P3D-BENCH: BENCHMARKING MLLMS FOR PARAMETRIC 3D GENERATION AND STRUCTURAL REASONING

Anonymous authors
Paper under double-blind review

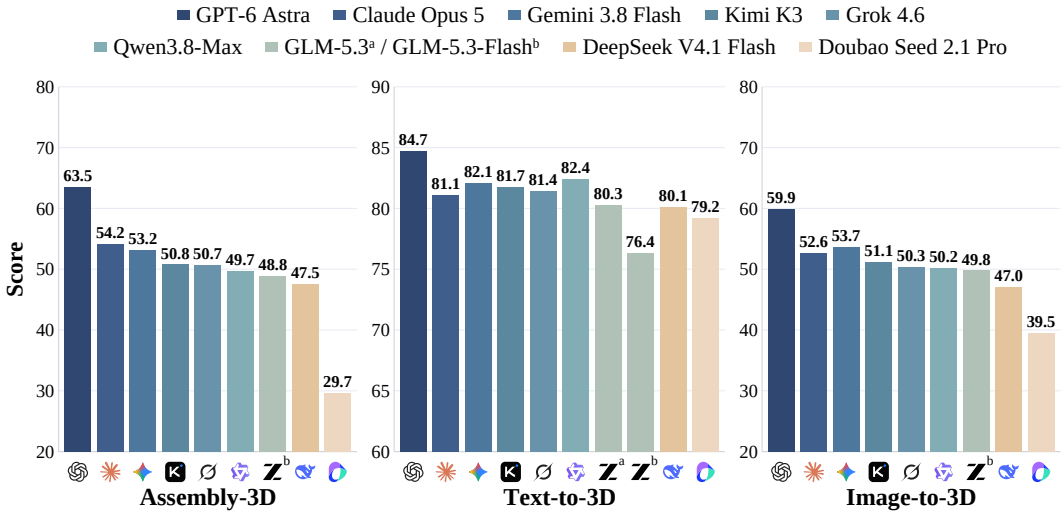


Figure 1: **Model scores across the three P3D-BENCH tasks.** Scores average the applicable Geo and Judge buckets, with Part additionally included for Assembly-3D. Per-bucket results are reported in Table 2.

ABSTRACT

Multimodal large language models can write code to produce complex programs as well as use programs to do 3D modeling, which opens up a new avenue for 3D generation powered by their priors, world knowledge and reasoning. Yet existing benchmarks rarely evaluate 3D modeling through code. Such modeling demands more than runnable code: from a text or visual specification, a model must generate a parametric 3D program that is geometrically precise, semantically aligned and assembly-consistent. We introduce P3D-BENCH, a benchmark for parametric 3D generation. Unlike a 3D mesh, a parametric 3D program exposes explicit dimensions, construction operations and part relations, revealing whether a model recovers a design's structure, not just its appearance. Under a unified protocol, P3D-BENCH covers three task families (Text-to-3D, Image-to-3D and Assembly-3D) and scores each output for executability, geometric fidelity, topology, text-grounded constraints, multiview semantic alignment and part-level structure. We construct P3D-Dataset, comprising 400 text cases, 400 image cases, and 203 annotated assemblies. Our evaluation on 100 cases from each task family yields three key findings. First, multi-part generation is substantially more challenging than single-part modeling, with models struggling to compose individual parts into a coherent structure. Second, models can often recover the global shape and semantic identity of the target object, yet fail to reproduce the precise parametric geometry specified by the input. Third, part-level modeling remains weak on assemblies, where models recover neither the geometry of each part nor the right number of parts. These results position P3D-BENCH as a benchmark for evaluating precise parametric geometry and part-level structure in parametric 3D generation.

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.

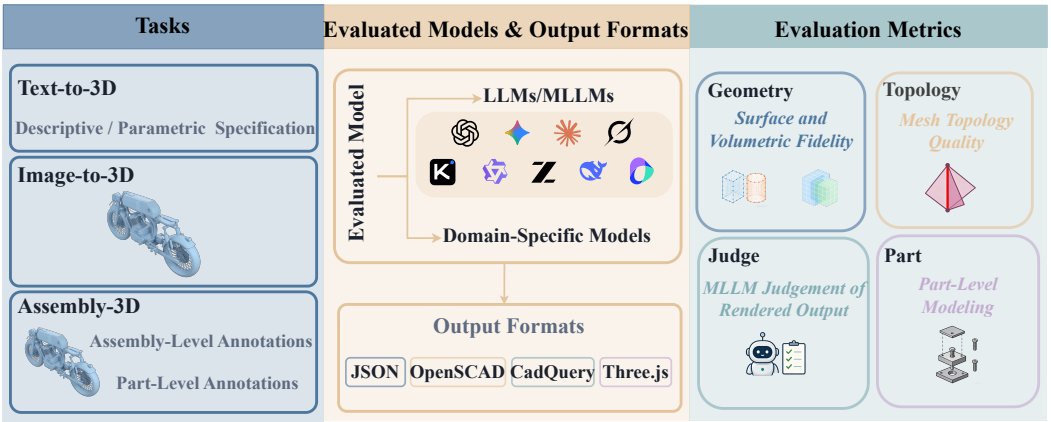


Figure 2: **Overview of the P3D-BENCH evaluation.** Given a task input—a text specification (descriptive or parametric), a single image or an image with assembly- and part-level annotations—a model (general-purpose LLM/MLLM or domain-specific) writes a program in one of the four output formats. P3D-BENCH executes and renders each program, reports validity (*Valid*) and scores it along the four evaluation dimensions defined in the *Evaluation protocol* section.

1 INTRODUCTION

Multimodal large language models (MLLMs) can write executable code and reason about the shapes and structures in images. Combining these abilities, current models can turn a text prompt or reference image into CadQuery or Three.js code that executes into the target 3D object. Unlike directly predicting a mesh, generating 3D models as programs gives an explicit, editable representation. The code records the dimensions, construction steps and decomposition into parts, and can be rerun with different parameters to produce a new valid model.

How well current models perform at parametric 3D generation remains an open question. Existing benchmarks emphasize complementary capabilities: code benchmarks check whether a program compiles and passes its tests (Chen et al., 2021; Jimenez et al., 2024); spatial benchmarks assess reasoning about layout and object relations (Wang et al., 2026a; Zhang et al., 2026b); text-to-3D benchmarks assess the shape's visual quality (He et al., 2023; Zhang et al., 2025). Parametric 3D generation requires these abilities jointly: the generated program must compile into valid geometry and, by inferring part structure and assembly relations, produce a result that is both visually plausible and geometrically precise.

We introduce P3D-BENCH, a benchmark that evaluates these properties of 3D modeling through code across three tasks. **Text-to-3D** provides a text description and requires a single part. **Image-to-3D** provides a single rendered image and requires a multi-part object, so the model must infer geometry and part layout not visible from one view. **Assembly-3D** adds assembly-level and part-level text annotations and requires the full assembly. In each task, the model produces a program in one of four formats (JSON (Khan et al., 2024), OpenSCAD (OpenSCAD Developers, 2021), CadQuery (CadQuery contributors, 2026) or Three.js (three.js authors, 2026)), which P3D-BENCH executes and renders; it reports executable validity and scores the model from four perspectives: geometry, topology, MLLM-based judgment and part-level structure (Figure 2).

Existing datasets are uneven in quality and lack the structured annotations these tasks require, so we build P3D-Dataset. Starting from Text2CAD v1.1 (Khan et al., 2024) and the Fusion 360 Gallery (Willis et al., 2022), we filter ambiguous and near-duplicate models, balance complexity levels, and generate and validate the task-specific text inputs: descriptive and parametric specifications for Text-to-3D, and assembly-level and part-level annotations for Assembly-3D. The resulting dataset comprises 400 text cases, 400 image cases and 203 annotated assemblies.

In summary, P3D-BENCH provides a unified benchmark for parametric 3D generation and structural reasoning from text and image specifications. Extensive evaluations of frontier MLLMs, text-only LLMs and domain-specific models reveal that: (1) building a full assembly is substantially harder

Table 1: **Comparison of representative benchmarks.** Task describes the kind of task evaluated. Model Types indicates whether the benchmark evaluates general-purpose LLMs/MLLMs and domain-specific models (Domain Spec.). Under Evaluation, Exec. means that generated outputs are executed as 3D geometry; Param. means that scoring explicitly checks dimensional or parametric accuracy against the specification or ground truth; Spatial means that evaluation includes the relative placement or layout of objects or parts; Asm. means that the benchmark contains multi-part assemblies; and Part means explicit geometric matching of predicted and ground-truth parts.

		Model Types		Evaluation				
Benchmark	Task	LLM/ MLLM	Domain Spec.	Exec.	Param.	Spatial	Asm.	Part
SWE-bench	Code fix	✓	✗	✗	✗	✗	✗	✗
InfiniBench	Spatial reasoning	✓	✗	✗	✗	✓	✗	✗
Theory of Space	Spatial reasoning	✓	✗	✗	✗	✓	✗	✗
T^3Bench	Mesh gen.	✗	✓	✗	✗	✗	✗	✗
3DGen-Bench	Mesh gen.	✗	✓	✗	✗	✗	✗	✗
VoxelCodeBench	Procedural 3D gen.	✓	✗	✓	✗	✓	✗	✗
3DCodeBench	Procedural 3D gen.	✓	✗	✓	✗	✗	✗	✗
Text2CAD-Bench	CAD gen. (CadQuery)	✓	✓	✓	✗	✗	✗	✗
UniCAD	CAD gen. (CadQuery)	✓	✓	✓	✗	✓	✗	✗
BenchCAD	CAD gen. (CadQuery)	✓	✓	✓	✓	✗	✗	✗
MUSE	CAD gen. (CadQuery)	✓	✗	✓	✗	✓	✓	✗
P3D-BENCH (Ours)	**CAD gen. (JSON, OpenSCAD, CadQuery, Three.js)**	✓	✓	✓	✓	✓	✓	✓

than building a single part because it requires correct spatial relations and part structure. (2) geometric alignment with the input is harder than semantic alignment: the strongest MLLM already aligns well with the input semantically (*J-Sem* ≈ 0.7), but its geometry matches the specified dimensions less accurately (*J-Geo* ≈ 0.5). (3) on assembly tasks, beyond errors in part placement, models also struggle to recover the correct number and geometry of parts: matching predicted parts against ground-truth parts, the strongest MLLM reaches a *PartMatchF1* of about 0.5. We make four contributions.

- **A unified benchmark.** Three tasks (Text-to-3D, Image-to-3D and Assembly-3D) and four code formats (JSON, OpenSCAD, CadQuery, Three.js) under a unified protocol that executes and renders every output, enabling comparison across input specifications and target formats.
- **A new dataset and its pipeline.** A filtering and annotation pipeline yielding P3D-Dataset: 400 text cases, 400 image cases and 203 assemblies with task-specific text inputs.
- **A structured evaluation protocol.** Beyond validity, four scores (*Geo*, *Topo*, *Judge*, *Part*) covering geometric fidelity, topology, MLLM-based assessment and part structure.
- **An evaluation of frontier models.** A unified evaluation of frontier MLLMs, text-only LLMs and domain-specific models, revealing that programs which execute and render plausibly still fail to recover correct parametric geometry.

2 RELATED WORK

Visual and parametric 3D generation. Most 3D generation produces visual geometry. DreamFusion (Poole et al., 2023) and LRM (Hong et al., 2024) build on the NeRF representation (Mildenhall et al., 2020) to recover a radiance field from text or a single image, while native 3D diffusion transformers (Wu et al., 2024; 2025) build on neural signed distance fields (Park et al., 2019) to generate meshes directly. Unlike these appearance-only outputs, parametric 3D generation targets a representation defined by its construction logic, making operations, dimensions and part structure explicit and editable. Early models are trained to generate sketch-extrude sequences directly (Wu et al., 2021; Xu et al., 2022; 2023; Khan et al., 2024). Later work fine-tunes code LLMs and VLMs to produce executable CAD code (Doris et al., 2025; Xie & Ju, 2025; Govindarajan et al., 2026; Kolodiazhnyi et al., 2026). Recent agentic systems clarify ambiguous prompts before generation (Yuan et al., 2026) or revise CAD code under geometric and visual feedback (Barkley et al., 2026); ArtiCAD and ArtiCraft extend agentic generation to articulated assemblies with explicit joints (Shui

et al., 2026; Zhou et al., 2026). A parallel line of work generates Blender Python instead of CAD code, with agents that write and iteratively repair the script from text (Lu et al., 2025) or a single image (Yin et al., 2026) under rendered feedback. Across these threads, parametric 3D is increasingly produced by general-purpose models writing code, motivating a unified evaluation of general-purpose LLMs/MLLMs and domain-specific models across tasks and code formats.

Benchmarks related to parametric 3D generation. Recent benchmarks for code-based and parametric 3D generation differ in their target representations, supported tasks and evaluation criteria. Code-based 3D benchmarks operate on generated programs: VoxelCodeBench (Zheng & Bordes, 2026) evaluates code against an Unreal voxel API, and 3DCodeBench (Gao et al., 2026) scores agentic Blender Python by executability, render similarity, human preference and geometric similarity. Both verify that the 3D program runs, but neither reports dedicated metrics for numerical dimensional accuracy or geometric part matching. Another line evaluates executable CAD code directly. Text2CAD-Bench (Wang et al., 2026b) focuses on text-to-CadQuery generation, while UniCAD (Chen et al., 2026) covers multiple input modalities and BenchCAD (Zhang et al., 2026a) also evaluates editing tasks. Text2CAD-Bench measures validity, Chamfer distance and IoU on its L1–L3 levels and uses a VLM to assess multiview renders on L4; BenchCAD also tests numerical CAD questions. These benchmarks use CadQuery and focus on single-body models rather than explicit geometric matching of assembly parts. MUSE (Dong et al., 2026) is the most related: it checks whether generated text-to-CadQuery assemblies are functional and assemblable. Its assembly structure, however, is judged by a VLM from rendered images. Table 1 summarizes these differences: unlike prior benchmarks, P3D-BENCH evaluates each case across multiple formats and includes multi-part assemblies with explicit part-level modeling.

3 P3D-BENCH: TASKS, DATASET AND EVALUATION

3.1 TASK DEFINITION

We formulate parametric 3D generation as code-based reconstruction from a task condition c—a text description, one rendered image or the image together with assembly-level and part-level text. Given an input c and a target format ϕ, a policy π generates a program $f_\pi = \pi(c, \phi)$, where ϕ is one of the four formats supported by P3D-BENCH: minimal JSON, OpenSCAD, CadQuery or Three.js. A format-specific deterministic execution operator $\mathcal{E}_\phi$ then compiles, executes and renders the resulting program to produce a 3D output $y_\pi = \mathcal{E}_\phi(f_\pi)$. Accordingly, we represent each evaluation case as a triplet $x = (c, f^\star, y^\star)$, where $f^\star$ is the source program and $y^\star$ is its target 3D output, including ground-truth geometry and, when available, part structure.

3.2 DATASET CONSTRUCTION PIPELINE

To evaluate the P3D-BENCH tasks, we construct P3D-Dataset from two open CAD sources with complementary geometry: Text2CAD v1.1 (Khan et al., 2024), which contains $176{,}017$ single-part sketch–extrude programs for Text-to-3D, and the Fusion 360 Gallery (Willis et al., 2022), which contains $8{,}251$ multi-part assemblies for Image-to-3D and Assembly-3D. Neither source is directly usable. Text2CAD includes unevaluable, simple and near-duplicate records, and does not distinguish descriptive from parametric text specifications. Fusion 360 needs screening for visual clarity and reconstructability; it also lacks the assembly-level and part-level text required by Assembly-3D. We therefore design a two-stage pipeline. The filtering stage removes unreliable and redundant cases and balances the retained set across category and complexity. The annotation stage supplies the task-specific text inputs and keeps them consistent with the source geometry and render. Appendix A summarizes the process and final splits.

Filtering Pipeline. Filtering constructs a balanced subset where failures reflect model limitations rather than broken sources, ambiguous inputs or repeated shapes. We retain cases that are executable, renderable, visually interpretable and non-redundant, while balancing complexity across the dataset. Specifically, deterministic checks first remove unevaluable records; a review MLLM then rejects cases that are visually ambiguous, not reconstructable from the task input or poorly oriented, while assigning category and complexity labels; and DINOv2 (Oquab et al., 2024) feature matching over render embeddings removes near-duplicate cases. Finally, complexity-balanced sampling draws the

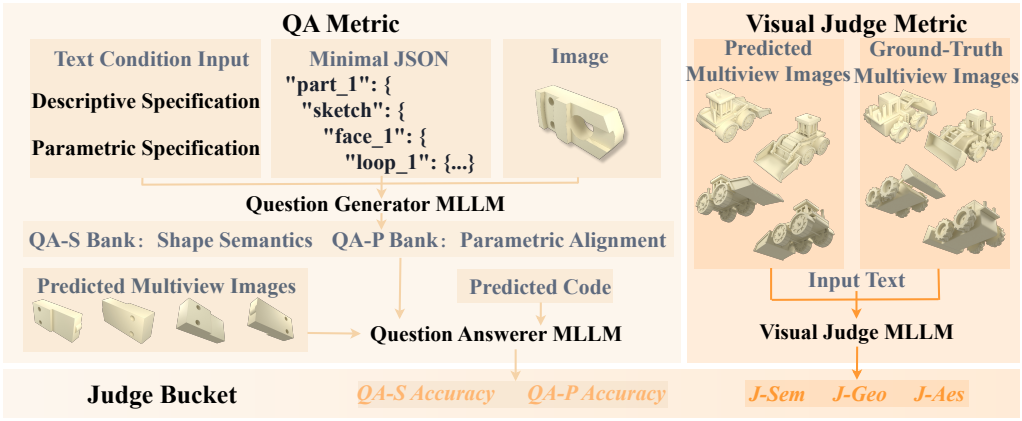


Figure 3: **Overview of the *Judge* bucket metrics.** The QA metric builds a QA-S bank from the descriptive specification and a QA-P bank from the parametric specification, then scores the prediction by its accuracy on those questions. The visual Judge rates the prediction from the predicted and ground-truth multiview images and, when available, the input text, along three axes: semantic similarity (*J-Sem*), geometric similarity (*J-Geo*) and aesthetic quality (*J-Aes*).

retained cases across semantic categories and easy, medium and hard tiers, yielding 400 Text-to-3D cases and 400 Image-to-3D cases.

Annotation Pipeline. Annotation converts retained geometry into the task-specific inputs absent from the raw sources. (1) *Text-to-3D annotation.* For each Text2CAD part, an annotation MLLM writes two specifications from the source geometry and render. The descriptive specification summarizes shape, salient features and function without numerical parameters, testing semantic shape construction from natural language. The parametric specification expresses dimensions, counts, offsets and placements in natural language, testing whether models can translate explicit parameters into executable geometry. (2) *Image-to-3D annotation.* For each Fusion 360 assembly, the annotation MLLM labels each part after geometric deduplication (assemblies with more than 20 unique parts are dropped), then combines these part-level annotations with the assembly's spatial layout to compose an assembly-level description; a verification MLLM checks both for consistency and render alignment. Only verified assemblies are retained for Assembly-3D, yielding 203 annotated cases.

3.3 Evaluation Protocol

Executable parametric 3D generation can fail at several levels: a program may not execute, the resulting shape may be geometrically inaccurate, the generated mesh may have poor topology, the render may miss semantic or parametric constraints, and an assembly may use the wrong parts despite a plausible overall shape. P3D-Bench therefore executes each generated program and exports the result to a mesh. Before scoring, we apply task-specific normalization to the predicted and ground-truth meshes, then align the prediction to the ground truth.

We measure executable validity (*Valid*) and group the quality metrics into *Geometry*, *Topology*, *Judge* and, for Assembly-3D, *Part*. Each group combines its applicable sub-metrics into a *bucket score* normalized to 0–100, where higher is better. The preprocessing and score aggregation procedures are detailed in Appendix C.

Geometry and Topology Metrics. The *Geometry* metrics measure how closely an executable prediction matches the ground truth after alignment, using complementary surface and volume criteria: Chamfer Distance (CD↓), F-score at the fine threshold (F@0.01↑), normal consistency (NC↑) and IoU (computed per task; see Appendix C). To measure topological quality of meshes, we report the no-open-edge score (NoOE↑), inverted-normal ratio (InvN↓) and non-manifold-edge ratio (NM↓).

MLLM Judge Metrics. Since geometry and topology metrics do not capture semantic and parametric correctness, we introduce an MLLM-based *Judge* bucket (Gemini 3.8 Flash (Google DeepMind,

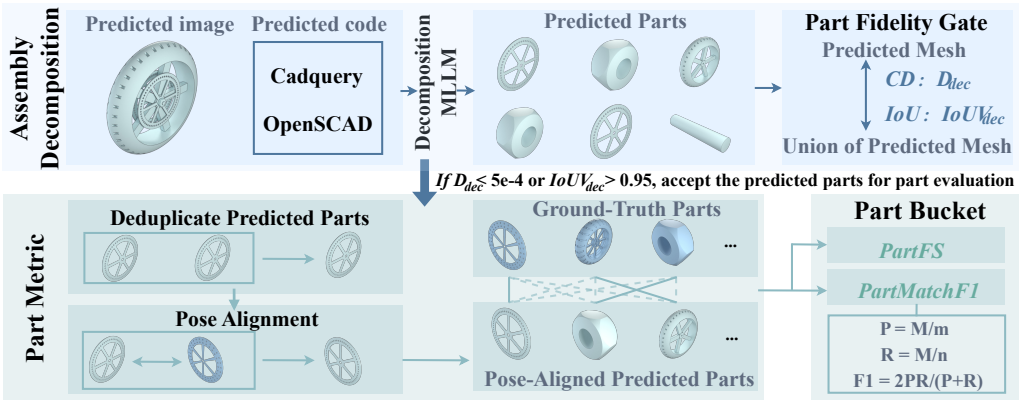


Figure 4: **Overview of the Assembly-3D *Part* bucket metrics.** *Assembly Decomposition*: a fixed decomposition MLLM splits the predicted assembly into parts, and a fidelity gate keeps only those whose reassembled union matches the original prediction. *Part Metric*: the retained parts are deduplicated, pose-aligned to GT parts and matched one-to-one to compute the scores, where M counts the successful matches and m, n count the predicted and GT parts.

2026)) with two components (Figure 3). The QA metric tests the prediction using questions derived from the textual specification: QA-S targets semantic constraints in the descriptive setting, while QA-P targets explicit dimensions, counts, placements and other parametric constraints in the parametric setting. The Visual Judge metric compares predicted and ground-truth multiview images along three dimensions: *J-Sem*, *J-Geo* and *J-Aes*; the input text specification, when available, is also given to the judge. *J-Sem* evaluates object identity and semantic consistency, *J-Geo* compares shape, proportions and alignment, and *J-Aes* rates visual quality and detail richness. We validate both components against human annotators on the same QA-S items and three axes (Table 3b).

Assembly-3D Part Metrics. The *Part* metrics evaluate per-part modeling ability: whether the predicted parts match the ground-truth parts in both shape and count. Since evaluated models emit a single whole-assembly program, we use a fixed decomposition MLLM, shared across all models, to split each valid predicted assembly into per-part programs (Figure 4). A fidelity gate then checks that the decomposed parts reassemble to the original prediction before any per-part scores are computed; cases that fail the gate are excluded rather than scored. Appendix D.4 reports per-model decomposition fidelity and gate-exclusion counts.

After the fidelity gate, we deduplicate the retained predicted parts, align their poses to the ground-truth parts and match them using Hungarian matching (Kuhn, 1955) on the pairwise part F-scores. Let $\mathcal{H}$ denote the set of one-to-one pairs returned by Hungarian matching. For a prediction with m distinct parts and a ground-truth assembly with n distinct parts, $|\mathcal{H}| = \min(m, n)$. We then compute two part-level scores, PartFS and PartMatchF1:

$$\text{PartFS} = \frac{1}{\min(m,n)} \sum_{(p,g)\in\mathcal{H}} F_{\text{part}}(p, g), \qquad \text{PartMatchF1} = \frac{2\,P\,R}{P+R}. \tag{1}$$

PartFS is the mean part F-score over all pairs in $\mathcal{H}$, measuring the geometric fidelity of matched parts. A pair is considered a *successful match* if its part F-score is at least the acceptance threshold. Let M denote the number of such successful matches. PartMatchF1 is then the F1 score of part precision $P = M/m$ (PartMatchP) and part recall $R = M/n$ (PartMatchR).

4 Experiments

Evaluation subsets and models. We evaluate two model groups on P3D-Bench using a shared subset of 100 cases per task. The first comprises ten general-purpose models: one text-only LLM and nine MLLMs. All ten are evaluated on Text-to-3D, while the MLLMs are also evaluated on Image-to-3D and Assembly-3D. The second comprises three domain-specific parametric CAD baselines: Text2CAD (Khan et al., 2024), Cadrille (Kolodiazhnyi et al., 2026), and CAD-Coder (Doris

Table 2: **P3D-BENCH main results, with bucket scores averaged across output formats.** *Geo*, *Topo*, *Judge* and *Part* measure geometry, topology, MLLM judgment and assembly part quality; *Valid* is executable validity (%). Each model is evaluated only on its applicable tasks; "–" denotes an unevaluated task–model pair. Best and second-best general-purpose models are bold and underlined, respectively, for *Geo*, *Judge* and *Part*.

	Assembly-3D					Text-to-3D						Image-to-3D			
						Descriptive		Parametric							
Model	Geo	Judge	Part	Topo	Valid	Judge	Valid	Geo	Judge	Topo	Valid	Geo	Judge	Topo	Valid
GPT-6 Astra	**68.2**	**59.9**	**62.3**	95.8	99.4%	**92.6**	100%	68.8	**92.8**	99.3	100%	**58.9**	**60.9**	92.2	99.6%
Claude Opus 5	56.3	52.0	54.3	98.5	100%	84.6	100%	67.6	91.2	99.5	100%	52.1	53.1	92.4	100%
Gemini 3.8 Flash	56.1	52.9	50.5	97.4	99%	86.6	100%	**69.2**	90.5	99.3	100%	53.1	54.3	89.0	99%
Grok 4.6	55.0	49.6	47.5	97.6	100%	85.9	100%	68.7	89.6	99.5	100%	50.0	50.6	90.5	99%
Kimi K3	52.8	48.3	51.2	94.9	99%	86.9	98.5%	67.0	91.3	98.8	99.5%	50.7	51.5	92.5	100%
Qwen3.8-Max	52.3	47.3	49.6	95.9	98.9%	88.6	100%	67.9	90.7	99.0	100%	50.3	50.2	94.4	99.7%
GLM-5.3	–	–	–	–	–	85.2	99.5%	65.6	90.0	99.5	100%	–	–	–	–
GLM-5.3-Flash	52.6	48.1	45.9	96.2	98.9%	81.1	97.5%	60.8	87.2	97.6	98.5%	49.7	49.9	92.1	99.3%
DeepSeek V4.1 Flash	50.6	46.3	45.6	95.1	97.9%	86.2	100%	65.5	88.6	98.3	99%	47.4	46.6	91.2	97%
Doubao Seed 2.1 Pro	32.8	27.1	29.2	68.6	69.5%	84.2	99.5%	64.6	88.7	98.8	99.5%	40.6	38.4	76.5	84.7%
Text2CAD	–	–	–	–	–	12.9	91%	29.4	13.4	97.7	98%	–	–	–	–
CADRILLE	–	–	–	–	–	–	–	–	–	–	–	21.7	16.6	74.5	77%
CAD-CODER	–	–	–	–	–	–	–	–	–	–	–	7.1	4.3	20.3	20.7%

et al., 2025). We evaluate their released checkpoints under the I/O settings in Table A1. Cases are randomly sampled for Text-to-3D and Assembly-3D; for Image-to-3D, we select the 100 highest-complexity cases to avoid saturation on simple inputs. All general-purpose models use their highest supported reasoning level. On Image-to-3D and Assembly-3D, general-purpose models may correct invalid outputs; we report the final outputs. Invalid outputs remain in the evaluation denominators and receive zero bucket scores. Table A1 lists all models, parameter counts, and baseline I/O contracts.

Output formats per task. Each of the three P3D-BENCH tasks is evaluated on the subset of the four output formats (minimal JSON, OpenSCAD, CadQuery and Three.js) appropriate for its inputs and targets, giving seven task–format combinations in total. Text-to-3D uses minimal JSON and OpenSCAD: minimal JSON matches the source construction format, while OpenSCAD is a higher-level, more readable CSG language. Image-to-3D uses OpenSCAD, CadQuery and Three.js, dropping minimal JSON, which is too limited to express multi-part assemblies. Assembly-3D uses only OpenSCAD and CadQuery, dropping Three.js as well, since its triangulated meshes are hard to decompose into per-part solids. The per-format results below support these choices.

4.1 MAIN RESULTS

Table 2 reports the cross-format average bucket scores of all evaluated models on the three P3D-BENCH tasks; all submetrics are reported in Appendix E. Figure 5 shows qualitative reconstructions for five representative general-purpose models plus the applicable domain-specific baseline across the three tasks; a larger gallery with additional cases is in Figure A7.

Across models and tasks. GPT-6 Astra achieves the highest mean Score on all three tasks. On Text-to-3D, the remaining models are tightly clustered: seven score between 80.1 and 82.4, while Doubao Seed 2.1 Pro (79.2) and GLM-5.3-Flash (76.4) fall noticeably behind. On Image-to-3D and Assembly-3D, Claude Opus 5 and Gemini 3.8 Flash form a second tier behind GPT-6 Astra; Kimi K3, Grok 4.6, Qwen3.8-Max and GLM-5.3-Flash occupy the middle, while DeepSeek V4.1 Flash and especially Doubao Seed 2.1 Pro score markedly lower. Domain-specific models fall behind even on their native tasks and formats. Scores are lower and more dispersed on the two assembly tasks than on the single-part task (Table 2, Figure 1). This pattern is consistent with the additional challenges of assembly reconstruction, although the tasks differ in inputs, cases and score components.

Across output formats and evaluation metrics. To compare the four output formats, Figure 6 reports Gemini 3.8 Flash across all four formats on every task. For this model, OpenSCAD offers a

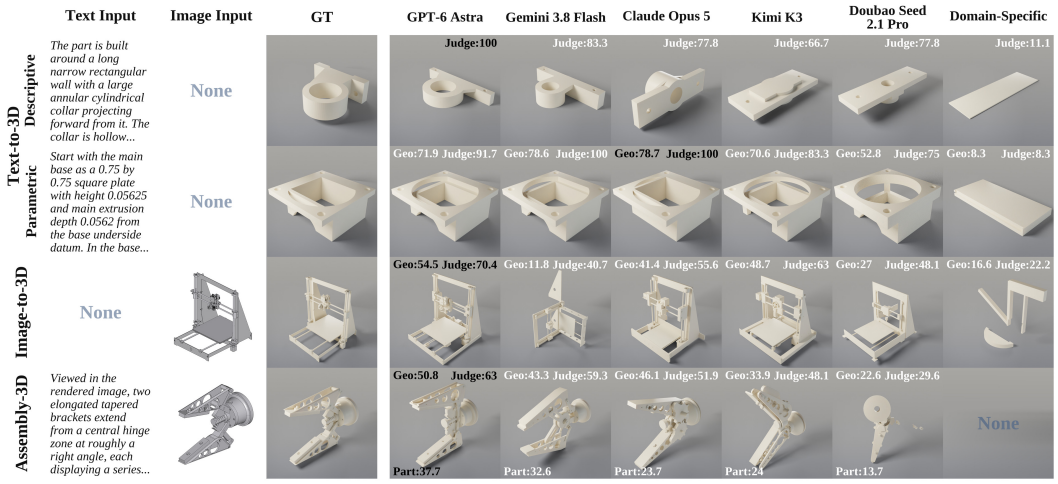


Figure 5: **Qualitative reconstructions across the three P3D-BENCH tasks.** Rows are tasks, with Text-to-3D split into descriptive and parametric. Five representative general-purpose models produce OpenSCAD, while Text2CAD on Text-to-3D and CADRILLE on Image-to-3D are shown in their native formats. Each output is annotated with its per-case bucket scores, with the best in each row in black. "None" marks an inapplicable input or baseline, and "Part: –" a prediction that failed the *Part* fidelity gate.

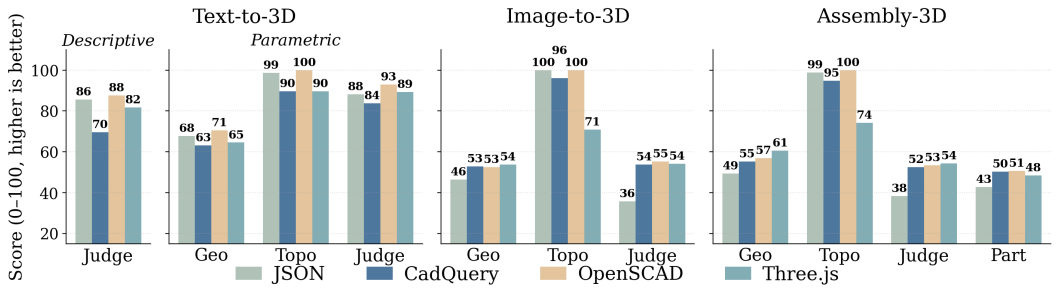


Figure 6: **Gemini 3.8 Flash bucket scores across all tasks and output formats.** Text-to-3D is split into descriptive and parametric.

strong balance across the applicable buckets, with no clear weakness. JSON, in contrast, is clearly the weakest on the assembly tasks, confirming our earlier analysis.

CadQuery and Three.js are close to OpenSCAD on the overall *Geo* and *Judge* buckets, but fall behind for different reasons. Models often produce invalid CadQuery programs that fail to run, especially weaker models (per-format *Valid* in Appendix E), thereby lowering their scores across all metrics. Three.js instead has no solid kernel and builds triangulated surfaces, so it scores poorly on *Topo* and *Part*: such meshes are not guaranteed watertight and do not yield clean per-part solids for matching.

4.2 RESULT DETAILS

The headline *Judge* and *Part* buckets in Table 2 each average several submetrics into a single score. In this section, we unpack both buckets and analyze their submetrics.

Judge bucket details. Table 3a reports Gemini 3.8 Flash's Judge submetrics using the same task-specific format averages as Table 2. **(i)** QA-S is lower under parametric than descriptive specifications (0.89 vs. 0.98), while parametric QA-P reaches 0.91. These results highlight that the remaining challenges in parametric generation extend beyond numerical accuracy to satisfying semantic constraints. **(ii)** On Image-to-3D and Assembly-3D, *J-Sem* reaches 0.65 and 0.64, respectively, while *J-Geo* rounds to 0.43 on both tasks. For Gemini 3.8 Flash, these ratings suggest stronger agreement on

Table 3: **The *Judge* bucket in detail. (a)** Gemini 3.8 Flash's submetrics averaged over the main-experiment formats, with Text-to-3D split into descriptive and parametric; "–" marks an inapplicable submetric, and qualitative cases are shown in Figure A4. **(b)** Agreement with human annotators: for Text-to-3D, the rate at which an annotator and the MLLM Judge select the same multiple-choice option; for Image-to-3D and Assembly-3D, Spearman's ρ against the mean human rating (95% CI).

(a) Judge submetrics

Task	QA-S	QA-P	J-Geo	J-Aes	J-Sem
Text-to-3D (Desc.)	0.98	–	–	–	0.75
Text-to-3D (Param.)	0.89	0.91	–	–	–
Image-to-3D	–	–	0.43	0.54	0.65
Assembly-3D	–	–	0.43	0.52	0.64

(b) Judge vs. human annotators

Task	QA-S agreement
Text-to-3D	68.3%
Task	**Spearman ρ [95% CI]**
Image-to-3D	0.924 [.70, .98]
Assembly-3D	0.921 [.69, .98]

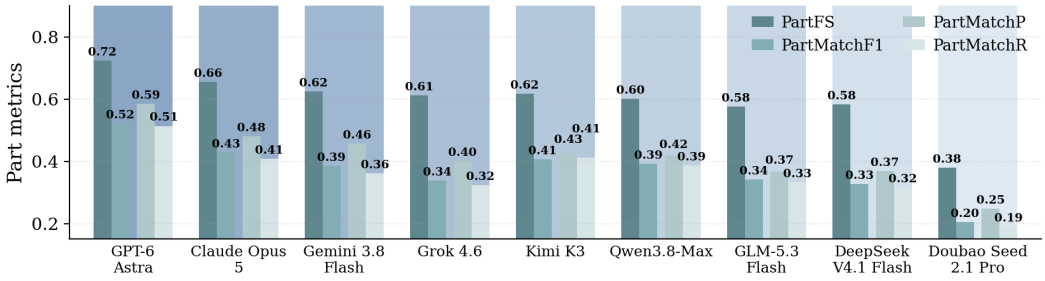


Figure 7: **Part metrics on Assembly-3D.** The *Part* bucket submetrics PartFS and PartMatchF1, with PartMatchP and PartMatchR, averaged across CadQuery and OpenSCAD. Qualitative cases are shown in Figure A5.

object identity than on precise shape and proportions. Table 3b shows overall consistency between the MLLM Judge and human judgments: QA-S agreement is 68.3% on predictions (human accuracy on ground truth: 80.0%), while visual scores correlate strongly with mean human ratings on Image-to-3D and Assembly-3D (Spearman $\rho \geq 0.92$); see Appendix D.1 for study details.

Part metric details. Figure 7 reports the Part metrics on Assembly-3D—PartFS and PartMatchF1, with PartMatchP and PartMatchR—averaged across CadQuery and OpenSCAD. **(i)** PartFS, per-part geometric fidelity of matched parts, reaches only ≈ 0.72 for the strongest model, GPT-6 Astra, and drops sharply for weaker ones. This is consistent with the Judge submetrics on the same task (Table 3a), where semantic alignment (*J-Sem*) exceeds geometric alignment (*J-Geo*): models capture overall shape semantics more reliably than reconstruct fine-grained per-part geometry. **(ii)** PartMatchF1 reaches only ≈ 0.52 even for the best model, with PartMatchP and PartMatchR both near 0.59 and 0.51: only about half of the predicted and ground-truth parts are successfully matched. Their ratio further reveals a mismatch between the predicted and ground-truth part counts. Current models thus recover neither the part count nor the per-part geometry of the assembly.

5 Conclusion

We introduced P3D-Bench, a unified benchmark for executable parametric 3D generation across text, image and assembly tasks. P3D-Bench clearly distinguishes models with different levels of parametric 3D generation capability. Across tasks and output formats, the main bottleneck is not producing runnable and visually plausible outputs, but translating specifications into precise geometry and part structure. A current limitation is that, while the appendix includes a preliminary multi-turn experiment (Table A5), P3D-Bench does not yet systematically evaluate general-purpose coding agents such as Codex and Claude Code; extending it to such agents, which execute and revise their programs over multiple turns, is left to future work.

AI USE STATEMENT

Generative AI was used in this work in three roles. **As the object of study.** The models listed in Table A1 are evaluated by P3D-BENCH under the protocol described in the paper and Appendix C. **As part of the benchmark pipeline.** MLLMs were used for three purposes: filtering candidate CAD models and generating and checking task-specific text annotations from source geometry and renders; scoring model outputs for the *Judge* metrics; and decomposing predicted assemblies into per-part programs for the *Part* metrics. These uses are described in Section 3.2 and Appendices A and C. **As a research and writing assistant.** LLM-based assistants supported code implementation and debugging, experiment execution and organization, data processing and plotting, retrieval of related work, and drafting and language editing of the manuscript. The benchmark methodology, experimental design and interpretation of the results were developed by the authors. The authors reviewed AI-assisted code and analysis outputs against the experimental records and reviewed and revised AI-assisted manuscript text. We take responsibility for the final content of this work, including its text, claims and artifacts.

ETHICS STATEMENT

P3D-BENCH is constructed from two publicly released CAD datasets: Text2CAD v1.1 (Khan et al., 2024), which provides single-part CAD models, and the Fusion 360 Gallery (Willis et al., 2022), which provides multi-part assemblies. We use and process both datasets in accordance with their respective licenses and redistribution terms. The commercial models evaluated in this work were used in accordance with the applicable service terms and model-provider usage policies. Human ratings used to validate the MLLM-based *Judge* metrics were provided by members of our research group as an internal methodological check. No external participants were recruited, and no personal or sensitive information was collected from the raters.

REPRODUCIBILITY STATEMENT

The paper and appendix document the main components and procedures of P3D-BENCH. Section 3.2 and Appendix A describe the data sources, preprocessing, filtering, deduplication, annotation and evaluation subsets. Section 3.3 and Appendix C specify the evaluation metrics and protocol, including alignment, normalization, aggregation, MLLM-based evaluation and part matching. Section 4 and Appendix B describe the evaluated models, reasoning setting, baseline input–output settings, output formats and execution environment. Per-format submetric results are reported in Appendix E.

REFERENCES

Alibaba Cloud. Qwen3.8-Max model documentation, 2026. URL https://help.aliyun.com/en/model-studio/qwen3-8-max.

Anthropic. Introducing Claude Opus 5, 2026. URL https://www.anthropic.com/news/claude-opus-5.

Jesse Barkley, Rumi Loghmani, and Amir Barati Farimani. CADSmith: Multi-agent CAD generation with programmatic geometric validation, 2026. URL https://arxiv.org/abs/2603.26512.

ByteDance Seed. Seed2.1 officially released: Advancing AI productivity, 2026. URL https://seed.bytedance.com/en/blog/seed2-1-officially-released-advancing-ai-productivity.

CadQuery contributors. CadQuery: A Python parametric CAD scripting framework based on OCCT, 2026. URL https://doi.org/10.5281/zenodo.18633916. Version 2.7.0; Apache-2.0 license; project page: https://github.com/CadQuery/cadquery.

Jingyuan Chen, Sheng Jin, Haopeng Sun, Wentao Liu, and Chen Qian. UniCAD: A unified benchmark and universal model for multi-modal multi-task CAD, 2026. URL https://arxiv.org/abs/2606.05058.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4.1-Flash.

Xiaoyu Dong, Zhi Li, and Xiao-Ming Wu. MUSE: Benchmarking manufacturable, functional, and assemblable text-to-CAD generation, 2026. URL https://arxiv.org/abs/2605.28579.

Anna C. Doris, Md Ferdous Alam, Amin Heyrani Nobari, and Faez Ahmed. CAD-Coder: An open-source vision-language model for computer-aided design code generation, 2025. URL https://arxiv.org/abs/2505.14646.

Yipeng Gao, Lei Shu, Genzhi Ye, Xi Xiong, Ameesh Makadia, Meiqi Guo, Laurent Itti, and Jindong Chen. 3DCodeBench: Benchmarking agentic procedural 3D modeling via code, 2026. URL https://arxiv.org/abs/2606.01057.

Google DeepMind. Gemini 3.8 Flash model card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-8-flash/.

Prashant Govindarajan, Davide Baldelli, Jay Pathak, Quentin Fournier, and Sarath Chandar. CADmium: Fine-tuning code language models for text-driven sequential CAD design. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=lExqWvQht8.

Yuze He, Yushi Bai, Matthieu Lin, Wang Zhao, Yubin Hu, Jenny Sheng, Ran Yi, Juanzi Li, and Yong-Jin Liu. T^3Bench: Benchmarking current progress in text-to-3D generation, 2023. URL https://arxiv.org/abs/2310.02977.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. In *The Twelfth International Conference on Learning Representations*, 2024.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations*, 2024.

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, 2024.

Maksim Kolodiazhnyi, Denis Tarasov, Dmitrii Zhemchuzhnikov, Alexander Nikulin, Ilya Zisman, Anna Vorontsova, Anton Konushin, Vladislav Kurenkov, and Danila Rukhovich. cadrille: Multi-modal CAD reconstruction with reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

Sining Lu, Guan Chen, Nam Anh Dinh, Itai Lang, Ari Holtzman, and Rana Hanocka. LL3M: Large language 3D modelers, 2025. URL https://arxiv.org/abs/2508.08228.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV 2020*, pp. 405–421. Springer, 2020.

Moonshot AI. Kimi K3 model card, 2026. URL https://huggingface.co/moonshotai/Kimi-K3.

OpenAI. GPT-6 Astra system card, 2026. URL https://deploymentsafety.openai.com/gpt-6-astra.

OpenSCAD Developers. OpenSCAD: The programmers solid 3D CAD modeller, 2021. URL https://github.com/openscad/openscad. Version 2021.01; GPL-2.0-or-later license; official website: https://openscad.org.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

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.

SpaceXAI. Introducing Grok 4.6, 2026. URL https://x.ai/news/grok-4-6.

three.js authors. three.js: JavaScript 3D library, 2026. URL https://github.com/mrdoob/three.js. Version 0.184.0; MIT license; official website: https://threejs.org.

Haoming Wang, Qiyao Xue, and Wei Gao. InfiniBench: Infinite benchmarking for visual spatial reasoning with customizable scene complexity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21594–21605, 2026a.

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, 2026b. URL https://arxiv.org/abs/2605.18430.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15849–15860, June 2022.

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*, pp. 6772–6782, 2021.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3D: Scalable image-to-3D generation via 3D latent diffusion transformer. In *Advances in Neural Information Processing Systems*, volume 37, pp. 121859–121881, 2024.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, and Yao Yao. Direct3D-S2: Gigascale 3D generation made easy with spatial sparse attention. In *Advances in Neural Information Processing Systems*, volume 38, pp. 170778–170804, 2025.

Haoyang Xie and Feng Ju. Text-to-CadQuery: A new paradigm for CAD generation with scalable large model capabilities, 2025. URL https://arxiv.org/abs/2505.06507.

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 *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24698–24724. PMLR, 2022. URL https://proceedings.mlr.press/v162/xu22k.html.

Xiang Xu, Pradeep Kumar Jayaraman, Joseph George Lambourne, Karl D.D. Willis, and Yasutaka Furukawa. Hierarchical neural coding for controllable CAD model generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38443–38461. PMLR, 2023. URL https://proceedings.mlr.press/v202/xu23f.html.

Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Chenyang Wang, Xiuyu Li, Michael J Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. Vision-as-inverse-graphics agent via interleaved multimodal reasoning, 2026. URL https://arxiv.org/abs/2601.11109.

Bo Yuan, Zelin Zhao, Petr Molodyk, Bin Hu, and Yongxin Chen. Clarify before you draw: Proactive agents for robust text-to-CAD generation, 2026. URL https://arxiv.org/abs/2602.03045.

Z.AI. GLM-5.3 model card, 2026a. URL https://huggingface.co/zai-org/GLM-5.3.

Z.AI. GLM-5.3-Flash model card, 2026b. URL https://huggingface.co/zai-org/GLM-5.3-Flash.

Haozhe Zhang, Kaichen Liu, Miaomiao Chen, Lei Li, Shaojie Yang, Cheng Peng, and Hanjie Chen. BenchCAD: A comprehensive, industry-standard benchmark for programmatic CAD, 2026a. URL https://arxiv.org/abs/2605.10865.

Pingyue Zhang, Zihan Huang, Yue Wang, Jieyu Zhang, Letian Xue, Zihan Wang, Qineng Wang, Keshigeyan Chandrasegaran, Ruohan Zhang, Yejin Choi, Ranjay Krishna, Jiajun Wu, Fei-Fei Li, and Manling Li. Theory of Space: Can foundation models construct spatial beliefs through active exploration? In *The Fourteenth International Conference on Learning Representations*, 2026b.

Yuhan Zhang, Mengchen Zhang, Tong Wu, Tengfei Wang, Gordon Wetzstein, Dahua Lin, and Ziwei Liu. 3DGen-Bench: Comprehensive benchmark suite for 3D generative models, 2025. URL https://arxiv.org/abs/2503.21745.

Yan Zheng and Florian Bordes. VoxelCodeBench: Benchmarking 3D world modeling through code generation, 2026. URL https://arxiv.org/abs/2604.02580.

Matt Zhou, Ruining Li, Xiaoyang Lyu, Zhaomou Song, Zhening Huang, Chuanxia Zheng, Christian Rupprecht, Andrea Vedaldi, and Shangzhe Wu. ArtiCraft: An agentic system for scalable articulated 3D asset generation, 2026. URL https://arxiv.org/abs/2605.15187.

Overview

This appendix collects implementation details and supplementary analyses that support the main benchmark description. *Dataset processing details* documents the preprocessing, filtering and annotation steps and the statistics of the resulting sets; *Experimental setup* lists the evaluated models and their inference setup; *Evaluation details* covers mesh preprocessing, score aggregation, QA bank construction and Assembly-3D *Part* metrics. *Additional analyses and experiments* reports human validation of the MLLM Judge, qualitative case studies of the Judge and Part submetrics, three diagnostic analyses, and a multi-turn agentic workflow. *Full submetric results* reports every submetric for the three tasks under each output format, and *Qualitative visualizations* presents qualitative outputs from the evaluated models.

A Dataset processing details

This section details dataset preprocessing, filtering and annotation, followed by statistics and annotation examples.

A.1 Pipeline implementation

Text2CAD preprocessing. Preprocessing reduces the 176,017 Text2CAD records (Khan et al., 2024) to the candidate pool for filtering. We discard records that cannot be evaluated at all — missing programs, zero-depth extrusions and empty shapes — and rank the survivors by a geometric complexity score over the modeling-operation sequence, namely the number of sketch and extrude operations and the face count, keeping the top-ranked cases as candidates. The Fusion 360 Gallery assemblies (Willis et al., 2022) already carry multi-part structure and skip this preselection.

Near-duplicate removal. The candidates retained after the review MLLM still contain many visually similar parts and assemblies, which we remove by DINOv2 (Oquab et al., 2024) feature matching over render embeddings, applied identically to both sources. Each candidate is embedded by the CLS token of DINOv2's last hidden state on its render, L2-normalized so that cosine similarity is a dot product. We then walk the pairs with similarity $\geq \tau$ in descending order and, for each pair still alive, drop the endpoint with the higher average similarity to the remaining alive set, since the case that looks more like everyone else is the more redundant one.

Text annotation. As shown in the annotation pipeline of Figure A1, the two Text-to-3D specifications are not written from the raw source program. We first parse the source minimal JSON into a structured geometric record that makes the part's sketches, extrusions and bounding boxes explicit, and pass it with the render to the annotation MLLM (GPT-5.5), so that both specifications are grounded in named geometric quantities rather than inferred from the render. The record also backs the static validator that closes the review-and-repair loop — a case is admitted only after every number and feature in its text checks out against it — and keeps the two specifications distinct, allowing no numerical parameters in the descriptive one and only recorded features in the parametric one. Both specifications are written as natural-language design descriptions.

A.2 Dataset statistics and annotation examples

Figure A2 summarizes the two 400-case filtered sets. The semantic category distribution (Figure A2b) is long-tailed for both: six part-level categories for Text-to-3D, dominated by support mounting, and thirteen assembly-level categories for Image-to-3D, led by mechanical systems and vehicles. Figure A2a shows example cases at the easy, medium and hard complexity levels. Complexity-balanced sampling keeps all three tiers populated: the Text-to-3D set splits $50/200/150$ over easy, medium and hard, and the Image-to-3D set $80/160/160$. The Assembly-3D annotation pipeline then drops Image-to-3D assemblies that exceed the deduplicated-part cap or fail MLLM verification, leaving 203. These are the complete dataset sizes; the main experiments use a fixed 100-case subset of each task. Figure A3 shows the resulting annotations for both tracks.

B Experimental setup

Table A1 lists the evaluated models by type, with parameter counts and each domain-specific baseline's input–output setting used here. All general-purpose LLMs and MLLMs use their highest supported reasoning level. Domain-specific models run from their released checkpoints under those settings, and generated programs are executed in their native toolchains—OpenSCAD (OpenSCAD Developers, 2021), CadQuery (CadQuery contributors, 2026) and Three.js (three.js authors, 2026). We use Cadrille's image-conditioned mode. Local evaluation used Ubuntu 22.04 with dual Intel Xeon Silver 4314 CPUs, 512 GB RAM and eight NVIDIA RTX A6000 GPUs, under Python 3.12.2, OpenSCAD 2021.01 and CadQuery 2.7.0.

C Evaluation details

Bucket score aggregation. Invalid predictions remain in the evaluation denominators, with zero normalized sub-metrics and bucket scores and worst-case raw errors (0.01 for CD). For otherwise valid predictions, unavailable IoU is omitted from per-case Geo and the standalone IoU mean; measured zeros and invalid-output penalties are retained. Sub-metrics are mapped to $[0, 1]$, with 1 best: visual-Judge ratings use $(s-1)/9$ for $s \in [1, 10]$, and CD uses $\max(0,\ 1-\mathrm{CD}/0.01)$; reported

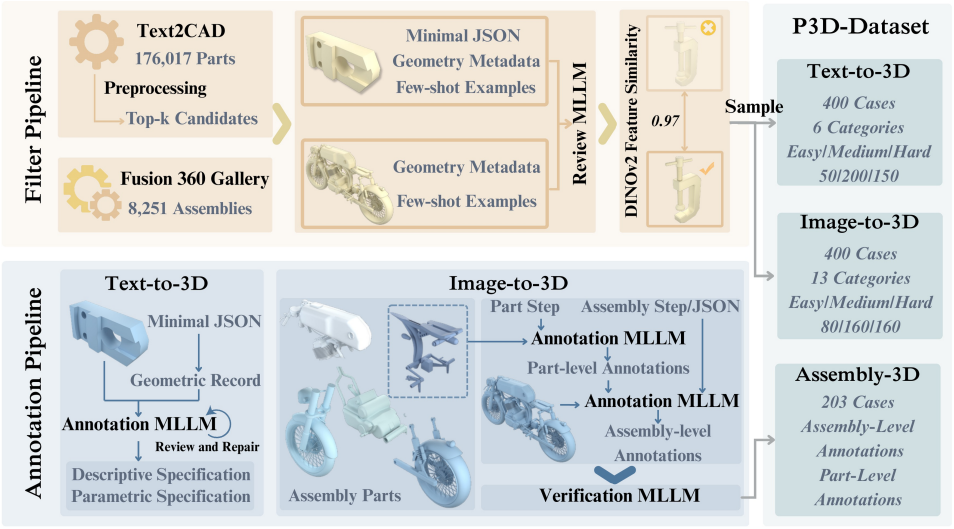
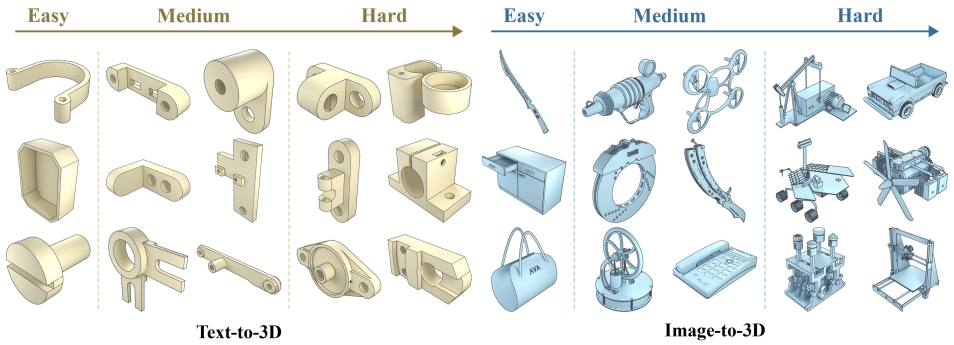
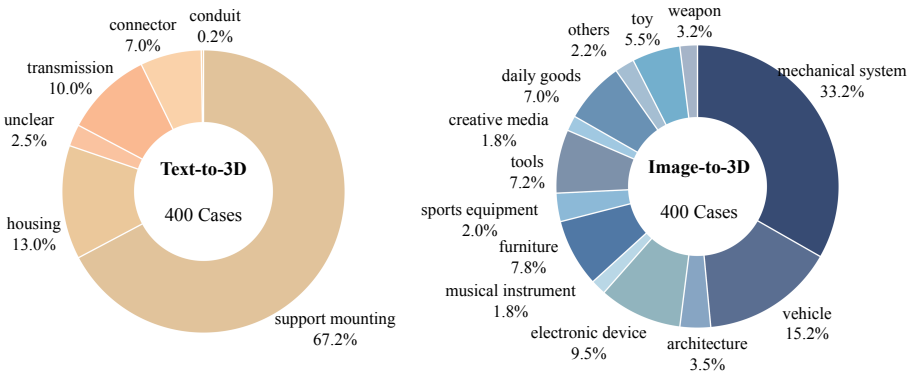


Figure A1: **Overview of our P3D-Dataset.** The *filter pipeline* draws from the Text2CAD and Fusion 360 Gallery data sources, filters candidates with a review MLLM, removes near-duplicates, and samples 400 Text-to-3D and 400 Image-to-3D cases with category and complexity distributions. The *annotation pipeline* then annotates the filtered Text-to-3D data with an annotation MLLM into descriptive (desc) and parametric (param) text specifications; for the Image-to-3D data, both parts and assemblies are labeled by an annotation MLLM and checked by a verification MLLM, yielding the 203-case Assembly-3D set.



(a) Example cases at the easy, medium and hard complexity levels.



(b) Semantic category distribution of the Text-to-3D and Image-to-3D sets.

Figure A2: **The Text-to-3D and Image-to-3D 400-case filtered sets.**

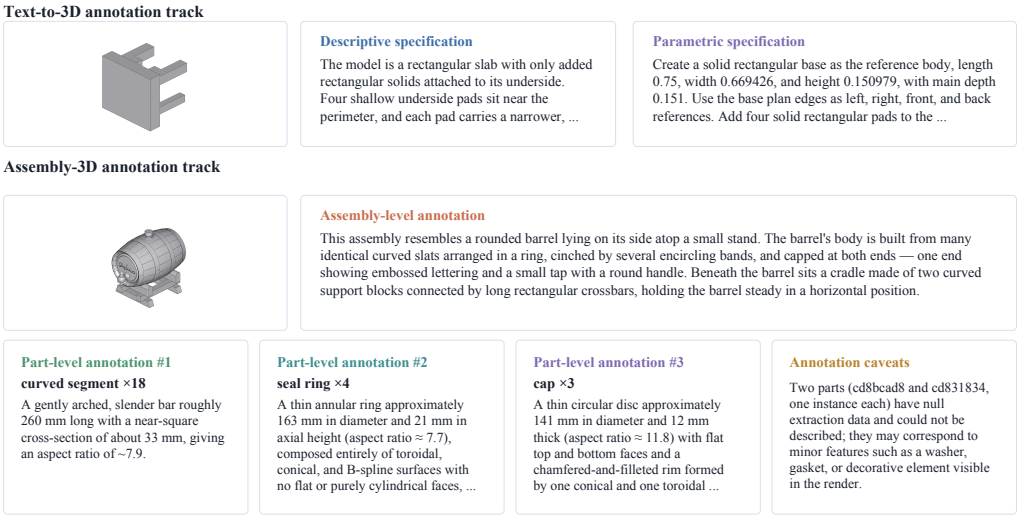


Figure A3: **Example annotations from the two data sources.** Top: a retained Text2CAD program with its source render, converted into the descriptive and parametric specifications used by Text-to-3D. Bottom: a Fusion 360 assembly render with the verified assembly-level caption and per-part descriptions used by Assembly-3D. Text boxes show verbatim excerpts of the released input fields; the benchmark prompts use the full fields.

Table A1: **Models evaluated in P3D-BENCH, grouped by model type.** Size is the developer-reported parameter count, with active parameters per token for MoE; "–" marks closed-weight models with undisclosed counts, and domain-specific sizes refer to the backbone. DeepSeek's count includes its backbone and conditional memory. Markers give each domain-specific baseline's input → output setting used here: * image → CadQuery, § text → minimal JSON. GLM-5.3-Flash is the multimodal variant evaluated on all three tasks; GLM-5.3 is the text-only variant evaluated only on Text-to-3D.

Model type	Model	Size (total / active)
Multimodal LLM	Claude Opus 5 (Anthropic, 2026)	–
	Gemini 3.8 Flash (Google DeepMind, 2026)	–
	GPT-6 Astra (OpenAI, 2026)	–
	Qwen3.8-Max (Alibaba Cloud, 2026)	2.4 T / –
	GLM-5.3-Flash (Z.AI, 2026b)	320 B / 18 B
	Doubao Seed 2.1 Pro (ByteDance Seed, 2026)	–
	Kimi K3 (Moonshot AI, 2026)	2.8 T / 104 B
	Grok 4.6 (SpaceXAI, 2026)	–
	DeepSeek V4.1 Flash (DeepSeek-AI, 2026)	748 B / 8–16 B
Text-only LLM	GLM-5.3 (Z.AI, 2026a)	753 B / –
Domain-specific model	CADRILLE* (Kolodiazhnyi et al., 2026)	2 B (Qwen2-VL-2B)
	CAD-CODER* (Doris et al., 2025)	13 B (Vicuna-13B)
	Text2CAD§ (Khan et al., 2024)	363 M (BERT-Large + decoder)

CD is capped at 0.01 before averaging. Each bucket averages its applicable normalized sub-metrics equally, except parametric Text-to-3D Judge, which uses $(\text{QA-S} + 2\,\text{QA-P})/3$ to reflect its four QA-S and eight QA-P questions. Geo is computed per prediction before averaging over cases within each output format; reported bucket scores weight the task-supported formats equally. Part measurements rejected by the fidelity gate are excluded from Part means.

The headline Score averages descriptive Judge, parametric Geo and parametric Judge on Text-to-3D; Geo and Judge on Image-to-3D; and Geo, Judge and Part on Assembly-3D, then scales each mean to 0–100.

Mesh normalization, alignment and IoU. We center both meshes and, when enabled, normalize their scale to the unit cube; parametric Text-to-3D retains absolute scale because dimensions are specified in the input. Alignment minimizes bidirectional CD over axis permutations and sign flips. Candidates within 5% of the minimum are treated as tied, and the rotation closest to identity is retained to preserve the generated orientation when possible. A search in the prediction's PCA frame is accepted only if it improves CD by at least 10%. We then refine translation and, when permitted, uniform scale with a locally bounded search. IoU uses CSG solids ($\mathrm{IoU_C}\uparrow$) for single-part parametric specification outputs and voxel grids ($\mathrm{IoU_V}\uparrow$) for Image-to-3D and Assembly-3D assemblies. Voxel IoU is more robust to non-manifold assembly geometry because it requires a closed surface rather than a manifold solid. Both variants require the predicted and ground-truth meshes to be free of open edges. For an otherwise valid prediction with an open surface, IoU is undefined rather than a measured zero, so it is omitted from that case's Geo mean and the standalone IoU mean; NoOE reports the open-edge condition separately. Invalid-output penalties and measured zeros are retained as described above.

QA bank construction. Both banks are synthesized offline by Gemini 3.1 Pro, once per case and then frozen; questions are limited to the given text specification, with the ground-truth program and render used only to verify answers. Each question has five options, including "none of the above", and some questions deliberately use that answer so that the grader cannot succeed by always choosing the nearest concrete value. QA-S excludes exact numeric values, whereas QA-P mainly asks for derived quantities that require combining measurements or applying a geometric formula rather than copying a stated parameter; its distractors reflect plausible incorrect derivations. Finally, the same model re-derives every answer from the ground truth and rewrites any question it finds incorrect. At evaluation time, Gemini 3.8 Flash answers the frozen questions to score each prediction.

Assembly-3D *Part* metrics. Parts share one alignment transform per side and are never rescaled, preserving differences in relative part size. We sample 2048 surface points per part with a fixed seed. Each candidate pair is centroid-aligned and takes the best of the 24 proper cube rotations under bidirectional CD. Hungarian matching uses a coverage F-score at $\tau_g = 0.03\,\mathrm{diag}_g$, and a pair is successful at $F_{\min} = 0.7$. A rotation- and translation-invariant fingerprint first deduplicates repeated instances of the same body.

D Additional analyses and experiments

We present human validation of the Judge, qualitative Judge and Part examples, and analyses of cost, decomposition fidelity, output validity and multi-turn feedback.

D.1 Human validation of the MLLM Judge

We validate the *Judge* components that humans can reproduce: QA-S for Text-to-3D, and *J-Geo*, *J-Aes* and *J-Sem* for Image-to-3D and Assembly-3D; QA-P is omitted because it requires deriving quantities from predicted source code. For each task, human raters assess the same 10 cases. Visual-Judge cases are score-stratified; QA-S items span correct and incorrect Judge answers.

Text-to-3D: QA-S. Given the input specification and the ground-truth or predicted render, annotators answer the same QA-S items as the Judge. Their responses agree with the reference answer key in 80.0% of trials and with the Judge's answer on the prediction in 68.3% (Table A2a).

Image-to-3D and Assembly-3D: visual judge. Annotators receive the same evidence and rubric as the MLLM Judge and rate *J-Geo*, *J-Aes* and *J-Sem* on the same 1–10 scale. We average the three axes per case and compare the Judge score with the mean human score using Pearson r and Spearman ρ. The composite agreement is strong on the sampled cases (Table A2b): $r = 0.961, \rho = 0.924$ on Image-to-3D and $r = 0.951, \rho = 0.921$ on Assembly-3D, indicating that the MLLM Judge closely tracks human score variation and rankings on these cases.

Table A2: **Human validation of the MLLM Judge.** **(a)** QA-S agreement with the reference answer key and with the Judge's answer on the prediction. **(b)** Pearson r and Spearman ρ between the visual Judge and mean human ratings over 10 cases per task; *Composite* is the mean of the three axes and *Pooled* uses all 20 cases.

(a) QA metric, on Text-to-3D QA-S

Metric	Reference	Prediction
QA-S agreement	80%	68.3%

(b) Visual judge, on Image-to-3D and Assembly-3D

	Image-to-3D		Assembly-3D		Pooled	
Axis	r	ρ	r	ρ	r	ρ
J-Geo	0.824	0.870	0.850	0.812	0.838	0.860
J-Aes	0.930	0.858	0.870	0.821	0.888	0.853
J-Sem	0.928	0.815	0.934	0.911	0.930	0.894
Composite	0.961	0.924	0.951	0.921	0.952	0.931

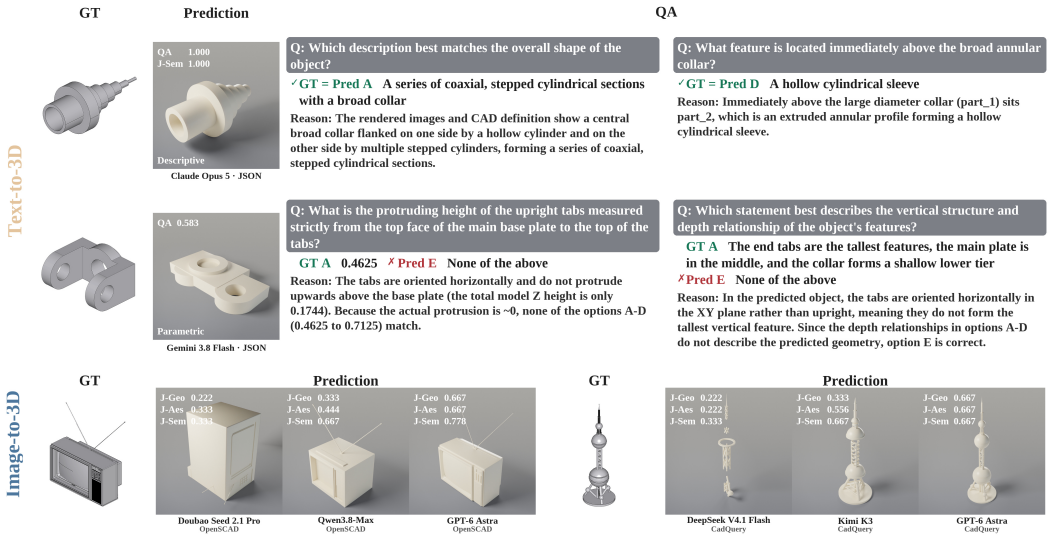


Figure A4: **Qualitative cases showing Judge submetrics.** *(top)* One descriptive and one parametric specification example, each with its QA results and reasons. *(bottom)* Two Image-to-3D cases with their visual Judge submetrics *J-Geo*/*J-Aes*/*J-Sem*.

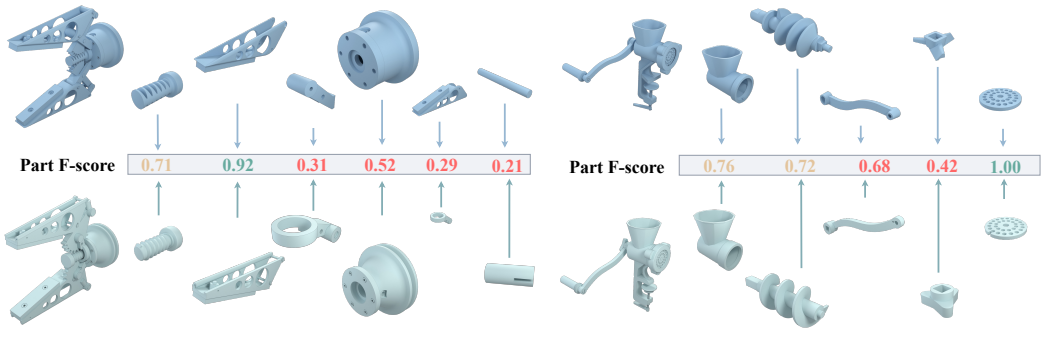


Figure A5: **Qualitative cases showing part-match results.** Selected ground-truth parts (top) from two complex assemblies are paired with predicted parts (bottom) by Hungarian assignment, with each pair annotated by its per-part F-score F at the 3% tolerance: green for $F \geq 0.9$, yellow for $0.7 \leq F < 0.9$, and red for rejected pairs with $F < 0.7$.

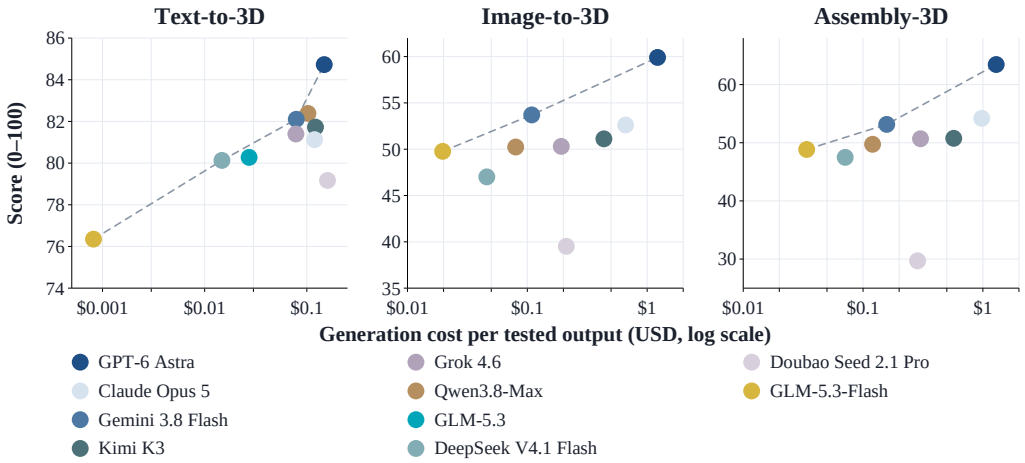


Figure A6: **Per-task P3D-BENCH Score versus mean API-equivalent generation cost per tested output (USD, log scale).** Each point represents a model; marker colour identifies the model, and the dashed line connects selected models to show the cost–quality trend.

D.2 QUALITATIVE JUDGE AND PART RESULTS

Figures A4 and A5 provide qualitative examples of the Judge and Part metrics, respectively. The former compares Judge submetrics across Text-to-3D and Image-to-3D cases, while the latter visualizes part correspondences and their per-part F-scores in Assembly-3D.

Across these examples, the metric values agree with the qualitative evidence: outputs that better match the reference receive higher scores, whereas visible structural, geometric, or part-matching errors receive lower scores. This consistency supports the validity of both metrics.

D.3 COST-QUALITY ANALYSIS

Figure A6 plots task-specific P3D-BENCH scores against mean API-equivalent generation cost per tested output. For each model and task, we divide total generation cost by the number of tested outputs, including invalid ones. For Text-to-3D, the cost includes only the selected generation for each output. For Image-to-3D and Assembly-3D, it also includes recorded execution-feedback corrections; the correction protocol is detailed in Appendix D.5.

GPT-6 Astra occupies the high-quality, high-cost end across all three tasks. Gemini 3.8 Flash offers a cost-effective option at intermediate budgets across all three tasks. GLM-5.3-Flash offers a cost-effective alternative across all three tasks. DeepSeek V4.1 Flash is also cost-effective on Text-to-3D.

D.4 ASSEMBLY-3D DECOMPOSITION FIDELITY

Table A3 reports per-model decomposition fidelity: how closely the union of the re-assembled decomposition matches the prediction it came from, in Chamfer Distance ($\mathrm{D}_{\mathrm{dec}}$, lower is better) and volumetric IoU ($\mathrm{IoUV}_{\mathrm{dec}}$, higher is better), together with the resulting gate exclusions.

Across models and output formats, valid predictions that are successfully decomposed retain high geometric fidelity after reassembly: $\mathrm{D}_{\mathrm{dec}}$ stays in the 10^{-5}–10^{-4} range and $\mathrm{IoUV}_{\mathrm{dec}}$ exceeds 0.92. At most 5 cases per model and format are excluded by the fidelity gate. These results indicate that the decomposition pipeline preserves the original predicted geometry well on the evaluated cases.

D.5 FIRST AND FINAL OUTPUT VALIDITY

For Image-to-3D and Assembly-3D, the generation protocol allows up to three generate–execute attempts: an initial output followed by at most two corrections using execution feedback. We call the

Table A3: **Decomposition fidelity diagnostic per model and output format.** *Valid* is the executable-prediction rate, reported as a percentage; $\mathrm{D}_{\mathrm{dec}}$ and $\mathrm{IoUV}_{\mathrm{dec}}$ are the predicted vs. re-assembled union Chamfer Distance and volumetric IoU, averaged over the same valid, successfully decomposed cases so that neither is confounded by *Valid*; *Excl.* counts the cases dropped by the fidelity gate.

	CadQuery				OpenSCAD			
Model	Valid↑	$\mathrm{D}_{\mathrm{dec}}$↓	$\mathrm{IoUV}_{\mathrm{dec}}$↑	Excl.↓	Valid↑	$\mathrm{D}_{\mathrm{dec}}$↓	$\mathrm{IoUV}_{\mathrm{dec}}$↑	Excl.↓
GPT-6 Astra	99%	0.000075	0.987	0	100%	0.000076	0.986	0
Claude Opus 5	100%	0.000210	0.973	1	100%	0.000071	0.983	0
Gemini 3.8 Flash	98%	0.000073	0.982	0	100%	0.000070	0.985	0
Kimi K3	98%	0.000072	0.984	0	100%	0.000274	0.964	3
Grok 4.6	100%	0.000581	0.964	1	100%	0.000068	0.985	0
Qwen3.8-Max	98%	0.000064	0.987	0	100%	0.000069	0.984	0
GLM-5.3-Flash	98%	0.000076	0.983	0	100%	0.000069	0.987	0
DeepSeek V4.1 Flash	97%	0.000260	0.978	1	99%	0.000071	0.989	0
Doubao Seed 2.1 Pro	41%	0.000524	0.935	2	98%	0.000480	0.927	5

Table A4: **First and final invalid rates on the grounded tasks.** Rates are reported as percentages and averaged over the output formats evaluated for each task in the main experiments. Final denotes the released outcome after the recorded corrections.

		Invalid rate								
Task	Stage	GPT-6 Astra	Claude Opus 5	Gemini 3.8 Flash	Kimi K3	Grok 4.6	Qwen 3.8-Max	GLM 5.3-Flash	DeepSeek V4.1 Flash	Doubao Seed 2.1 Pro
Image-to-3D	First	11.4%	10.7%	17.7%	13.7%	15%	19.7%	10.7%	19%	31.4%
	Final	0.4%	0%	1%	0%	1%	0.3%	0.7%	3%	15.3%
Assembly-3D	First	2.7%	5.3%	17%	17.7%	13.9%	21.3%	6.9%	20.7%	50%
	Final	0.6%	0%	1%	1%	0%	1.1%	1.1%	2.1%	30.5%

result of the first attempt *First* and the released result after this process *Final*. An output is invalid if it cannot be parsed, executed, structurally validated or exported. Table A4 compares the two stages.

Text-to-3D uses simpler single-part targets and does not adopt the three-attempt execution-feedback correction protocol used for Image-to-3D and Assembly-3D. Its mean final invalid rate is only 0.425% across all ten general-purpose models, both specification types and JSON/OpenSCAD.

The model columns follow descending final Assembly-3D Score in Table 2. The ranking broadly agrees with final validity: all models except Doubao Seed 2.1 Pro reach at most 3.0% final invalid on Image-to-3D and 2.1% on Assembly-3D. Doubao remains the clear outlier after correction, at 15.3% and 30.5%, respectively, consistent with its substantially lower final scores.

D.6 Multi-turn agentic workflow

Beyond execution-error correction, we test whether iterative visual feedback improves reconstruction quality. On the fixed 100-case hard subset of the Image-to-3D/OpenSCAD task, the single-shot row in Table A5 reuses Gemini 3.8 Flash's primary benchmark results, including execution-error correction but no visual review. For the multi-turn setting, the model first drafts a program from the same reference image and then uses the reference image, current code, and multi-view renders to revise its output for up to ten review turns.

Multi-turn feedback raises Geo from 0.526 to 0.542, Judge from 0.552 to 0.586, and Score from 53.88 to 56.38; Topo remains 1.000 at the reported precision. These gains show that iterative visual feedback improves reconstruction quality on this subset. The model averages 8.82 review turns, with 66% of runs reaching the ten-turn limit, indicating frequent use of the full review budget.

Table A5: **Multi-turn agentic workflow vs. single-shot generation on the 100-case hard subset of Image-to-3D/OpenSCAD.** *Score* aggregates Geo and Judge; Topo is reported as an independent diagnostic. *Turns* is the mean number of review turns after the initial draft (10 is the maximum); "–" marks the single-shot setting, for which the turn count does not apply. Geo, Judge and Score use bold for the better setting.

Setting	Model	Geo	Topo	Judge	Score	Turns
Single-shot	Gemini 3.8 Flash	0.526	1.000	0.552	53.876	–
Multi-turn	Gemini 3.8 Flash	**0.542**	1.000	**0.586**	**56.380**	8.82

E FULL SUBMETRIC RESULTS

Tables A6, A7 and A8 report per-format metrics for Assembly-3D, Text-to-3D (both specification styles) and Image-to-3D, respectively, following Appendix C. *Valid* and NoOE are percentages; other submetrics retain their native or normalized scales. In particular, the visual-Judge submetrics report 1–10 ratings s normalized to $[0, 1]$ by $(s - 1)/9$. The cross-format mean of *Valid* appears in Table 2.

Table A6: **Assembly-3D full submetric results on the 100-case evaluation subset.** Panels (a) and (b) report CadQuery and OpenSCAD outputs, respectively. All panels include executable validity *Valid*. F_1, P and R denote $\mathrm{PartMatchF1}$, $\mathrm{PartMatchP}$ and $\mathrm{PartMatchR}$. The Part bucket averages only F_1 and PartFS; P and R are diagnostics. Normalization and aggregation follow Appendix C; emphasis on *Geo*, *Judge* and *Part*-related metrics follows Table 2.

(a) CadQuery output.

	Geo				*Topo*			*Judge*			*PartMatch*		*Part*		
Model	CD↓	$\mathrm{IoU_V}$↑	F@.01↑	NC↑	NoOE↑	InvN↓	NM↓	J-Geo↑	J-Aes↑	J-Sem↑	P ↑	R ↑	F_1 ↑	PartFS↑	*Valid*↑
GPT-6 Astra	**0.0009**	**0.632**	**0.478**	**0.727**	77%	0.011	0.012	**0.536**	**0.598**	**0.640**	**0.599**	**0.539**	**0.541**	**0.738**	98.9%
Claude Opus 5	0.0017	0.427	0.325	0.657	91%	0.000	0.001	0.416	0.526	0.636	0.502	0.426	0.444	0.681	100%
Gemini 3.8 Flash	0.0020	0.429	0.310	0.637	89%	0.021	0.022	0.431	0.516	0.628	0.460	0.366	0.391	0.616	97.9%
Kimi K3	0.0025	0.359	0.271	0.618	74%	0.021	0.022	0.366	0.454	0.608	0.415	0.423	0.409	0.619	97.9%
Grok 4.6	0.0023	0.381	0.290	0.635	86%	0.000	0.002	0.367	0.474	0.584	0.407	0.322	0.338	0.609	100%
Qwen3.8-Max	0.0029	0.337	0.273	0.612	80%	0.022	0.022	0.349	0.437	0.576	0.392	0.364	0.368	0.576	97.8%
GLM-5.3-Flash	0.0023	0.374	0.276	0.610	82%	0.023	0.024	0.347	0.462	0.587	0.362	0.335	0.342	0.566	97.7%
DeepSeek V4.1 Flash	0.0029	0.383	0.269	0.594	81%	0.032	0.037	0.344	0.449	0.569	0.381	0.330	0.340	0.591	96.8%
Doubao Seed 2.1 Pro	0.0072	0.119	0.102	0.240	37%	0.596	0.598	0.128	0.156	0.208	0.159	0.118	0.130	0.217	41%

(b) OpenSCAD output.

	Geo				*Topo*			*Judge*			*PartMatch*		*Part*		
Model	CD↓	$\mathrm{IoU_V}$↑	F@.01↑	NC↑	NoOE↑	InvN↓	NM↓	J-Geo↑	J-Aes↑	J-Sem↑	P ↑	R ↑	F_1 ↑	PartFS↑	*Valid*↑
GPT-6 Astra	**0.0007**	**0.583**	**0.448**	**0.723**	100%	0.000	0.000	**0.539**	**0.614**	**0.666**	**0.573**	**0.489**	**0.501**	**0.710**	100%
Claude Opus 5	0.0019	0.404	0.317	0.647	100%	0.000	0.000	0.392	0.517	0.637	0.461	0.392	0.414	0.631	100%
Gemini 3.8 Flash	0.0018	0.437	0.329	0.659	100%	0.000	0.001	0.424	0.532	0.646	0.455	0.357	0.380	0.632	100%
Kimi K3	0.0021	0.379	0.287	0.642	100%	0.000	0.001	0.369	0.471	0.628	0.436	0.403	0.406	0.615	100%
Grok 4.6	0.0017	0.408	0.322	0.656	100%	0.000	0.001	0.398	0.513	0.638	0.397	0.325	0.338	0.616	100%
Qwen3.8-Max	0.0019	0.401	0.297	0.636	100%	0.000	0.000	0.371	0.481	0.623	0.447	0.409	0.416	0.625	100%
GLM-5.3-Flash	0.0022	0.379	0.287	0.635	100%	0.000	0.000	0.370	0.498	0.621	0.373	0.330	0.341	0.587	100%
DeepSeek V4.1 Flash	0.0023	0.355	0.266	0.617	99%	0.010	0.011	0.350	0.472	0.596	0.356	0.301	0.315	0.577	99%
Doubao Seed 2.1 Pro	0.0033	0.308	0.236	0.595	98%	0.020	0.021	0.287	0.357	0.494	0.337	0.262	0.279	0.541	98%

Table A7: **Text-to-3D full submetric results on the 100-case evaluation subset.** Panel (a) reports descriptive specifications for JSON and OpenSCAD; panels (b) and (c) report parametric specifications for JSON and OpenSCAD, respectively. All panels include executable validity *Valid*. The domain-specific baseline Text2CAD uses native JSON and is listed separately below the general-purpose models. "–" denotes an unevaluated model–format pair. Normalization and aggregation follow Appendix C; emphasis on *Geo* and *Judge* metrics follows Table 2.

(a) descriptive specification.

	JSON			*OpenSCAD*		
Model	QA-S↑	J-Sem↑	*Valid*↑	QA-S↑	J-Sem↑	*Valid*↑
GPT-6 Astra	**0.978**	**0.877**	100%	0.993	**0.859**	100%
Claude Opus 5	0.932	0.726	100%	0.975	0.751	100%
Gemini 3.8 Flash	0.960	0.751	100%	**0.995**	0.758	100%
Kimi K3	0.897	0.764	97%	0.990	0.826	100%
Grok 4.6	0.960	0.726	100%	**0.995**	0.756	100%
Qwen3.8-Max	0.963	0.772	100%	0.988	0.822	100%
GLM-5.3	0.900	0.791	100%	0.935	0.783	99%
GLM-5.3-Flash	0.905	0.757	98%	0.870	0.712	97%
DeepSeek V4.1 Flash	0.910	0.771	100%	0.975	0.793	100%
Doubao Seed 2.1 Pro	0.880	0.771	99%	0.938	0.780	100%
Text2CAD	0.058	0.200	91%	–	–	–

(b) parametric specification, JSON.

	Geo				*Topo*			*Judge*		
Model	CD↓	IoU_C ↑	F@.01↑	NC↑	NoOE↑	InvN↓	NM↓	QA-S↑	QA-P↑	*Valid*↑
GPT-6 Astra	0.0013	0.618	0.412	0.756	96%	0.0000	0.0004	**0.917**	**0.907**	100%
Claude Opus 5	0.0012	0.603	0.402	0.746	97%	0.0000	0.0005	0.868	0.895	100%
Gemini 3.8 Flash	**0.0011**	0.620	0.428	0.767	96%	0.0000	0.0008	0.858	0.892	100%
Kimi K3	0.0013	0.602	0.399	0.742	95%	0.0100	0.0117	0.868	0.894	99%
Grok 4.6	**0.0011**	**0.623**	**0.436**	**0.768**	97%	0.0000	0.0009	0.853	0.890	100%
Qwen3.8-Max	**0.0011**	0.610	0.409	0.762	94%	0.0000	0.0016	0.877	0.905	100%
GLM-5.3	0.0013	0.574	0.382	0.733	97%	0.0000	0.0012	0.860	0.875	100%
GLM-5.3-Flash	0.0016	0.518	0.354	0.711	93%	0.0200	0.0211	0.843	0.865	98%
DeepSeek V4.1 Flash	0.0014	0.588	0.398	0.742	95%	0.0100	0.0118	0.820	0.886	99%
Doubao Seed 2.1 Pro	0.0015	0.572	0.389	0.738	95%	0.0100	0.0111	0.848	0.871	99%
Text2CAD	0.0056	0.173	0.081	0.473	98%	0.0200	0.0278	0.077	0.163	98%

(c) parametric specification, OpenSCAD.

	Geo				*Topo*			*Judge*		
Model	CD↓	IoU_C ↑	F@.01↑	NC↑	NoOE↑	InvN↓	NM↓	QA-S↑	QA-P↑	*Valid*↑
GPT-6 Astra	**0.0007**	**0.676**	**0.445**	**0.790**	100%	0.0000	0.0000	**0.945**	0.944	100%
Claude Opus 5	0.0009	0.653	0.432	0.775	100%	0.0000	0.0008	**0.945**	0.934	100%
Gemini 3.8 Flash	**0.0007**	0.671	0.437	0.786	100%	0.0000	0.0001	0.917	0.936	100%
Kimi K3	0.0009	0.647	0.418	0.764	100%	0.0000	0.0005	0.930	**0.945**	100%
Grok 4.6	0.0009	0.652	0.430	0.780	100%	0.0000	0.0002	0.885	0.929	100%
Qwen3.8-Max	0.0009	0.650	0.416	0.773	100%	0.0000	0.0000	0.910	0.921	100%
GLM-5.3	0.0010	0.622	0.398	0.756	100%	0.0000	0.0007	0.932	0.929	100%
GLM-5.3-Flash	0.0017	0.543	0.336	0.707	99%	0.0100	0.0105	0.848	0.906	99%
DeepSeek V4.1 Flash	0.0012	0.610	0.401	0.748	99%	0.0100	0.0114	0.887	0.919	99%
Doubao Seed 2.1 Pro	0.0012	0.593	0.388	0.747	100%	0.0000	0.0003	0.885	0.922	100%

Table A8: **Image-to-3D full submetric results on the 100-case evaluation subset.** Panels (a), (b) and (c) report CadQuery, OpenSCAD and Three.js outputs, respectively. All panels include executable validity *Valid*. The domain-specific baselines CADRILLE and CAD-CODER use native CadQuery, are listed separately below the general-purpose models. Normalization and aggregation follow Appendix C; emphasis on *Geo* and *Judge* metrics follows Table 2.

(a) CadQuery output.

	Geo				*Topo*			*Judge*			
Model	CD↓	$\mathrm{IoU_V}$ ↑	F@.01↑	NC↑	NoOE↑	InvN↓	NM↓	J-Geo↑	J-Aes↑	J-Sem↑	*Valid*↑
GPT-6 Astra	**0.0012**	**0.414**	**0.315**	**0.599**	81%	0.012	0.015	**0.508**	**0.628**	**0.661**	98.8%
Gemini 3.8 Flash	0.0017	0.351	0.257	0.566	93%	0.021	0.022	0.436	0.531	0.646	98%
Claude Opus 5	0.0021	0.316	0.240	0.561	88%	0.000	0.002	0.394	0.518	0.628	100%
Kimi K3	0.0021	0.307	0.223	0.551	95%	0.000	0.005	0.382	0.508	0.632	100%
Grok 4.6	0.0027	0.274	0.216	0.525	90%	0.031	0.031	0.364	0.462	0.598	97%
Qwen3.8-Max	0.0021	0.312	0.221	0.541	92%	0.010	0.013	0.387	0.481	0.623	99%
GLM-5.3-Flash	0.0021	0.301	0.216	0.553	93%	0.000	0.000	0.361	0.480	0.616	100%
DeepSeek V4.1 Flash	0.0033	0.248	0.184	0.484	86%	0.092	0.092	0.310	0.403	0.514	91%
Doubao Seed 2.1 Pro	0.0061	0.116	0.101	0.287	46%	0.450	0.451	0.187	0.227	0.318	55%
CADRILLE	0.0074	0.056	0.076	0.372	71%	0.237	0.238	0.167	0.163	0.167	77%
CAD-CODER	0.0091	0.018	0.024	0.107	20%	0.793	0.794	0.044	0.041	0.043	20.7%

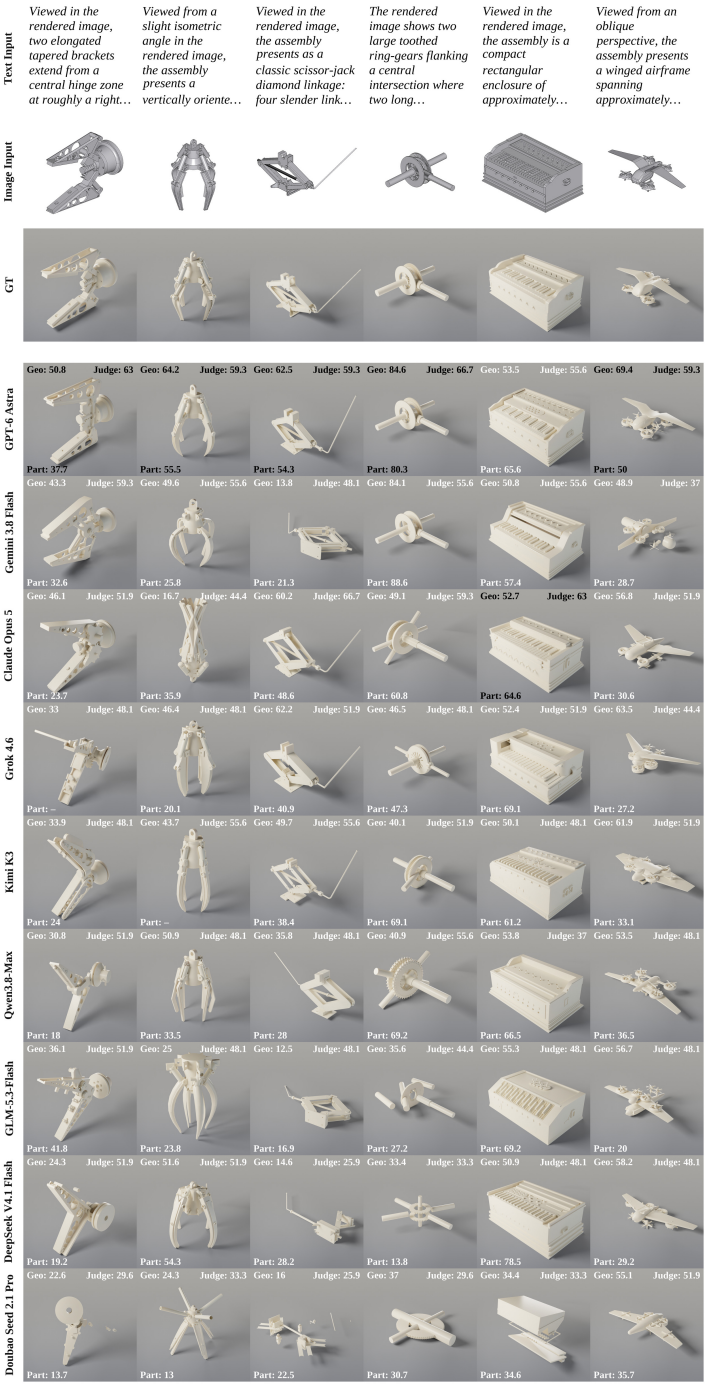
(b) OpenSCAD output.

	Geo				*Topo*			*Judge*			
Model	CD↓	$\mathrm{IoU_V}$ ↑	F@.01↑	NC↑	NoOE↑	InvN↓	NM↓	J-Geo↑	J-Aes↑	J-Sem↑	*Valid*↑
GPT-6 Astra	**0.0011**	**0.401**	**0.283**	**0.601**	100%	0.000	0.000	**0.510**	**0.633**	**0.677**	100%
Gemini 3.8 Flash	0.0018	0.358	0.258	0.566	100%	0.000	0.001	0.436	0.556	0.664	100%
Claude Opus 5	0.0018	0.327	0.234	0.557	100%	0.000	0.000	0.404	0.551	0.648	100%
Kimi K3	0.0020	0.315	0.226	0.557	100%	0.000	0.000	0.399	0.532	0.643	100%
Grok 4.6	0.0020	0.312	0.232	0.552	100%	0.000	0.000	0.400	0.523	0.641	100%
Qwen3.8-Max	0.0021	0.312	0.214	0.549	100%	0.000	0.000	0.384	0.481	0.637	100%
GLM-5.3-Flash	0.0022	0.291	0.203	0.529	98%	0.021	0.021	0.359	0.493	0.612	98%
DeepSeek V4.1 Flash	0.0023	0.300	0.214	0.548	100%	0.000	0.000	0.366	0.516	0.624	100%
Doubao Seed 2.1 Pro	0.0027	0.293	0.192	0.536	99%	0.010	0.010	0.321	0.417	0.579	99%

(c) Three.js output.

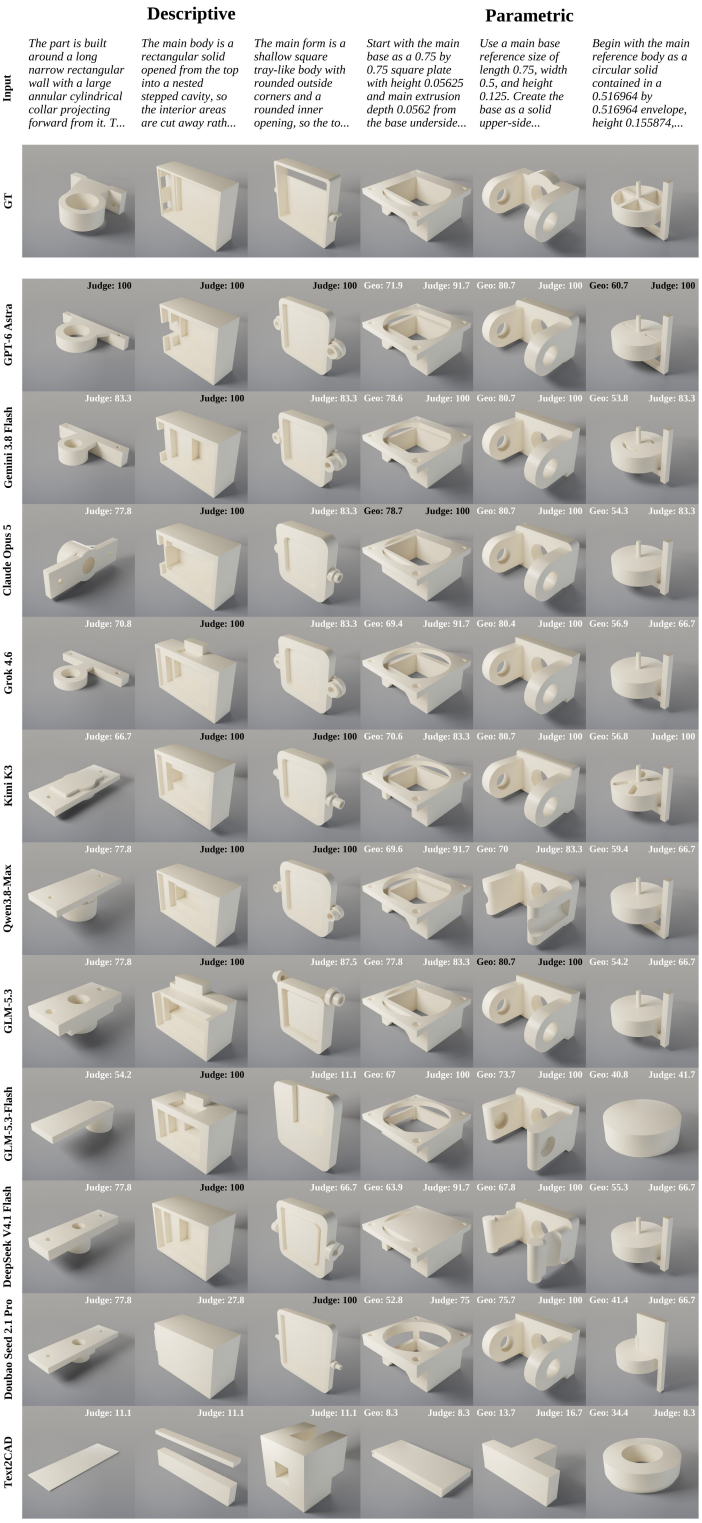
	Geo				*Topo*			*Judge*			
Model	CD↓	$\mathrm{IoU_V}$ ↑	F@.01↑	NC↑	NoOE↑	InvN↓	NM↓	J-Geo↑	J-Aes↑	J-Sem↑	*Valid*↑
GPT-6 Astra	**0.0007**	**0.465**	**0.343**	**0.619**	52%	0.002	0.006	**0.548**	**0.643**	**0.676**	100%
Gemini 3.8 Flash	0.0018	0.270	0.258	0.555	16%	0.012	0.016	0.432	0.543	0.647	99%
Claude Opus 5	0.0015	0.378	0.246	0.563	44%	0.001	0.003	0.422	0.559	0.656	100%
Kimi K3	0.0020	0.331	0.240	0.552	40%	0.000	0.012	0.380	0.523	0.636	100%
Grok 4.6	0.0019	0.346	0.248	0.555	31%	0.001	0.002	0.396	0.528	0.640	100%
Qwen3.8-Max	0.0018	0.323	0.236	0.554	61%	0.001	0.008	0.390	0.491	0.644	100%
GLM-5.3-Flash	0.0020	0.314	0.229	0.553	43%	0.001	0.007	0.396	0.520	0.651	100%
DeepSeek V4.1 Flash	0.0021	0.332	0.231	0.545	54%	0.002	0.006	0.362	0.487	0.612	100%
Doubao Seed 2.1 Pro	0.0021	0.324	0.223	0.548	36%	0.002	0.007	0.357	0.439	0.614	100%

F QUALITATIVE VISUALIZATIONS



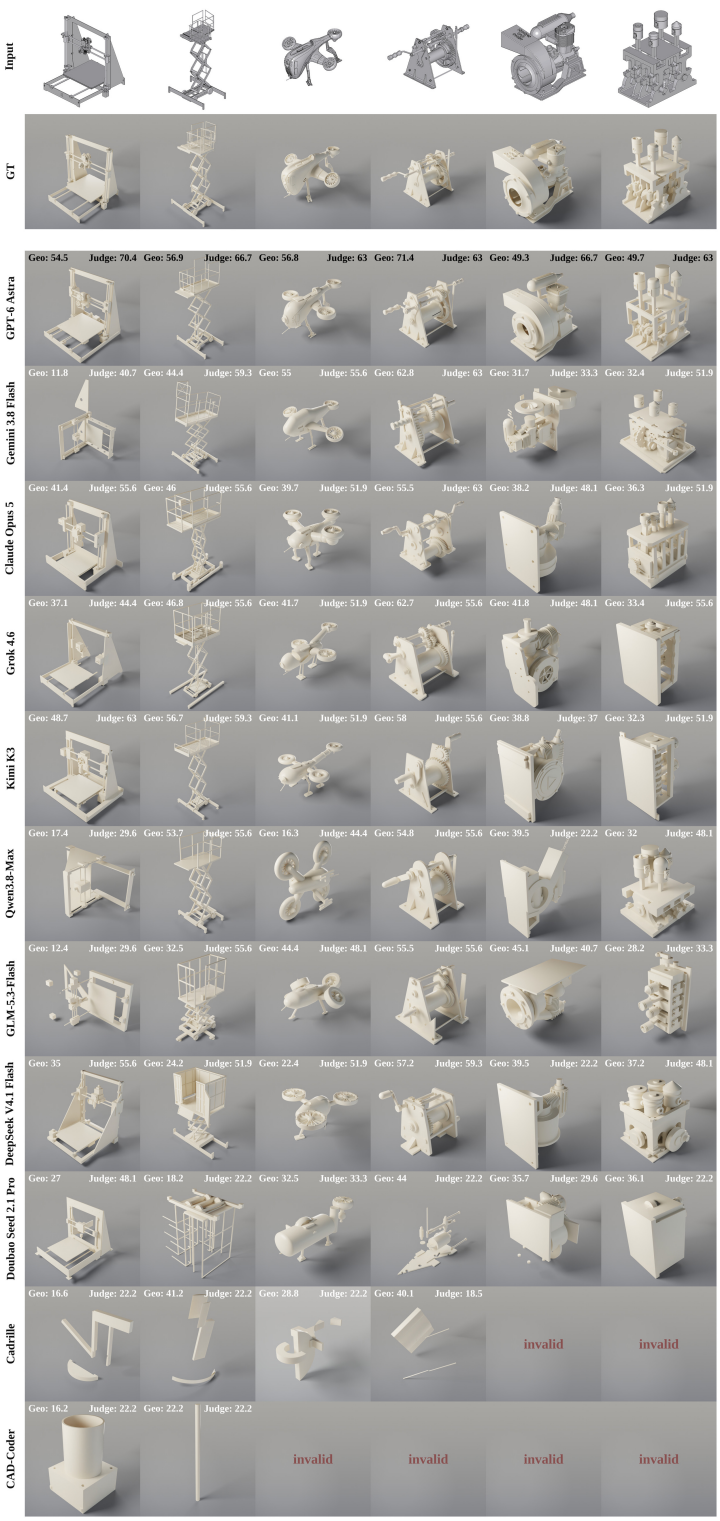
(a) Assembly-3D: each column pairs the assembly input, a render together with the part and assembly text, with the reconstructions of the selected models, all in OpenSCAD.

Figure A7: **Qualitative reconstructions on the three P3D-BENCH tasks.** Columns are cases; rows are the task input, the ground truth and the selected models. The general-purpose models produce OpenSCAD, while the domain-specific baselines are shown in their native formats. Each output is annotated with the per-case bucket scores; "invalid" marks a prediction that failed to execute and "Part: –" one that failed the *Part* fidelity gate. Sub-figures (a)–(c) cover Assembly-3D, Text-to-3D and Image-to-3D, respectively.



(b) Text-to-3D: three Desc. and three Param. cases, where "Desc." and "Param." denote the descriptive and parametric setting, and the domain-specific baseline is Text2CAD.

Figure A7: (Continued.)



(c) Image-to-3D: each column pairs an input render with the reconstructions of the selected models; the domain-specific baselines are CADRILLE and CAD-Coder.

Figure A7: (Continued.)