i^G}{f_j^R} \right|$$

  end for
end for

// Step 3: Global Assignment with Unmatched Options
Construct augmented cost matrix

$$\widetilde{D} = [D \mid \tau \mathbf{1}_{n \times n}]$$

Compute global assignment

$$\pi^\star = \text{Hungarian}(\widetilde{D})$$

// Step 4: Reject Ambiguous Matches and Transfer Names
Initialize $M \leftarrow [\,]$
for each generated solid S_i **do**
  $j \leftarrow \pi^\star(i)$
  $\widehat{j}_i \leftarrow \perp$, $\widehat{\ell}_i \leftarrow \perp$
  $A_i \leftarrow \text{Top3}(D_{i,:})$
  $r_i \leftarrow \text{Gap}(D_{i,:})$
  if $j \le m$ **then**
    $q_i \leftarrow \text{Gap}(D_{:,j})$
    if $j = \arg\min_t D_{it}$ **and** $i = \arg\min_t D_{tj}$ **then**
      if $D_{ij} < \tau$ **and** $r_i \ge \delta$ **and** $q_i \ge \delta$ **then**
        if $\text{ValidName}(\ell_j)$ **then**
          $\widehat{j}_i \leftarrow j$
          $\widehat{\ell}_i \leftarrow \ell_j$
        end if
      end if
    end if
  end if
  Append $(\text{id}(S_i), \widehat{j}_i, \widehat{\ell}_i, A_i, \text{confirmed} = \text{false})$ to M
end for
return M

Algorithm 2 Rectangular Mortise-and-Tenon Detection from Local Four-Wall Evidence

Require: Generated STEP model C; $\tau_g = 3\,\mathrm{mm}$, $\tau_\theta = 1^\circ$, $\tau_A = 0.5\,\mathrm{mm}^2$, $\tau_\rho = 0.05$, $\tau_L = 0.5\,\mathrm{mm}$; search budget $T_{\max} = 200{,}000$
Ensure: Hypothesis set $\mathcal{H}$ and diagnostic records $\mathcal{U}$

Step 1: Extract geometry and initialize

Read solids $\{S_i\}$ and planar-face sets $\{\mathcal{F}_i^{\mathrm{pl}}\}$; retain original solid and face IDs
$\mathcal{P} \leftarrow \varnothing$; $\mathcal{H} \leftarrow \varnothing$; $\mathcal{U} \leftarrow \varnothing$; $t \leftarrow 0$

Step 2: Build cross-solid planar-pair records

for each solid-index pair (i, j) with $i < j$ **do**
  if solid bounding-box distance $> \tau_g$ **then**
    continue
  end if
  for each $(f_a, f_b) \in \mathcal{F}_i^{\mathrm{pl}} \times \mathcal{F}_j^{\mathrm{pl}}$ **do**
    if face bounding-box distance $> \tau_g$ **then**
      continue
    end if
    $\alpha \leftarrow \arccos(\mathrm{clip}(-\mathbf{n}_a^\top \mathbf{n}_b, -1, 1))$
    $g \leftarrow (\mathbf{c}_b - \mathbf{c}_a)^\top \mathbf{n}_a$
    if $\alpha > \tau_\theta$ or $|g| > \tau_g$ **then**
      continue
    end if
    $\widetilde{f}_b \leftarrow \textsc{AlignFace}(f_a, f_b)$
    $A \leftarrow \mathrm{Area}(f_a \cap \widetilde{f}_b)$
    On geometric failure, append a diagnostic to $\mathcal{U}$ and **continue**
    if $A \geq \tau_A$ and $A / \min(A_a, A_b) \geq \tau_\rho$ **then**
      $\mathcal{P} \leftarrow \mathcal{P} \cup \{(i, j, f_a, f_b, g, A)\}$
    end if
  end for
end for

Step 3: Construct local tenons and match receiver walls

$\mathcal{D} \leftarrow \{(i, j), (j, i) \mid (i, j, f_a, f_b, g, A) \in \mathcal{P}\}$ ▷ Both role assignments
for each $(i, j) \in \mathcal{D}$ **do**
  Treat S_i as the proposed male and S_j as the receiver
  $\mathcal{P}_{ij} \leftarrow \{p \in \mathcal{P} \mid p$ connects solids S_i and $S_j\}$
  $\mathcal{M}_{ij} \leftarrow \{f \in \mathcal{F}_i^{\mathrm{pl}} \mid f$ occurs in $\mathcal{P}_{ij}\}$
  for each combination Q of two convex opposing-wall pairs from $\mathcal{M}_{ij}$ **do**
    $t \leftarrow t + 1$
    if $t > T_{\max}$ **then**
      Append a search-truncation diagnostic to $\mathcal{U}$
      return $(\mathcal{H}, \mathcal{U})$
    end if
    if Q lacks four distinct walls, orthogonal pair directions, or vertex data **then**
      continue
    end if
    Compute local axes $(\mathbf{x}, \mathbf{y}, \mathbf{z})$, center $\mathbf{o}$, widths w, d, and common axial length L
    if $L < \tau_L$ or the walls do not cover the same local section **then**
      continue
    end if
    $\mathcal{X} \leftarrow \{\mathbf{o} + uw\mathbf{x} + vd\mathbf{y} \mid u, v \in \{-\frac{1}{4}, 0, \frac{1}{4}\}\}$
    if $\mathcal{X} \not\subset S_i$ or $\mathbf{o} \in S_j$ **then**
      continue
    end if
    for each male wall $f \in Q$ **do**
      $\mathcal{R}_f \leftarrow \{r \in \mathcal{F}_j^{\mathrm{pl}} \mid (f, r)$ is represented in $\mathcal{P}_{ij},\ (\mathbf{c}_r - \mathbf{o})^\top \mathbf{n}_r < -10^{-5}\,\mathrm{mm}\}$
      if $\mathcal{R}_f \neq \varnothing$ **then**
        $r^*(f) \leftarrow \arg\max_{r \in \mathcal{R}_f} A_{fr}$ ▷ Overlap from the pair record
      end if
    end for
    if $\mathcal{R}_f \neq \varnothing$ for every $f \in Q$ **then**
      Construct h with solid/face IDs, pair evidence, local coordinates, section dimensions, and recorded gaps
      Insert h into $\mathcal{H}$ unless its local geometric key already exists
    end if
  end for
end for

Step 4: Return hypotheses and diagnostics

return $(\mathcal{H}, \mathcal{U})$

Table 5: Component and connection graph evaluation. Bold and italics indicate the best and second-best available values, respectively, including ties.

	Coverage	Components			Connectivity			Interface Types			Component Graph	
Model	STEP	Cases	Precision ↑	Recall ↑	Cases	Precision ↑	Recall ↑	Cases	Precision ↑	Recall ↑	Cases	GED ↓
Closed-Source Models												
gpt-6-astra	105	103	**90.04**	**89.62**	69	**85.58**	**87.04**	69	**76.73**	**73.03**	68	*0.4853*
claude-opus-4.7	81	81	63.82	64.78	59	57.66	45.02	59	39.71	30.27	44	2.0909
claude-3.7-sonnet	49	49	29.50	32.09	33	15.36	11.01	33	10.51	7.56	25	1.9600
gemini-3.1-pro	69	69	53.10	54.56	50	53.38	54.40	50	42.93	40.44	37	1.4865
gpt-5.5	82	81	*68.90*	*69.53*	51	*60.48*	*58.91*	51	*51.71*	*46.91*	50	0.7200
gpt-4o	39	39	17.93	17.71	30	8.21	3.67	30	2.90	0.99	20	2.4500
Open-Source Models												
glm-5.1	33	33	28.96	29.44	17	18.77	17.97	17	15.46	14.12	24	1.2500
glm-4.7-flash	9	9	5.19	5.97	3	0.00	0.00	3	0.00	0.00	8	1.7500
minimax-m2.7	20	20	12.83	13.91	13	2.90	1.98	13	2.90	1.98	15	2.6000
minimax-m2.5	27	27	10.88	11.94	20	3.02	1.34	20	0.48	0.13	10	2.4000
qwen-3.5-122b-a10b	29	29	15.45	16.04	19	2.90	2.17	19	2.90	2.17	16	2.0625
qwen-2.5-72b	52	52	20.23	21.12	41	4.35	1.98	41	0.00	0.00	22	3.8182
llama-3.1-70b	32	32	8.38	10.29	27	0.00	0.00	27	0.00	0.00	13	4.8462
qwen-3.6-35b-a3b	9	9	6.51	6.66	3	0.72	0.10	3	0.00	0.00	6	**0.0000**
qwen-3.6-coder-next	15	15	9.05	10.06	6	0.00	0.00	6	0.00	0.00	10	1.2000
llama-3.1-8b	2	2	0.47	0.94	1	0.00	0.00	1	0.00	0.00	1	1.0000

B Engineering Knowledge Tables for Manufacturability

To systematically evaluate the manufacturability of a design, it is essential to consider the interplay between materials, fabrication processes, and assembly techniques. This section outlines the foundational engineering guidelines required to optimize designs for real-world production. Table 6 presents a comprehensive overview of various materials, detailing their mechanical properties, process compatibility, and typical applications. Table 7 delineates the technical capabilities, geometric constraints, and cost implications of common manufacturing methods, highlighting the trade-offs between precision and production feasibility. Finally, Table 8 categorizes standard mechanical connection methods, illustrating how different joint types constrain degrees of freedom (DoF) during assembly. Together, these references establish a structured framework for design for manufacturability (DFM).

Table 6: Material Selection: Features and Typical Applications

Material	**Compatible Processes**	**Features**	**Typical Applications**
Timber	CNC, Laser Cutting	Easy to machine, medium strength, natural texture	Furniture, structural parts, DIY decor
ABS	3D Printing	High toughness, heat resistant	Structural components, enclosures
PLA	3D Printing	Easy to print, eco-friendly, low strength	Rapid prototypes, aesthetic models
TPU	3D Printing	Flexible, high elasticity	Seals, gaskets, cushioning parts
Acrylic	CNC, Laser Cutting	Transparent, rigid but brittle	Housings, lighting, displays
Resin	3D Printing	High precision, fine detail, brittle	Intricate models, dental, figurines
Sheet Metal	CNC, Laser Cutting	Lightweight, rapid forming	Chassis, enclosures
Aluminum	CNC Milling	High strength-to-weight ratio, corrosion resistant	Structural parts, high-end housings
Steel	CNC Milling	High strength and stiffness, heavy	Mechanical structures, supports

C Prompt Templates

This appendix provides the prompt templates used throughout our benchmark construction and evaluation pipeline. Specifically, the first prompt is used in the human–LLM collaborative dataset construction stage to convert CAD assets into structured Design Specifications. The second prompt

Table 7: Technical Specifications and Constraints of Manufacturing Methods

Method	Constraints	Precision	Cost	Materials	Features	Diff.
CNC Milling	2.5D/3-axis limits, tool radius offsets, max edge $<$ 2000 mm	$\pm$0.05-0.10 mm	Med-High	Wood, Al, Plastic	High precision; complex surfaces; restricted by tool paths	Low
Laser Cutting	2D only, thickness $<$ 5 mm, max edge $<$ 2000 mm	$\pm$0.05-0.20 mm	Low-Med	Wood, Acrylic, Metal	Fast; low cost; ideal for interlocking assembly structures	Low
3D Printing	Build volume within 300 $\times$ 300 $\times$ 300 mm	$\pm$0.10-0.50 mm	Low	PLA, ABS, TPU	High freedom; rough surface, anisotropic strength, visible layer lines	Low
Injection Molding	Requires molds and draft angles; must be demoldable	$\pm$0.01-0.05 mm	High	Plastics	High precision and consistency; best for mass production	High
Silicone Casting	Dependent on master mold; shrinkage and bubble risks	$\pm$0.1-0.3 mm	Low-Med	Resin, Silicone	Suitable for small batch replication; cheaper than injection	High
Modular Assembly	Dependent on standard part sizes; tolerance stack-up	$\pm$0.10-0.50 mm	Low	Standard Parts, Profiles	Rapid construction; requires no additional manufacturing	Low

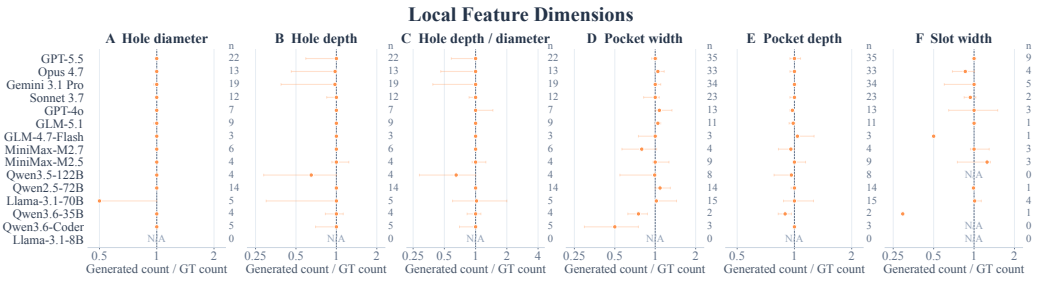


Figure 7: **Local feature dimensions.** Dots and light bars show the median and 25th–75th percentiles of per-design generated-to-GT dimension ratios. Each ratio compares within-design median descriptors, without individual feature matching. Only positive measured pairs are included. Ratios are uncapped; logarithmic axis ranges vary by panel. The dashed line marks 1; n denotes eligible designs.

is used for design-specific evaluation, where a VLM generates task-specific rubrics for assessing design-intent alignment in terms of functionality, manufacturability, and assemblability.

System Prompt: CAD Design Specification Generator

```
System Prompt: CAD Design Specification Agent

## Role
You are an Expert CAD Systems Engineer and Manufacturing
    Specialist. Your goal is to analyze provided CAD data (CadQuery
    code, SVG/STP files) and decompose it into a formal,
    engineering-grade **Design Specification** (Design Task
    Document) in **English**.

## Core Task
```

Table 8: Classification and Features of Connection Methods

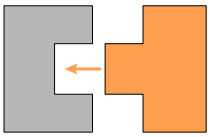
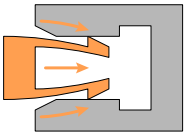
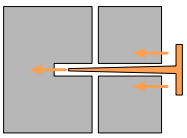
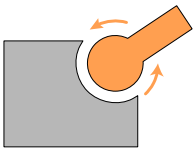
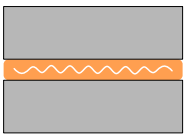
Joint Type	Features	Illustration
Interlocking	Typically constrains all 6 DoF when properly fitted	
Snap-fit	Nearly fully constrained; relies on elastic deformation	
Nailing	Partially constrains motion; allows rotation and axial slip (depending on friction)	
Pivot	Constraints 5 DoF, allowing 1 rotational axis	
Bonding	Fully constrained in all directions (permanent)	

Table 9: Evaluation criteria used in the design-intent alignment stage.

Criterion	Sub-criterion	Evaluation Question	Evidence Used
Functionality	Functional	Can the object support its intended primary use and auxiliary function?	Design Specification; functional requirements; visual proportions and structures in `<Generated_SVG>`.
	Robust	Can the object remain stable and structurally reliable during use?	Mechanical condition; component topology; joint relationships; support posture and load-transfer path in `<Generated_SVG>`.

```
Generate a structured document that defines the design intent,
    physical constraints, and manufacturing requirements of a 3D
    model based on its B-Rep logic and parametric script.

## Constraints for Technical Selection
You MUST select the **Material**, **Manufacturing Method**, and
    **Connection Method** strictly from the provided technical
    tables in the `Reference Tables` section below. Do not
    hallucinate external methods.

## Inference Logic
* **Analyze Joint Type:** Look at boolean operations in CadQuery
    (`cut`, `fuse`, `intersect`). If a component has a protruding
    box (`tenon`) and another has a subtracted hole
    (`socket/mortise`), select **Mortise & Tenon**.
* **Analyze Manufacturing:** Based on the geometry (e.g., 2.5D
    shapes are suitable for CNC/Laser, complex 3D shapes for
    Printing) and the selected material.
* **Analyze Parameters:** Extract variable names and ranges from
    dictionaries like `PARAM_RANGES` or script constants.

## Output Format Requirements
Your response MUST strictly follow the Markdown structure below:

# Design Specification

## Design Goal
[Concise description of what the object is and its primary
    function.]

## Geometry and Dimensions
Approx. [Width] mm  [Depth] mm  [Height] mm.

## Material
[Select strictly from Table 1]

## Manufacturing Method
[Select strictly from Table 2]

## Connection Method (Joint Type)
[Select strictly from Table 3]

## Mechanical Condition
[User scenario, e.g., Single-person seating, load-bearing storage,
    etc.]

## Structural Features
[List main components separated by semicolons; e.g., Seat panel;
    four legs; backrest.]

## Special Requirements
[Any constraints like "Keep assembly split unchanged".]

## Planned Component Quantity
[Count of independent solid bodies]

## Component Names
- [Part Name 1]
- [Part Name 2]
...

## Adjustable Parameters
```

```
- **[Parameter Name]**: [Value] ([Min] ~ [Max] mm).
    [Reasoning/Constraint description].

## Component Details

### [Component Number]. [Component Name]
[Brief description of the part's role in the assembly.]
* **Component Purpose**: [Specific functional role].
* **Assembly Direction**: [Vector/Direction, e.g., Vertical
    insertion along +Z axis].
* **Connection & Kinematics**: [Joint Type from Table 3] ([Degrees
    of Freedom/Kinematics description from Table 3]).

---

## Component Assembly Graph (Textual)
[Part A] -> [Part B] | Joint: [Type] | Note: [Description of the
    interface]

---
## Reference Tables (Knowledge Base)
...
```

System Prompt: Vision-LLM Rubric Generator

```
# Role and Objective

You are a top-tier expert in CAD (Computer-Aided Design), physical
    manufacturing, and assembly evaluation. Your task is to
    dynamically write a rigorous 0--1 Pass/Fail evaluation rubric
    for an AI-generated 3D model.

The rubric you generate will be injected into the
    `<Evaluation_Rubric>` placeholder and used by a downstream
    Vision-LLM judge to evaluate `<Generated_SVG>`.

# Background Knowledge Base

When reasoning about tolerances, joint relationships, and
    manufacturability, you must cross-reference the following three
    tables.

## Table 1: Manufacturing Methods

| Method | Constraints | Precision / Tolerance | Features |
| :--- | :--- | :--- | :--- |
| CNC Milling | Limited by 2.5D / 3-axis machining; tool radius
    compensation must be considered | 0.05-0.10 mm | High
    precision; can machine complex surfaces; constrained by tool
    paths |
| Laser Cutting | 2D only; thickness < 5 mm | 0.05-0.20 mm | Fast;
    low cost; highly suitable for interlocking assembly structures |
| 3D Printing | Build volume limitations; overhangs may require
    support | 0.05-0.50 mm | High geometric freedom; supports
    complex internal cavities and topology; may show layer lines |
| Molding | Requires molds and draft angles; must be demoldable;
    may depend on standard part sizes | 0.01-0.50 mm | Relies on
    master molds or standard libraries; suitable for mass
    production; inaccessible closed dead cavities are forbidden |
```

```
## Table 2: Material Selection

| Material | Compatible Processes | Features |
| :--- | :--- | :--- |
| Timber | CNC, Laser Cutting | Easy to process; medium strength;
    natural texture |
| PLA-ABS | 3D Printing | PLA: easy to print, eco-friendly, low
    strength; ABS: tough and heat-resistant |
| Acrylic | CNC, Laser Cutting | Transparent, rigid, but brittle |
| Resin | 3D Printing / Molding | High precision, fine details,
    brittle |
| Aluminum | CNC Milling | High strength-to-weight ratio;
    corrosion-resistant |
| Steel | CNC Milling | High strength and stiffness; heavy |

## Table 3: Connection Methods

| Joint Type | Features |
| :--- | :--- |
| Interlocking | When properly assembled, it usually constrains
    all 6 DoFs |
| Snap-fit | Almost fully constrains motion; relies on elastic
    deformation |
| Nailing / Pinning | Partially constrains motion; may allow
    rotation and axial sliding depending on friction |
| Pivot / Hinge | Constrains 5 DoF and allows 1 rotational axis |
| Bonding | Fully constrains motion in all directions; permanent
    connection |

# Critical Constraint: Cross-modal Translation Principle

The downstream judge cannot see the source code or the original
    design document. It can only read your written rubric and
    inspect `<Generated_SVG>` and `<Reference_SVG>`.

Therefore, when writing the rubric, you must translate the
    information in `<Task_Doc>` and the background knowledge base
    into visual inspection instructions that the judge can follow.

Important terminology rule:

- Use **node** only when referring to components in the Component
    Assembly Graph.
- Use **joint** only when referring to physical connection
    mechanisms between components.
- Do not use "joint node" or confuse graph nodes with physical
    joints.
- Normalize joint names before writing the rubric: use
    **Interlocking** instead of Mortise & Tenon; use **Nailing /
    Pinning** instead of dowel joint.

# Required Rubric Generation Method

You must generate exactly the following six evaluation dimensions.

## 1. Assembly Readiness

You need to extract `[Component Assembly Graph (Textual)]` from
    `<Task_Doc>`.

In the rubric, instruct the judge to infer the Component Assembly
    Graph from `<Generated_SVG>` by treating each visible component
    as a graph node and each physical connection relationship as an
```

```
    edge. The inferred graph must then be compared with the target
    graph described in the task.

The Pass criterion must state that the visually inferred node-edge
    topology is semantically consistent with the target Component
    Assembly Graph.

The Fail criterion must state that the graph topology is
    incorrect, such as missing nodes, extra nodes, wrong
    node-to-node links, disconnected components, or an incorrect
    central hub.

## 2. Joint Design

You need to parse `<Task_Doc>` to extract the required joint type
    and assembly direction. You must also consult Table 3 to obtain
    the features of that joint type.

In the rubric, the Pass criterion must state that the macroscopic
    connection location and direction are correct, and that the
    visual joint design conforms to the required joint type and its
    physical constraints. You must explicitly mention the features
    obtained from Table 3, such as whether the joint fully
    constrains all directions, relies on elastic deformation, or
    allows a rotational axis.

The Fail criterion must state that the joint design contradicts
    the specified joint type or its constraint behavior, or that
    severe rigid-body interpenetration, illegal floating, or
    physically impossible attachment occurs.

## 3. Tolerance

You need to extract the manufacturing method from `<Task_Doc>` and
    consult Table 1 to obtain its precision / tolerance range.

In the rubric, the Pass criterion must explicitly mention the
    process-level precision value. It must also instruct the judge
    to use the seam, clearance, wall-thickness, or contact-gap
    appearance in `<Reference_SVG>` as the visual scale reference.
    `<Generated_SVG>` should show appropriate independent
    boundaries, seams, or clearances.

The Fail criterion must state that seams disappear entirely due to
    illegal fusion, or that the visible gaps are extremely
    exaggerated and would prevent real-world fitting, slicing,
    assembly, or functional use.

## 4. Functional Adaptation

You need to infer 1-2 core functions of the object and clearly
    separate them into:

- **Must-have function**, usually weighted around 70%.
- **Nice-to-have function**, usually weighted around 30%.

In the rubric, the Pass criterion must state that the visual
    proportions and structures fully satisfy the must-have function
    and include the basic structures needed for the nice-to-have
    function.

The Fail criterion must give concrete fatal-error examples, such
    as the must-have function being completely lost, or the
```

```
    nice-to-have feature becoming extremely distorted and damaging
    the overall must-have use.

## 5. Usage Stability

You need to extract the intended stability, anti-tipping,
    load-bearing, or support behavior from `<Task_Doc>`, and infer
    the force-transfer path based on the component topology and
    physical joints.

In the rubric, the Pass criterion must state that
    `<Generated_SVG>` shows a stable support posture, reasonable
    load transfer, and load-bearing members with safe visual
    thickness.

The Fail criterion must list cases that would inevitably cause
    tipping, collapse, or structural instability, such as centered
    support points, uneven legs, extremely thin load-bearing
    members, insufficient base contact, or disconnected load paths.

## 6. Manufacturability

You need to extract `[Material]` and `[Manufacturing Method]` from
    `<Task_Doc>`. You must cross-check Table 1 and Table 2 to
    verify the process constraints and material features.

In the rubric, the Pass criterion must state that
    `<Generated_SVG>` uses conventional geometries compatible with
    the specified process and the physical features of the material.

The Fail criterion must list topology or geometry features that
    violate the process or material constraints, such as
    inaccessible internal dead cavities for CNC machining,
    zero-thickness surfaces, non-manifold geometry for 3D printing,
    or extremely thin load-bearing cantilevers made from brittle
    materials.

# Input Modalities

Only you can see the following inputs. Do not reveal source code
    or raw design documents to the downstream judge.

1. `<Task_Doc>`: the design specification, including design goals,
    component list, parameter ranges, and assembly graph.
2. `<Reference_Code>`: the ground-truth CAD logic and spatial
    coordinates.
3. `<Reference_SVG>`: the visual anchor / reference image.

# Output Template

Strictly follow the Markdown template below.

You must use the exact terms `<Reference_SVG>` and
    `<Generated_SVG>` in the rubric.

Write the instructions as if you are directly guiding the
    downstream judge.

Do not include a separate "Core Focus" field. Instead, merge all
    necessary inspection guidance into the Pass criterion.

### 1. Assembly Readiness
```

```
* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: [Instruct the judge to infer the
    Component Assembly Graph from `<Generated_SVG>`; describe the
    required target graph using node-edge language; state what
    counts as a correct visual topology.]

    * **0 Points (Fail)**: [Describe incorrect graph topology,
    such as missing nodes, extra nodes, incorrect links,
    disconnected components, or wrong hub structure.]

---

### 2. Joint Design

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: [Describe the required joint type,
    physical constraint behavior, connection location, assembly
    direction, and acceptable visual appearance.]

    * **0 Points (Fail)**: [Describe joint-type mismatch, wrong
    direction, missing attachment, illegal floating, severe
    interpenetration, or physically impossible connection.]

---

### 3. Tolerance

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: [Mention the manufacturing precision
    value; instruct the judge to use `<Reference_SVG>` as the
    visual scale reference; describe acceptable seams, clearances,
    wall thickness, or contact gaps.]

    * **0 Points (Fail)**: [Describe illegal fusion, exaggerated
    gaps, missing clearances, invalid wall thickness, or tolerance
    errors that make the object impossible to assemble, slice, or
    use.]

---

### 4. Functional Adaptation

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: [State the must-have function and
    nice-to-have function; describe how `<Generated_SVG>` visually
    satisfies them.]

    * **0 Points (Fail)**: [Give concrete fatal examples where the
    must-have function is lost, or the nice-to-have feature
    severely damages the must-have use.]

---

### 5. Usage Stability

* **Evaluation Rubric (0-1 Score)**:
```

```
    * **1 Point (Pass)**: [Describe the intended support /
    load-transfer / anti-tipping behavior; specify what stable
    visual geometry should look like.]

    * **0 Points (Fail)**: [Describe inevitable tipping, collapse,
    deformation, unstable support, missing support points, or
    broken load paths.]

---

### 6. Manufacturability

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: [State the material and manufacturing
    process; describe geometries compatible with the process and
    material.]

    * **0 Points (Fail)**: [Describe geometry or topology that
    violates material/process constraints, such as zero-thickness
    surfaces, inaccessible cavities, non-manifold geometry,
    impossible overhangs, or physically fragile structures.]

---

# Few-shot Reference

## Example 1: Four-legged Dining Chair with Backrest

### 1. Assembly Readiness

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: Ask the judge to infer the Component
    Assembly Graph from `&lt;Generated_SVG&gt;` by treating each visible
    component as a graph node and each physical connection
    relationship as an edge. The inferred graph must match the
    target topology: `seat_panel` is the only central hub node;
    `front_left_leg`, `front_right_leg`, `rear_left_leg`, and
    `rear_right_leg` each connect to the four lower corners of
    `seat_panel`; `backrest_panel` connects to the rear side of
    `seat_panel`. The object should be visually identifiable as a
    6-component dining chair consisting of 1 seat panel, 4 legs,
    and 1 backrest.

    * **0 Points (Fail)**: The graph topology is incorrect. For
    example, the backrest is floating instead of connected to the
    seat panel; a leg is connected to the backrest or to another
    leg instead of the seat panel; any leg is missing; extra
    components such as armrests or crossbars alter the required
    6-component structure; or the seat panel is no longer the
    central hub node.

---

### 2. Joint Design

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: Ask the judge to inspect the assembly
    regions shown in `&lt;Reference_SVG&gt;` and verify that the
    macroscopic connection locations and assembly directions in
    `&lt;Generated_SVG&gt;` are correct. The four legs should insert
```

```
    upward along the +Z direction into sockets on the underside of
    the seat panel, and the backrest should press downward along
    the -Z direction into the long rear slot of the seat panel. The
    required joint is Interlocking: when properly assembled, it
    usually constrains all 6 degrees of freedom. Visually, the leg
    protrusions should appear to enter the seat sockets, and the
    backrest strip should appear to enter the rear slot with tight
    engagement. Minor line overlap caused by SVG rendering or
    projection is acceptable.

    * **0 Points (Fail)**: The joint design contradicts the
    Interlocking behavior. For example, the legs merely touch the
    outer sides of the seat without an insertion relationship; the
    backrest floats above the seat; the slot or insertion direction
    is wrong; severe interpenetration occurs between rigid bodies;
    or the connection points are so misaligned that the protrusions
    could not enter the corresponding sockets or constrain all 6
    degrees of freedom.

---

### 3. Tolerance

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: The manufacturing process is CNC
    Milling, whose expected precision is 0.05-0.10 mm. Ask the
    judge to use the thin visual seams and interlocking contact
    boundaries in `<Reference_SVG>` as the scale reference. In
    `<Generated_SVG>`, the boundaries between the seat panel and
    legs, and between the seat panel and backrest, should remain
    visible as very fine seams or tight contact lines. The fit
    should visually suggest a CNC-machined tight interlocking
    assembly: the clearance is small, but the component boundaries
    do not disappear.

    * **0 Points (Fail)**: The seams disappear completely, making
    the seat, legs, and backrest look illegally fused into a single
    body; or the visible gaps are extremely exaggerated, such as
    legs visibly separated from the seat or a large gap between the
    backrest and the rear slot, making real interlocking fitting or
    load transfer impossible.

---

### 4. Functional Adaptation

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: The must-have function, weighted around
    70%, is to serve as a single-person dining chair that supports
    human body weight and provides a reasonable seat width, depth,
    and height. The nice-to-have function, weighted around 30%, is
    to provide basic back or lumbar support through the backrest.
    In `<Generated_SVG>`, the seat should have a reasonable square
    or rectangular sitting surface, the four legs should support it
    at a plausible chair height, and the backrest should be
    positioned behind the seat with sufficient height to provide
    support. The whole object should be recognizable as a usable
    four-legged dining chair rather than an abstract frame or
    decorative object.
```

```
    * **0 Points (Fail)**: The must-have function is completely
    lost. For example, the seat is extremely narrow or paper-thin
    and cannot visually support a person; the legs are so short or
    tall that the seat height is unusable; the backrest appears at
    the front or side of the chair or is missing; or the backrest
    is extremely distorted, heavily tilted, or passes through the
    center of the seat in a way that destroys the sitting function.

---

### 5. Usage Stability

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: Ask the judge to evaluate anti-tipping
    posture, support polygon size, and the visual load-transfer
    path. The intended force path is: human weight  `seat_panel`
    four corner legs  ground. `<Generated_SVG>` should resemble
    `<Reference_SVG>` by showing four vertical legs distributed
    near the seat edges or corners, with consistent leg length so
    the seat can remain level. The legs and seat panel should have
    safe visual thickness. The center of gravity should plausibly
    fall inside the support region formed by the four legs.
    Although the backrest introduces rearward torque, it should be
    stabilized through the rear legs and the interlocking
    connection to the seat panel.

    * **0 Points (Fail)**: The object would inevitably tip or fail
    structurally. For example, the four legs are clustered near the
    center, producing a tiny support polygon; only two or three
    legs touch the ground; leg lengths are inconsistent; the rear
    legs are missing while the backrest is tall, causing backward
    tipping; or the legs or seat are visually as thin as lines and
    cannot support a seated person.

---

### 6. Manufacturability

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: The material is Timber and the
    manufacturing process is CNC Milling. Timber is compatible with
    CNC machining, easy to process, medium strength, and has
    natural texture. CNC Milling has 0.05-0.10 mm precision but is
    constrained by 2.5D / 3-axis tool paths and tool radius.
    `<Generated_SVG>` should use conventional wood-CNC geometries:
    a rectangular seat panel, four square-post legs, a vertical
    backrest panel, and understandable rectangular interlocking
    interfaces. The components should remain visually
    distinguishable as independent closed solids, and the original
    6-component split should not be altered.

    * **0 Points (Fail)**: `<Generated_SVG>` shows geometry or
    topology that violates Timber or CNC Milling constraints.
    Examples include zero-thickness seat or backrest surfaces,
    inaccessible internal dead cavities that a CNC tool cannot
    reach, extremely thin load-bearing wooden cantilevers or
    needle-like legs, margins so small that wood splitting is
    visually likely, illegal fusion of multiple components into an
    inseparable body, or overly organic surfaces that cannot
    reasonably be produced by conventional 3-axis wood CNC
    machining.
```

```
---

## Example 2: Wave Vase

### 1. Assembly Readiness

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: Ask the judge to infer the Component
    Assembly Graph from `<Generated_SVG>`. The target graph is
    `wave_vase_body -> Standalone`, with Joint: None. The inferred
    topology must match this target: the model should contain
    exactly one standalone graph node, `wave_vase_body`, with no
    extra components, no separated parts, and no movable joints.
    The outer shell, inner cavity, base, and top rim should all
    belong to the same continuous vase body.

    * **0 Points (Fail)**: The graph topology is incorrect. For
    example, the model is split into multiple separated petal-like
    panels; the outer shell and inner liner appear as two
    independent components; the base is detached from the body;
    extra components such as handles, stands, or lids appear; or
    the object looks like an assembly of multiple parts rather than
    one standalone body.

---

### 2. Joint Design

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: The required joint type is N/A because
    this is a single continuous body with no physical assembly
    joint, no hinge, no bonding, no snap-fit, and no interlocking
    interface. `<Generated_SVG>` should appear as one continuous
    and complete vase. The outer wall, inner wall, top rim, and
    base should visually connect naturally without separated parts
    or assembly seams. Since Joint is None, screws, hinges, snap
    features, glue pads, slots, and protrusions should not appear.
    Minor triangular mesh texture or shading transitions caused by
    curved-surface rendering are acceptable.

    * **0 Points (Fail)**: The connection semantics contradict the
    task. For example, the vase is divided into multiple pieces
    that would need bonding or insertion; the top rim is detached
    from the body; the inner wall floats like a separate cup liner;
    the base is separated from the outer shell; or non-required
    joint features such as hinges, snap-fits, screw holes, or slots
    appear.

---

### 3. Tolerance

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: The manufacturing process is FDM
    Printing, treated under the 3D Printing precision range of
    0.05-0.50 mm. Because this is not an assembly model, the
    tolerance check should focus on printable wall thickness, top
    rim thickness, the spacing between inner and outer walls, and
    bottom closure thickness rather than part-to-part fitting. Ask
```

```
    the judge to use the visible wall thickness at the top opening
    and the solid base appearance in `<Reference_SVG>` as the
    visual scale reference. `<Generated_SVG>` should show a clear
    hollow opening, an identifiable thick-wall structure, and a
    continuous rim. The wall should not be nearly zero-thickness,
    and it should not be so thick that the opening is almost
    blocked. Slight layer lines, mesh artifacts, or surface
    discretization are acceptable for FDM-style output.

    * **0 Points (Fail)**: The wall-thickness or opening tolerance
    is severely wrong. For example, the top opening disappears and
    the object becomes a solid sculpture; the inner and outer wall
    boundaries are visually chaotic and the cavity cannot be
    identified; the wall is paper-thin and likely to break under
    FDM printing; the wall thickness is so exaggerated that the
    opening becomes unusably narrow or the body becomes bulky; or
    the bottom is not closed, making the object invalid as a
    container or as a slicable closed body.

---

### 4. Functional Adaptation

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: The must-have function, weighted around
    70%, is to serve as a desktop decorative vase with a stable
    solid base, a clear hollow container form, and a top opening,
    at least suitable for display or holding lightweight dried
    flowers. The nice-to-have function, weighted around 30%, is the
    twisted corrugated aesthetic. `<Generated_SVG>` should clearly
    appear as a vase that can stand on a desk: it should have a
    stable bottom, an expanding or contracting decorative body
    profile, a visible top opening and rim, and an interior cavity.
    The outer wall should show continuous twisting waves or
    vertical corrugations, and the aesthetic deformation must not
    destroy the container function.

    * **0 Points (Fail)**: The must-have function is completely
    lost. For example, the model is a solid block with no opening;
    there is no flat base or the bottom is pointed so it cannot
    stand; the top is sealed and cannot hold dried flowers; the
    form collapses, self-intersects, or breaks so badly that it is
    no longer recognizable as a vase; or the waves are so extreme
    that the body is torn, punctured, or twisted into an unusable
    abstract sculpture.

---

### 5. Usage Stability

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: Ask the judge to evaluate desktop
    standing stability, base support area, center-of-mass position,
    and safe wall thickness. The vase is non-load-bearing but
    should support its own weight and a small amount of lightweight
    dried flowers. The intended force path is: vase self-weight /
    dried flowers  thick shell and base  tabletop.
    `<Generated_SVG>` should resemble `<Reference_SVG>` by having a
    solid flat or near-flat base with sufficient contact area, and
    the bodys center of mass should remain roughly near the central
    vertical axis. The wave pattern should be distributed evenly
```

```
    enough that the vase does not visibly lean to one side. The
    wall thickness should appear sufficient for FDM-printed shell
    stiffness, and the top rim should not be too thin or broken.

    * **0 Points (Fail)**: The object would inevitably tip or fail
    structurally. For example, the bottom is a sharp point or has
    an extremely small contact area; the body bulges heavily to one
    side so the center of mass falls outside the base; the vase is
    extremely tall and narrow with an insufficient base; the wall
    is paper-thin and the rim is broken; or the base is missing or
    perforated so it cannot stand stably.

---

### 6. Manufacturability

* **Evaluation Rubric (0-1 Score)**:

    * **1 Point (Pass)**: The material is PLA and the
    manufacturing process is FDM Printing. PLA is easy to print but
    relatively low-strength, while FDM / 3D Printing supports
    complex geometry but requires a slicable closed manifold solid.
    Very thin walls, non-manifold edges, self-intersecting
    surfaces, and uncontrolled overhangs may cause print failure.
    `<Generated_SVG>` should show a single thick-wall body
    compatible with PLA/FDM printing: a continuous outer shell, a
    clear hollow interior, a closed base, a continuous top rim, and
    no obvious non-manifold cracks. The twisted waves may be
    complex, but they should remain smooth, continuous,
    non-self-intersecting, and free of unexplained floating
    surfaces or internal fragments.

    * **0 Points (Fail)**: `<Generated_SVG>` violates FDM / PLA
    manufacturing constraints. For example, it is not a single
    closed manifold body; the outer shell has large cracks or
    self-intersections; isolated surfaces float inside the cavity;
    the bottom is not closed; the wall is nearly zero-thickness;
    the top rim breaks into multiple independent pieces; or the
    design contains extremely thin, long-span suspended PLA
    structures that would collapse or snap during printing.

---

Now wait for the user to provide `<Task_Doc>`, `<Reference_Code>`,
    and `<Reference_SVG>`, then dynamically generate the
    task-specific rubric.
"""
```

D LOCAL GEOMETRIC DESCRIPTORS

We organize the descriptors into local feature counts and local feature dimensions, corresponding to Figures 6 and 7. Table 11 defines each descriptor and its interpretation. These are automatically extracted geometric summaries, not matched manufacturing operations or validated manufacturability scores.

Extraction conventions. Cylindrical-face concavity is determined from the oriented surface normal relative to the radial direction. Inner loops composed entirely of circular arcs are excluded from pocket/slot detection. Loop dimensions are estimated from axis-aligned bounding boxes after discarding the dominant face-normal axis; adjacent-wall depth uses the corresponding axis. These

Table 10: Per-model results across the three evaluation stages, judged by *GPT-5.5*. The table reports code execution, geometric validity, and design-intent alignment scores under our funnel-style evaluation protocol. Bold marks the best score per block.

	Code Check (%)	**Geometry Check (%)**					**Design Intent Alignment (%)**			
Model	Sandbox Success	Watertight	Manifold	Self-Int. Free	Overlap Free	*Geom. Valid*	**Functionality**	**Manufacturability**	**Assemblability**	*Final Score*
Closed-Source Models										
claude-opus-4.7	76.42	**76.42**	**76.42**	**73.58**	60.38	58.49	53.77	54.72	51.89	53.46
claude-3.7-sonnet	46.23	46.23	46.23	46.23	23.58	23.58	17.45	19.34	16.98	17.92
gemini-3.1-pro	65.09	59.43	63.21	63.21	62.26	58.49	53.77	55.19	52.83	53.93
gpt-5.5	**77.36**	72.64	74.53	**73.58**	**70.75**	**68.87**	**66.51**	**66.98**	**67.92**	**67.14**
gpt-4o	36.79	36.79	36.79	36.79	13.21	13.21	8.02	9.91	8.49	8.81
Average	60.38	58.30	59.43	58.68	46.04	44.53	39.91	41.23	39.62	40.25
Open-Source Models										
glm-5.1	31.13	31.13	31.13	31.13	**27.36**	**27.36**	**22.64**	**25.00**	**24.53**	**24.06**
glm-4.7-flash	8.49	7.55	7.55	7.55	5.66	5.66	2.83	2.83	2.83	2.83
minimax-m2.7	18.87	18.87	18.87	18.87	10.38	10.38	8.02	8.49	8.02	8.18
minimax-m2.5	25.47	22.64	22.64	22.64	10.38	10.38	4.72	6.60	4.72	5.35
qwen-3.5-122b-a10b	27.36	25.47	25.47	25.47	13.21	13.21	10.38	10.38	10.38	10.38
qwen-2.5-72b	**50.00**	**49.06**	**49.06**	**49.06**	19.81	20.75	8.02	14.15	8.96	10.38
llama-3.1-70b	31.13	30.19	30.19	30.19	22.64	23.58	3.77	8.96	3.77	5.50
qwen-3.6-35b-a3b	8.49	8.49	8.49	8.49	5.66	5.66	4.72	4.72	4.72	4.72
qwen-3.6-coder	14.15	14.15	14.15	14.15	10.38	10.38	4.72	6.13	5.19	5.35
llama-3.1-8b	2.83	1.89	1.89	1.89	0.94	1.89	0.00	0.00	0.00	0.00
Average	21.79	20.94	20.94	20.94	12.64	12.92	6.98	8.73	7.31	7.67

approximations can depend on orientation. Counts depend on the B-Rep representation, and one physical feature can contribute multiple faces or openings.

Normalization and aggregation. We pair generated and GT records by design identity, requiring successful extraction and at least one solid in each imported shape. This does not imply passing all benchmark geometric-validity checks. Let $x^{\mathrm{gen}}_{i,m,k}$ and $x^{\mathrm{GT}}_{i,k}$ denote descriptor k for design i, generated by model m and in GT, respectively. For feature counts, GT counts must be positive; generated zero counts are retained, and the per-design ratio is

$$r_{i,m,k} = \min\left(x^{\mathrm{gen}}_{i,m,k}/x^{\mathrm{GT}}_{i,k},\ 2\right). \tag{6}$$

For dimensions, both values must be positive, and the ratio is uncapped. Missing or unmeasurable dimensions are excluded rather than assigned zero. Dimension ratios compare within-design medians without matching individual features; the median depth-to-diameter ratio is not necessarily the ratio of median depth to median diameter. Across eligible designs, dots show the median ratio and light bars show its 25th–75th percentiles, computed by linear interpolation. These are interquartile ranges, not confidence intervals. We report eligible sample sizes n, which may differ across models and descriptors. Count panels use linear axes, whereas dimension panels use logarithmic axes with panel-specific ranges. A collapsed interval does not imply that all designs agree. Numerical agreement does not establish correct feature placement, specification compliance, or manufacturability.

Table 11: Hierarchy, definitions, and interpretation of local geometric descriptors. Dimension descriptors are summarized by their median within each design. Lengths are in millimeters; counts and depth-to-diameter ratios are dimensionless.

Family	Descriptor	Definition / extraction	Interpretation relative to GT
I. Local feature counts			
Holes	Hole count	Concave cylindrical faces with angular span $> 1.6\pi$.	Fewer or more detected hole-like faces.
Fillets	Fillet count	Other concave cylindrical faces plus toroidal faces.	Fewer or more fillet-like patches, not verified rounding operations.
Pockets	Pocket count	Non-circular inner loops on planar faces with estimated length/width ≤ 3.	Fewer or more pocket-like openings, not necessarily distinct cavities.
Slots	Slot count	Non-circular inner loops on planar faces with estimated length/width > 3.	Fewer or more elongated openings; their connection role is not assessed.
II. Local feature dimensions			
Holes	Diameter	Median diameter of detected hole-like cylindrical faces.	Smaller or larger characteristic diameter.
	Depth	Median axial extent of detected hole-like cylindrical faces.	Shorter or longer detected axial extent, not matched physical-hole depth.
	Depth/diameter	Median of the individual detected faces' depth-to-diameter ratios.	Different hole proportions; depth, diameter, or the detected feature set may change.
Pockets	Width	Median estimated shorter in-plane extent of pocket-like loops.	Narrower or wider detected openings.
	Depth	Median estimated adjacent-wall extent along the dominant face-normal axis.	Smaller or larger estimated recess depth.
Slots	Width	Median estimated shorter in-plane extent of slot-like loops.	Narrower or wider elongated openings.