PARTGROUND: BENCHMARKING, ANALYZING, AND IMPROVING VLM-BASED 3D RECONSTRUCTION

Anonymous authors
Paper under double-blind review

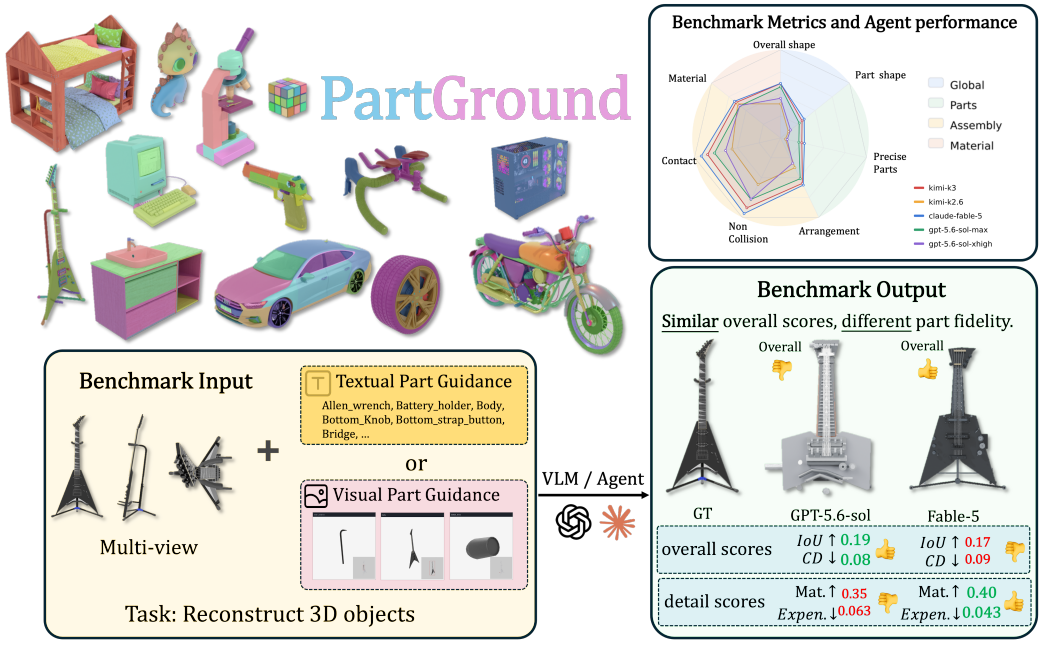

Figure 1: **PartGround-Bench overview.** Reference views and part guidance specify the asset to construct. Evaluation compares the assembled output with an annotated reference, measuring overall geometry, designated components and their materials, and inter-part relationships.

ABSTRACT

Vision-Language Models (VLMs) can generate editable 3D assets through code, complementing diffusion-based methods. Whole-object similarity alone cannot establish component, material, and relational fidelity in an assembled asset. To address this evaluation need, we introduce PartGround-Bench for reconstruction from reference images and supplied part specifications, with 1,195 manually annotated objects selected from 82,717 Blendkit models across 12 category groups. Fixed part correspondences and shared whole-object alignment enable direct scoring of overall shape, part geometry and completeness, per-part materials, and spatial relationships, preserving placement errors. We evaluate nine single-call configurations on the full benchmark and five agent configurations on a subset. Under Visual Part Guidance, two agents achieve similar mean IoU (0.370 and 0.372) but different mean Part-F1 (0.149 and 0.168). Higher reasoning effort accompanies better single-call geometry scores, without uniformly improving materials or reducing inter-part penetration. We further evaluate a reusable multi-agent modeling skill with shared planning and independent review, improving overall and part geometry, layout, and contact consistency across three agent configurations. For Kimi-K3 (max) under Visual Part Guidance, our relative gains are 5.9% in whole-object IoU, 17.4% in Part-F1, 3.2% in layout ΔAUC, and 8.2% in ContactF-AUC, demonstrating the effectiveness of our multi-agent modeling skill in improving geometric and relational fidelity.

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

Diffusion-based methods synthesize 3D geometry and appearance from text or images using learned visual priors (Poole et al., 2023; Lin et al., 2023; Shi et al., 2024; Xu et al., 2024), while Vision-Language Models (VLMs) offer a complementary route by translating references into executable modeling programs (Huang et al., 2024a; He et al., 2026). Programs can expose named components, dimensions, and spatial relationships for targeted edits (Jones et al., 2020; Lin et al., 2026). We ask: *given reference images and specified parts, how faithfully can VLMs realize those components together as a complete 3D asset?* This requires their geometry, materials, and arrangement to agree with the reference.

A generated object may resemble the reference overall while omitting a small component, distorting its geometry, or placing it incorrectly. Benchmarks measure overall agreement through IoU, Chamfer distance, photometric error, and feature similarity (Gu et al., 2025; Gao et al., 2026; Yang et al., 2026a). Assessing part-structured assets therefore requires checking the specified components, their materials, and their relationships alongside overall shape (Figure 1).

Existing part-aware benchmarks already evaluate structure, with differing targets and scoring rules. Hi3DEval predicts object, part, and material quality with learned evaluators (Zhang et al., 2025b). P3D-Bench supplies part descriptions for assembly generation and combines geometric, structural, and part-shape assessment (Yang et al., 2026a). Its part scores use model-assisted decomposition, geometric matching, and independent pose alignment to isolate shape recovery; global metrics and a learned judge also assess the assembled result. PartGround asks whether each specified component is faithfully realized at its intended position in the completed asset. Fixed semantic correspondences and a shared whole-object alignment retain both shape and placement errors in part scores.

In this paper, we introduce **PartGround-Bench**, comprising 1,195 manually annotated objects selected from 82,717 Blendkit models across 12 category groups. Tasks provide overall views and textual or visual part guidance, while source scenes remain hidden for evaluation. Equal part weighting and completeness measurements give small components an explicit contribution. Separate scores assess overall geometry, part geometry and material attributes, as well as inter-part layout, excess penetration, and contact consistency directly against the annotated scenes.

Our experiments find different part fidelity despite similar aggregate whole-object scores. We evaluate nine single-call configurations on all 1,195 objects and five agent configurations on a 185-object subset to limit the cost of iterative modeling. Under Visual Part Guidance, Kimi-K3 (max) and Claude-Fable-5 (high) achieve nearly identical mean IoU (0.370 and 0.372, respectively), yet different mean Part-F1 (0.149 and 0.168, respectively), as shown in Table 1. High execution success and coverage of supplied part names also coexist with low Part-F1. Higher reasoning effort accompanies better single-call geometry scores, but material fidelity and excess penetration do not improve uniformly. Together, these findings show that successful execution and improved geometry do not ensure faithful components, materials, and assembly.

We use PartGround-Bench to evaluate a reusable multi-agent modeling skill that combines shared planning and independent review. Under Visual Part Guidance, all three evaluated agent configurations improve overall and part geometry, layout, and contact consistency. Kimi-K3 (max) gains 17.4% in Part-F1 and 8.2% in ContactF-AUC relative to its baseline. Higher resource use and model-specific material and penetration tradeoffs accompany these gains, as shown in Table 3.

Our contributions are threefold:

A benchmark for part-structured 3D asset reconstruction. We introduce 1,195 manually annotated objects across 12 category groups, with textual and visual part guidance for single-call and agent evaluation. Each task links the supplied references and part specifications to a hidden source scene for direct assessment of the generated asset.

Evaluation of component fidelity within the assembled asset. We score specified parts under fixed semantic correspondences and a shared whole-object alignment, retaining both shape and placement errors and complementing geometry with per-part material and inter-part relation measurements.

Empirical analysis and a reusable modeling skill. We characterize component fidelity across model, reasoning, and guidance settings, then evaluate coordinated part construction, reporting geometric and relational gains while quantifying resource costs as well as material and penetration tradeoffs.

2 Related Work

2.1 3D Object Generation

Learned generative priors support text- and image-conditioned 3D asset creation through diffusion-guided optimization (Poole et al., 2023; Lin et al., 2023), multi-view synthesis and reconstruction (Shi et al., 2024; Xu et al., 2024), or native 3D latent models (Xiang et al., 2025). Part-aware methods produce structured assets through image-conditioned reconstruction (Liu et al., 2024; Chen et al., 2025), amodal completion (Yang et al., 2026b), and compositional mesh generation (Lin et al., 2025).

Language-driven procedural modeling represents assets as executable programs. 3D-GPT and SceneCraft generate procedural instructions or Blender code (Sun et al., 2023; Hu et al., 2024), while BlenderAlchemy and Thinking in Blender incorporate visual references and rendered feedback (Huang et al., 2024a; He et al., 2026). Procedura organizes named parts and their connections with geometric checks and visual review (Lin et al., 2026).

The two routes differ in representation and geometric control. Diffusion priors support detailed geometry and appearance (Lin et al., 2023; Huang et al., 2024b), while procedural programs provide *parametric editability* of component dimensions and spatial relations (He et al., 2026; Lin et al., 2026). Primitive-based modeling and constructive solid geometry (CSG) represent organic, freeform surfaces less naturally (Lin et al., 2026); richer operations extend this range, as illustrated by MeshCoder's point-cloud reconstruction using sweeps and Boolean operations (Dai et al., 2025).

2.2 Evaluation of 3D Generation and Modeling

Geometric and rendered-view comparisons measure overall fidelity through shape distances, occupancy overlap, photometric error, and feature similarity, as in BlenderGym and 3DCodeBench (Gu et al., 2025; Gao et al., 2026). For text-conditioned generation, T^3Bench combines rendered-view quality and consistency with caption-based prompt-alignment assessment (He et al., 2023).

Human and learned judgments assess perceptual quality beyond fixed similarity scores. GPTEval3D uses VLM pairwise comparisons (Wu et al., 2024), 3DGen-Bench learns evaluators from human preferences (Zhang et al., 2025a), and 3D-DefectBench tests VLM judges against annotated defects (Zhao et al., 2026). Hi3DEval predicts object and material quality from video representations and part quality from 3D features (Zhang et al., 2025b).

Structural evaluation examines parts and assembly. Image-guided assembly evaluates supplied parts in predicted poses (Li et al., 2020), CubePart tests mesh-conditioned decomposition (Zhu et al., 2026), and PartCrafter measures whole-object reconstruction and inter-part overlap (Lin et al., 2025). P3D-Bench evaluates parametric assemblies through global geometry, topology, learned judgments, and part-shape matching (Yang et al., 2026a).

Executability and engineering assessments test whether programs run and outputs satisfy functional requirements. 3DCodeBench and P3D-Bench check code or generation validity (Gao et al., 2026; Yang et al., 2026a), while CADEngBench evaluates parametric design, editing, assembly reasoning, and physics simulation (Singh et al., 2026).

These benchmarks establish component and structural evaluation, but their task assumptions and scoring rules answer different questions. Supplied-part assembly and mesh-conditioned decomposition start from geometry that image-conditioned construction must generate. P3D-Bench uses model-assisted decomposition, deduplication, independent part alignment, and geometric matching to assess shape recovery; its global geometry and learned judgments also assess assembled structure (Yang et al., 2026a). PartGround-Bench instead requires explicitly named components and scores them as realized in the complete reference asset. Fixed correspondences and one whole-object alignment preserve both shape and placement errors, while equal part weighting and completeness scores give small components an explicit contribution. Matched-part material comparisons and reference-relative layout, penetration, and contact measurements further describe how the specified asset has been reproduced.

3 PARTGROUND-BENCH

PartGround-Bench evaluates how faithfully models reconstruct complete reference assets, including their specified components, materials, and spatial relationships. Data construction filters and orients source assets, annotates their parts, and produces visual inputs linked to hidden reference scenes, enabling direct comparison of generated objects and their designated components.

3.1 DATASET CURATION AND ANNOTATION

Preprocessing selects single-object assets with usable parts from 82,717 Blendkit models. VLM-assisted inspection of rendered views excludes scenes and multi-object collections. We require at least two geometric parts after removing rig controls, excluding single-mesh assets and rigged assets with fewer than two remaining parts. Scattered components that do not form one assembled target and reference-rendering timeouts also lead to exclusion. Category and part-count sampling cover object types and structural scales. Retained objects are oriented upright along $+Z$ and facing $+X$.

Manual annotation assigns unique, descriptive part names by inspecting each existing component in isolation and within the complete object. VLM-assisted names are verified by humans, written back to the scene, and recorded with material and dimension information. Repeated components may be annotated individually or as groups; completeness follows each asset's annotated units.

Benchmark construction produces visual inputs and hidden evaluation references from the annotated scenes. Each task has front, side, and top views, plus named part close-ups with whole-object context insets highlighting their locations. Assets with blank, poorly framed, misoriented, or distorted renders are corrected or excluded after quality checks. Scene export produces a reference manifest of part identities, geometric bounds, material attributes, and contact edges, checked for consistency between the part inventory and exported geometry. The resulting 1,195 manually annotated objects span 12 category groups. The pipeline is summarized in Figure 2.

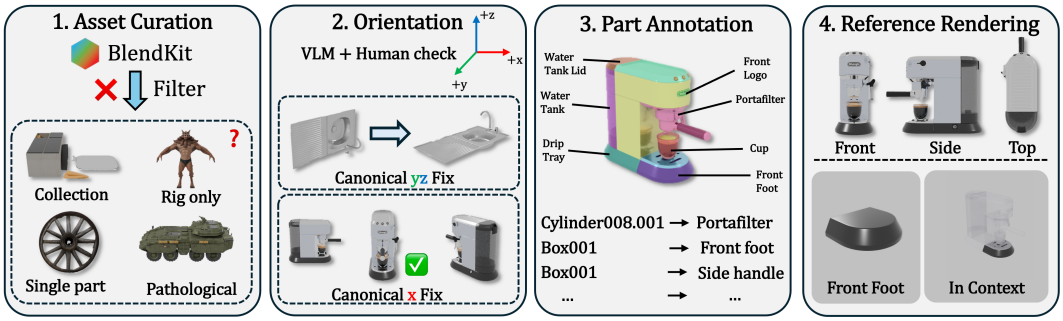

Figure 2: **Data construction pipeline of PartGround-Bench.** Assets are filtered, oriented, semantically annotated, and rendered to produce overall and part-level reference views.

3.2 TASK SETUP AND EVALUATION PROTOCOL

Two execution tracks assess direct generation and iterative modeling under the same reconstruction requirements and scoring rules. The *single-call VLM* track generates a program in one invocation, then executes it without further model revision; the *agent* track allows code execution, inspection of outputs and renders, and revision within a specified budget. Single-call evaluation covers all 1,195 objects, while agent evaluation uses a 185-object subset because of its higher cost (Section 4). Both require a Blender scene with a separate mesh object for each specified part, using its assigned name; source scenes and reference geometry remain hidden.

Two part-guidance settings support construction and assembly from a supplied decomposition in either execution track. Both provide three overall views and semantic part names. *Textual Part Guidance* adds material and approximate dimension hints for planning component geometry, appearance, and proportions. *Visual Part Guidance* instead adds named part close-ups and whole-object context views, revealing local shape and appearance alongside component locations. It provides no textual material or dimension hints.

Evaluation fixes part identity and preserves assembled placement. Names determine the correspondences specified by the task, rather than selecting pairs to maximize geometric similarity. After scale normalization, we align the complete prediction to the reference and apply the same transformation to every part. Global and part-level scores use this shared frame, reducing global pose and scale discrepancies while preserving internal proportions and spatial relationships. No part is independently translated, rotated, or rescaled; a correctly shaped but misplaced component therefore loses overlap credit because its placement in the complete asset remains incorrect.

We report object, part, and inter-part scores for the completed scene to expose errors that whole-object similarity can obscure. These levels cover overall geometry; component completeness, geometry in assembled positions, and matched-part material fidelity; and relative layout, excess penetration, and contact consistency. We average valid task scores, except for Chamfer distance, summarized by its median. Section 3.3 defines the metrics, with implementation details in Appendix A.

3.3 Evaluation Metrics

Let N_P and N_G count predicted and reference parts, with k name-matched pairs $\{(\widetilde{P}_a, G_a)\}_{a=1}^{k}$. All geometry and relationship metrics use the shared alignment from Section 3.2: $\widetilde{P}_a = T(P_a)$, with one whole-object transformation T.

Whole-object geometry. Chamfer distance measures surface agreement and IoU compares voxel occupancy from dilated surface samples. Both use the union of all components, including unmatched predictions, without giving each designated part an equal contribution.

Part geometry and completeness. Part metrics jointly assess component shape and placement in the assembled object. Writing v_a for the in-situ IoU of matched pair $(\widetilde{P}_a, G_a)$, mean part overlap (SQ) and Part-F1 are

$$\mathrm{SQ} = \frac{1}{k}\sum_{a=1}^{k} v_a, \qquad \mathrm{PartF1}(\tau) = \frac{2\sum_{a=1}^{k}\mathbf{1}[v_a \geq \tau]}{N_P + N_G}. \tag{1}$$

SQ weights matched parts equally; Part-F1 also penalizes missing, extra, and below-threshold parts. Both scores are zero when $k = 0$; we report Part-F1 at $\tau = 0.5$.

Part material fidelity. Material fidelity compares representative base color, roughness, and metallicity. For matched pairs $\mathcal{M}_{\mathrm{mat}} \subseteq \{1, \ldots, k\}$ with all three attributes available on both sides, let e_c^a, e_r^a, and e_μ^a denote their normalized discrepancies (Appendix A.4). We compute

$$B_{\mathrm{material}} = \frac{1}{|\mathcal{M}_{\mathrm{mat}}|}\sum_{a\in\mathcal{M}_{\mathrm{mat}}} \left(1 - 0.5e_c^a - 0.3e_r^a - 0.2e_\mu^a\right). \tag{2}$$

Valid parts contribute equally; tasks without valid pairs have an undefined score. This direct comparison of material attributes and texture summaries uses no learned image judge; it does not measure spatial texture patterns or overall perceptual quality.

Relative layout. Relative layout compares pairwise center offsets, size ratios, and orientations through errors e_t^{ab}, e_s^{ab}, and e_R^{ab} over $\mathcal{E} = \{(a, b) : 1 \leq a < b \leq k\}$ (Appendix A.2). Orientation estimates quantify discrepancies without repositioning parts for IoU scoring. Translation and scale use $\mathcal{E}_t = \mathcal{E}_s = \mathcal{E}$; rotation uses $\mathcal{E}_R \subseteq \mathcal{E}$, excluding pairs with unstable orientation estimates. Writing $[x]_+ = \max(0, x)$ and $\mathcal{Q} = \{q \in \{t, s, R\} : |\mathcal{E}_q| > 0\}$, we define

$$\Delta\mathrm{AUC} = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}\frac{1}{|\mathcal{E}_q|}\sum_{(a,b)\in\mathcal{E}_q}\left[1 - \frac{e_q^{ab}}{h_q}\right]_+, \quad (h_t, h_s, h_R) = (0.1, \ln 1.5, 20^\circ). \tag{3}$$

Each component is the normalized area under the empirical cumulative distribution of errors up to h_q; higher is better, and the score is undefined for $k < 2$. Deformation can alter centers, sizes, and orientations, so layout does not fully separate shape from placement errors.

Excess interpenetration. Excess penetration penalizes overlap beyond that already present in the reference assembly. Let ρ_{ab}^{X} denote the intersection of two parts' filled voxel occupancies in scene $X \in \{P, G\}$, normalized by the smaller occupancy (Appendix A.3). We average the excess over all matched-part pairs:

$$\mathrm{ExPen} = \frac{1}{|\mathcal{E}|} \sum_{(a,b)\in\mathcal{E}} \left[\rho_{ab}^{P} - \rho_{ab}^{G} - \epsilon\right]_{+}, \qquad \epsilon = 0.005. \tag{4}$$

Lower is better; reference intersections are unpenalized. The score is undefined for $k < 2$ and excludes unmatched parts, so it is read alongside Part-F1.

Contact consistency. ContactF-AUC compares predicted proximity edges with the reference contact graph, averaging edge F1 over three distance thresholds (Appendix A.5). Relational scores measure geometric arrangement and connections, not load bearing, stability, or functional correctness.

4 Evaluation on PartGround

4.1 Evaluation Setup

We evaluate fidelity to reference assets in the *single-call VLM track* on all 1,195 objects with nine GPT-5.5 and GPT-5.6-Sol reasoning configurations (Table 1). Repeated calls, execution, and visual feedback make agent evaluation more costly, motivating a 185-object subset. We allocate samples evenly across the named categories in Table 2, grouping unlisted types as *Other* to complete the subset.

The five agent configurations in Table 1 use Kimi Code, Claude Code, or Codex CLI for their respective model families. Each completes all 185 tasks under both guidance settings from Section 3.2. Runs allow one visual self-check and one correction within two hours, without an explicit token cap. Appendix B.1 gives execution and prompt details.

Execution success is reported separately from quality, which uses valid scores from successful outputs. We report task-level means except for median CD in metres, with scores and guidance differences rounded to three decimals. Comparisons use the stated per-task limits, without matching actual runtime or token use. Different object sets, tools, and reasoning settings preclude a controlled cross-track comparison. Appendix B.2 reports name recall and RQ/PQ diagnostics.

4.2 Main Results

Similar mean whole-object scores can mask differences in part fidelity. Under Visual Part Guidance, Kimi-K3 (max) and Claude-Fable-5 (high) have nearly identical mean IoU (0.370 and 0.372), yet differ in mean Part-F1 (0.149 and 0.168) and SQ (0.187 and 0.207; Table 1). Median CD favors Kimi-K3 (0.069 versus 0.071), reversing the part-level ranking. These model-level summaries show that similar aggregate geometry can accompany different fidelity of designated components, scored in their assembled positions.

Successful execution and naming do not ensure faithful parts. Single-call execution succeeds on 94.2–97.8% of tasks, yet mean Part-F1@0.5 ranges from 0.060 to 0.183. On the separate agent subset, every run executes successfully and name recall reaches 0.982–0.986, but Part-F1 remains 0.021–0.168 (Appendix B.2). Since names are supplied as input, recall indicates that requested components are instantiated, not independently recognized. Low Part-F1 despite high recall indicates that instantiating the requested inventory leaves substantial errors in component geometry or placement.

Geometric gains with higher reasoning effort do not ensure better assembly. Across both single-call model families and both guidance settings, IoU, SQ, and Part-F1 increase monotonically with reasoning effort, while median CD decreases or remains unchanged. For GPT-5.6-Sol under Visual guidance, moving from high to max effort raises Part-F1 from 0.113 to 0.183 and lowers CD from 0.071 to 0.055. However, ExPen rises from 0.047 to 0.064: stronger geometric agreement coexists with more inter-part overlap beyond that present in the reference. Better agreement with reference parts therefore does not ensure faithful relationships in the assembled asset.

Table 1: Single-call VLM results on 1,195 objects and agent results on the 185-object subset. Exec. reports successful program execution; quality metrics use valid scores from successful outputs. CD is the median in metres; other metrics are means. Mat. is B_{material}, ContactF is ContactF-AUC, and Part-F1 uses IoU ≥ 0.5. Best reconstruction scores within each track and setting are bold.

		Exec.	Object		Part			Inter-part		
Configuration	Setting	(%) ↑	IoU ↑	CD ↓	SQ ↑	Part-F1 ↑	Mat. ↑	ΔAUC ↑	ExPen ↓	ContactF ↑
Single-call VLM: full benchmark (1,195 objects)										
GPT-5.5 (low)	Textual	95.8	0.237	0.098	0.124	0.060	**0.391**	0.264	0.076	0.214
GPT-5.5 (low)	Visual	94.9	0.244	0.094	0.142	0.070	0.394	0.259	0.067	0.237
GPT-5.5 (medium)	Textual	96.9	0.286	0.084	0.162	0.087	**0.391**	0.294	0.060	0.228
GPT-5.5 (medium)	Visual	96.5	0.283	0.082	0.171	0.096	**0.400**	0.286	0.052	0.263
GPT-5.5 (high)	Textual	95.4	0.294	0.072	0.174	0.099	0.383	0.310	0.060	0.240
GPT-5.5 (high)	Visual	97.6	0.301	0.075	0.188	0.098	0.389	0.291	0.053	0.266
GPT-5.5 (xhigh)	Textual	95.8	0.304	0.072	0.192	0.113	0.367	0.325	0.063	0.266
GPT-5.5 (xhigh)	Visual	96.4	0.303	0.067	0.199	0.112	0.384	0.310	0.055	0.263
GPT-5.6-Sol (low)	Textual	95.3	0.277	0.089	0.155	0.084	0.378	0.281	0.057	0.231
GPT-5.6-Sol (low)	Visual	95.4	0.279	0.084	0.164	0.084	0.385	0.279	**0.047**	0.267
GPT-5.6-Sol (medium)	Textual	97.8	0.295	0.078	0.182	0.105	0.378	0.312	0.052	0.258
GPT-5.6-Sol (medium)	Visual	94.4	0.293	0.078	0.181	0.110	0.390	0.297	0.050	0.267
GPT-5.6-Sol (high)	Textual	94.2	0.312	0.067	0.189	0.120	0.385	0.330	**0.050**	0.271
GPT-5.6-Sol (high)	Visual	94.5	0.303	0.071	0.190	0.113	0.389	0.314	**0.047**	0.283
GPT-5.6-Sol (xhigh)	Textual	94.4	0.342	0.060	0.227	0.169	0.371	0.353	0.073	0.276
GPT-5.6-Sol (xhigh)	Visual	94.8	0.329	0.066	0.215	0.154	0.379	0.316	0.052	0.289
GPT-5.6-Sol (max)	Textual	94.7	**0.344**	**0.057**	**0.245**	**0.178**	0.390	**0.383**	0.071	**0.277**
GPT-5.6-Sol (max)	Visual	95.8	**0.345**	**0.055**	**0.307**	**0.183**	**0.400**	**0.382**	0.064	**0.319**
Agent: subset of 185 objects										
Kimi-K2.6 (thinking)	Textual	100.0	0.211	0.149	0.055	0.021	0.382	0.158	0.066	0.096
Kimi-K2.6 (thinking)	Visual	100.0	0.232	0.129	0.069	0.034	0.386	0.225	0.065	0.107
Kimi-K3 (max)	Textual	100.0	0.339	0.080	0.136	0.090	0.424	0.256	0.059	0.190
Kimi-K3 (max)	Visual	100.0	0.370	**0.069**	0.187	0.149	0.435	0.343	0.047	**0.255**
Claude-Fable-5 (high)	Textual	100.0	**0.352**	**0.074**	**0.148**	**0.100**	**0.435**	**0.325**	**0.052**	0.203
Claude-Fable-5 (high)	Visual	100.0	**0.372**	0.071	**0.207**	**0.168**	**0.452**	**0.358**	**0.043**	0.246
GPT-5.6-Sol (xhigh)	Textual	100.0	0.255	0.119	0.070	0.034	0.398	0.165	0.054	0.140
GPT-5.6-Sol (xhigh)	Visual	100.0	0.268	0.114	0.084	0.043	0.404	0.186	0.054	0.170
GPT-5.6-Sol (max)	Textual	100.0	0.346	0.084	0.136	0.095	0.395	0.274	0.058	**0.232**
GPT-5.6-Sol (max)	Visual	100.0	0.349	0.085	0.167	0.134	0.404	0.315	0.054	0.253

Material fidelity follows a different trend from geometry. In the single-call GPT-5.5 sweep under Textual guidance, Part-F1 rises from 0.060 at low effort to 0.113 at xhigh, while the material score falls from 0.391 to 0.367. These scores compare matched-part color, roughness, and metallicity, rather than complete texture appearance. Improved component geometry therefore does not imply more faithful material attributes.

Object categories reveal distinct difficulties and model strengths. Under Visual Part Guidance, all five agents attain their highest category-level Part-F1 on furniture or architectural components (Table 2). Animals average fewer annotated parts than furniture (7.4 versus 9.6), yet score lower for every agent (Part-F1 of 0.000–0.061), showing that part count alone does not explain reconstruction difficulty. Model rankings also vary: Kimi-K3 (max) leads Claude-Fable-5 (high) on electronics and appliances in Part-F1 (0.113 versus 0.065) and IoU (0.393 versus 0.317), whereas Claude leads on vehicles (Part-F1: 0.130 versus 0.071; IoU: 0.399 versus 0.349). Category-level evaluation thus reveals differences in model strengths that aggregate rankings conceal.

4.3 Textual versus Visual Part Guidance

We compare the two input settings from Section 3.2, replacing textual material and dimension hints with part close-ups and contextual views.

Table 2: Category-level results on the 185-object agent subset under Visual Part Guidance. Each cell reports IoU / Part-F1 above Mat. / ΔAUC (all ↑). Parts gives the mean annotated part count; bold marks the best score per category and metric.

Category	Parts	Kimi-K2.6 (thinking)	Kimi-K3 (max)	Claude-Fable-5 (high)	GPT-5.6-Sol (xhigh)	GPT-5.6-Sol (max)
Furniture	9.6	0.235 / 0.052 0.417 / 0.278	**0.426** / 0.273 0.435 / 0.428	**0.426** / **0.323** **0.478** / **0.465**	0.278 / 0.120 0.414 / 0.296	0.421 / 0.266 0.386 / 0.439
Vehicles	35.5	0.278 / 0.039 0.368 / 0.167	0.349 / 0.071 0.424 / 0.242	**0.399** / **0.130** **0.449** / **0.304**	0.283 / 0.012 0.405 / 0.112	0.338 / 0.034 0.398 / 0.192
Electronics & appliances	18.5	0.203 / 0.008 0.396 / 0.218	**0.393** / **0.113** **0.507** / **0.340**	0.317 / 0.065 0.492 / 0.278	0.289 / 0.020 0.428 / 0.134	0.311 / 0.093 0.462 / 0.281
Tools & equipment	25.6	0.266 / 0.045 0.362 / 0.219	0.354 / 0.095 0.370 / 0.389	**0.370** / 0.141 0.388 / 0.416	0.292 / 0.080 0.360 / 0.259	0.356 / **0.187** **0.390** / **0.437**
Architectural components	11.8	0.314 / 0.056 0.355 / 0.253	0.487 / 0.267 0.434 / **0.366**	0.431 / 0.271 **0.457** / 0.352	0.458 / 0.161 0.393 / 0.268	**0.501** / **0.278** 0.398 / **0.366**
Animals	7.4	0.158 / 0.000 0.387 / 0.295	0.184 / 0.044 0.466 / 0.280	0.221 / 0.008 0.458 / **0.300**	0.182 / 0.000 0.416 / 0.203	**0.257** / **0.061** **0.468** / 0.255
Household items	8.0	0.162 / 0.000 0.391 / 0.194	**0.472** / **0.208** **0.421** / 0.363	0.350 / 0.186 0.419 / **0.411**	0.276 / 0.006 0.406 / 0.160	0.263 / 0.037 0.407 / 0.227
Lighting	9.7	0.171 / 0.055 0.393 / 0.179	0.354 / 0.103 **0.450** / 0.391	0.345 / **0.205** 0.391 / **0.438**	0.194 / 0.013 0.415 / 0.217	**0.363** / 0.110 0.397 / 0.401
Outdoor & landscape	21.9	0.230 / 0.055 0.394 / 0.150	**0.291** / **0.182** **0.446** / 0.338	0.262 / 0.118 0.423 / **0.347**	0.180 / 0.033 0.384 / 0.147	0.271 / 0.107 0.391 / 0.312
Weapons & military	28.1	0.191 / 0.000 0.379 / 0.150	0.367 / 0.072 0.358 / 0.273	**0.464** / 0.188 **0.383** / **0.311**	0.186 / 0.000 0.354 / 0.115	0.381 / **0.217** 0.368 / 0.279
Humanoid characters	8.2	0.192 / 0.042 0.444 / 0.196	0.263 / **0.167** 0.480 / 0.361	**0.395** / 0.156 0.487 / **0.429**	0.219 / 0.021 **0.514** / 0.163	0.275 / 0.021 0.499 / 0.388
Other	29.3	0.221 / 0.034 0.386 / 0.250	**0.370** / 0.166 0.429 / **0.364**	**0.370** / **0.170** **0.449** / 0.360	0.248 / 0.030 0.398 / 0.183	0.343 / 0.155 0.385 / 0.330

Table 3: Skill comparison under Visual Part Guidance on 185 objects. The w/o (without skill) baselines match Table 1; w/ denotes with skill. CD is in metres; time and tokens are per object. Δ is with minus without skill: a difference of medians for CD and means otherwise.

Configuration	Skill	Object		Part			Inter-part			Resources	
		IoU ↑	CD ↓	SQ ↑	Part-F1 ↑	Mat. ↑	ΔAUC ↑	ExPen ↓	ContactF ↑	Time (min)	Tokens (K)
Kimi-K3 (max)	w/o	0.370	0.069	0.187	0.149	0.435	0.343	0.047	0.255	18.4	228.1
	w/	0.392	0.068	0.211	0.175	0.438	0.354	0.043	0.276	21.2	321.8
	Δ	+0.022	–0.001	+0.024	+0.026	+0.003	+0.011	–0.004	+0.021	+2.8	+93.7
Claude-Fable-5 (high)	w/o	0.372	0.071	0.207	0.168	0.452	0.358	0.043	0.246	12.6	260.3
	w/	0.401	0.067	0.213	0.171	0.450	0.361	0.040	0.259	14.4	353.9
	Δ	+0.029	–0.004	+0.006	+0.003	–0.002	+0.003	–0.003	+0.013	+1.8	+93.6
GPT-5.6-Sol (max)	w/o	0.349	0.085	0.167	0.134	0.404	0.315	0.054	0.253	24.6	205.3
	w/	0.354	0.077	0.175	0.136	0.421	0.332	0.063	0.273	27.1	290.5
	Δ	+0.005	–0.008	+0.008	+0.002	+0.017	+0.017	+0.009	+0.020	+2.5	+85.2

Visual guidance accompanies consistent agent gains in part fidelity. All five agents score higher in IoU, SQ, Part-F1, material fidelity, ΔAUC, and ContactF with Visual than Textual guidance (Table 1). Part-F1 gains range from +0.009 to +0.068; Kimi-K3 (max) has the largest layout and contact gains (+0.087 and +0.065). Exceptions are slightly worse CD for GPT-5.6-Sol (max), 0.084 to 0.085, and unchanged ExPen for GPT-5.6-Sol (xhigh).

Single-call part and layout scores respond differently. Visual guidance raises SQ in seven of nine single-call configurations, whereas Part-F1 rises in four, falls in four, and is unchanged in one. Layout ΔAUC falls in all nine by 0.001–0.037. For GPT-5.6-Sol (max), SQ rises from 0.245 to 0.307, while Part-F1 changes from 0.178 to 0.183 and ΔAUC from 0.383 to 0.382, separating average overlap gains from thresholded recovery and layout agreement. Single-call quality averages use successful outputs, so differences in execution success can also change the set of tasks contributing to each mean.

4.4 Qualitative Analysis and Skill Evaluation

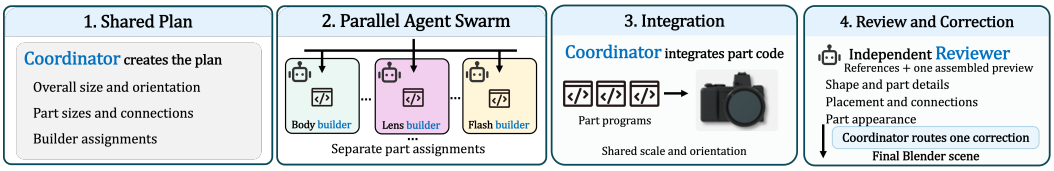

Figure 3: **Multi-agent modeling workflow.** The coordinator defines a shared plan, assigns part builders, and integrates their outputs. An independent reviewer checks the assembled preview before one targeted correction.

Multi-agent modeling skill. We evaluate a reusable Blender modeling skill (Figure 3), inspired by Procedura's structured construction and separate visual critic (Lin et al., 2026). A coordinator sets a shared plan, assigns disjoint part groups to builders, and integrates their geometry and materials in a common frame. An independent reviewer inspects the assembly and proposes a targeted correction within the task's feedback budget. Appendix C details the roles, handoffs, and budget controls.

Qualitative comparison. Figure 4 shows local improvements in four selected Kimi-K3 (max) examples. With the skill, the camera gains lens-ring and top-control detail, and the pistol better matches the reference slide and grip. The radial object has a more compact arrangement of repeated parts; the signboard better reproduces the face pattern and slender rear support. Residual errors remain visible, including the lighter signboard frame and the pistol's inaccurate proportions and color.

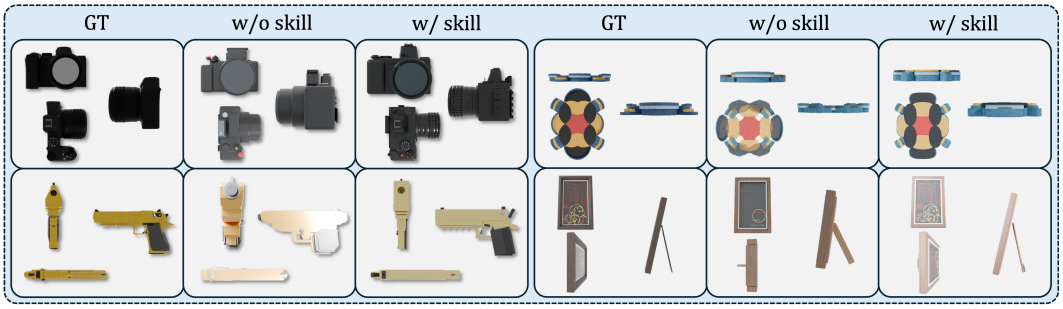

Figure 4: **Qualitative skill comparison.** Kimi-K3 (max) in Kimi Code, with and without the skill under Visual Part Guidance.

Evaluation across models and metrics. All three skill configurations improve IoU, SQ, Part-F1, ΔAUC, and ContactF and lower median CD under Visual Part Guidance (Table 3). Part-F1 gains are +0.026 for Kimi-K3 (max), +0.003 for Claude-Fable-5 (high), and +0.002 for GPT-5.6-Sol (max). Material and penetration gains remain uneven: Claude's material score falls (0.452 to 0.450), while GPT's ExPen rises (0.054 to 0.063).

Across configurations, the skill adds 1.8–2.8 minutes and 85.2K–93.7K tokens per object. For Kimi-K3, the largest relative Part-F1 gain (17.4%) accompanies increases from 18.4 to 21.2 minutes and from 228.1K to 321.8K tokens. These comparisons measure workflow gains and costs without isolating coordination from additional computation.

5 Conclusion

PartGround-Bench evaluates reference-based asset construction on 1,195 annotated objects with supplied part specifications. Fixed part correspondences and shared alignment support direct checks of component geometry, materials, and relationships in the final scene. Experiments show different part fidelity despite similar mean whole-object scores. Stronger reasoning and the modeling skill improve geometry without uniformly improving materials or assembly; skill gains also require more resources. PartGround supports assessing modeling progress through the fidelity of designated components and their relationships alongside overall geometry.

AI USE STATEMENT

During manuscript preparation, generative AI tools were used solely to polish the writing, including grammar, wording, and readability. The authors reviewed the AI-assisted edits and take full responsibility for the final manuscript. The use of VLMs in benchmark construction and reconstruction experiments is described in Sections 3.1 and 4.

ETHICS STATEMENT

This work focuses on evaluating and improving part-structured 3D reconstruction. To the best of our knowledge, it raises no specific ethical concerns beyond those generally associated with machine learning research.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 3 describes the dataset curation and annotation procedures, task inputs, evaluation protocol, and metrics. Section 4.1 specifies the evaluated models, reasoning settings, execution budgets, and score aggregation; Appendix B.1 provides execution and prompt details. Appendix A provides additional metric implementation details, including sampling procedures, alignment settings, and numerical thresholds, while Appendix C documents the multi-agent modeling protocol. An anonymized package containing the source code, prompt templates, and ground-truth manifests is provided in the supplementary material. The repository will be made public upon publication.

REFERENCES

Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi. PartGen: Part-level 3D Generation and Reconstruction with Multi-view Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5881–5892, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Chen_PartGen_Part-level_3D_Generation_and_Reconstruction_with_Multi-view_Diffusion_Models_CVPR_2025_paper.html.

Bingquan Dai, Li Ray Luo, Qihong Tang, Jie Wang, Xinyu Lian, Hao Xu, Minghan Qin, Xudong Xu, Bo Dai, Haoqian Wang, Zhaoyang Lyu, and Jiangmiao Pang. MeshCoder: LLM-Powered Structured Mesh Code Generation from Point Clouds. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-0301. URL https://papers.neurips.cc/paper_files/paper/2025/hash/0d094d83efa5ea99f86527ac7207e023-Abstract-Conference.html.

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. arXiv preprint arXiv:2606.01057, 2026. URL https://arxiv.org/abs/2606.01057.

Yunqi Gu, Ian Huang, Jihyeon Je, Guandao Yang, and Leonidas Guibas. BlenderGym: Benchmarking Foundational Model Systems for Graphics Editing. *arXiv preprint arXiv:2504.01786*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gu_BlenderGym_Benchmarking_Foundational_Model_Systems_for_Graphics_Editing_CVPR_2025_paper.html. CVPR 2025.

Guangzhao He, Rundong Luo, Wei-Chiu Ma, and Hadar Averbuch-Elor. Thinking in Blender: Staged Executable Inverse Graphics with Vision-Language Models. *arXiv preprint arXiv:2606.02580*, 2026. URL https://arxiv.org/abs/2606.02580.

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. arXiv preprint arXiv:2310.02977, 2023. URL https://arxiv.org/abs/2310.02977.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A. Ross, Cordelia Schmid, and Alireza Fathi. SceneCraft: An LLM Agent for Synthesizing 3D Scenes as Blender Code. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19252–19282. PMLR, 2024. URL https://proceedings.mlr.press/v235/hu24g.html.

Ian Huang, Guandao Yang, and Leonidas Guibas. BlenderAlchemy: Editing 3D Graphics with Vision-Language Models. In *European Conference on Computer Vision*, 2024a. URL https://arxiv.org/abs/2404.17672.

Xin Huang, Ruizhi Shao, Qi Zhang, Hongwen Zhang, Ying Feng, Yebin Liu, and Qing Wang. HumanNorm: Learning Normal Diffusion Model for High-quality and Realistic 3D Human Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4568–4577, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_HumanNorm_Learning_Normal_Diffusion_Model_for_High-quality_and_Realistic_3D_CVPR_2024_paper.html.

R. Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J. Mitra, and Daniel Ritchie. ShapeAssembly: Learning to Generate Programs for 3D Shape Structure Synthesis. *ACM Transactions on Graphics*, 39(6):1–20, 2020. doi: 10.1145/3414685.3417812. URL https://arxiv.org/abs/2009.08026.

Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3D Part Assembly from a Single Image. In *Computer Vision – ECCV 2020*, pp. 664–682. Springer International Publishing, 2020. doi: 10.1007/978-3-030-58539-6_40. URL https://doi.org/10.1007/978-3-030-58539-6_40.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-Resolution Text-to-3D Content Creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 300–309, 2023. URL https://arxiv.org/abs/2211.10440.

Youtian Lin, Yikang Yang, Zhanpeng Hu, Mengqi Zhou, Feihu Zhang, Xun Cao, Jiaheng Liu, and Yao Yao. Procedura: Agentic 3D Modeling with Procedural Control. *arXiv preprint arXiv:2608.26238*, 2026. URL https://arxiv.org/abs/2608.26238.

Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. PartCrafter: Structured 3D Mesh Generation via Compositional Latent Diffusion Transformers. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.05573.

Anran Liu, Cheng Lin, Yuan Liu, Xiaoxiao Long, Zhiyang Dou, Hao-Xiang Guo, Ping Luo, and Wenping Wang. Part123: Part-aware 3D Reconstruction from a Single-view Image. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–12, 2024. URL https://arxiv.org/abs/2405.16888.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.14988.

Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations. *Color Research & Application*, 30(1):21–30, 2005. doi: 10.1002/col.20070. URL https://hajim.rochester.edu/ece/sites/gsharma/ciede2000/ciede2000noteCRNA.pdf.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view Diffusion for 3D Generation. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.16512.

Harmanjot Singh, Abhra Dubey, and Jorge Alejandro Amador Herrera. CADEngBench: It Looks Like CAD, but Does It Work? Evaluating Parametric Design, Assembly Reasoning, and Physics Simulation. *arXiv preprint arXiv:2608.09296*, 2026. URL https://arxiv.org/abs/2608.09296.

Chunyi Sun, Junlin Han, Weijian Deng, Xinlong Wang, Zishan Qin, and Stephen Gould. 3D-GPT: Procedural 3D Modeling with Large Language Models. arXiv preprint arXiv:2310.12945, 2023. URL https://arxiv.org/abs/2310.12945.

Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. GPT-4V(ision) is a Human-Aligned Evaluator for Text-to-3D Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22227–22238, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wu_GPT-4Vision_is_a_Human-Aligned_Evaluator_for_Text-to-3D_Generation_CVPR_2024_paper.html.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3D Latents for Scalable and Versatile 3D Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2412.01506.

Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. InstantMesh: Efficient 3D Mesh Generation from a Single Image with Sparse-view Large Reconstruction Models. arXiv preprint arXiv:2404.07191, 2024. URL https://arxiv.org/abs/2404.07191.

Yikang Yang, Zhanpeng Hu, Youtian Lin, Mengqi Zhou, Jingxi Xu, Feihu Zhang, Jiaheng Liu, and Yao Yao. P3D-Bench: Benchmarking MLLMs for Parametric 3D Generation and Structural Reasoning. *arXiv preprint arXiv:2606.11152*, 2026a.

Yunhan Yang, Yuan-Chen Guo, Yukun Huang, Zi-Xin Zou, Zhipeng Yu, Yangguang Li, Yan-Pei Cao, and Xihui Liu. HoloPart: Generative 3D Part Amodal Segmentation. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2504.07943.

Yuhan Zhang, Mengchen Zhang, Tong Wu, Tengfei Wang, Gordon Wetzstein, Dahua Lin, and Ziwei Liu. 3DGen-Bench: Comprehensive Benchmark Suite for 3D Generative Models. arXiv preprint arXiv:2503.21745, 2025a. URL https://arxiv.org/abs/2503.21745.

Yuhan Zhang, Long Zhuo, Ziyang Chu, Tong Wu, Zhibing Li, Liang Pan, Dahua Lin, and Ziwei Liu. Hi3DEval: Advancing 3D Generation Evaluation with Hierarchical Validity. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025b. doi: 10.52202/085713-1564. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/42ffaddcc6edc9fb05ff9f9b49fca700-Paper-Datasets_and_Benchmarks_Track.pdf.

Zhenyu Zhao, Nanshan Jia, Jihyeon Je, Yifu Tang, Alvin Chan, Michael Spedden, Michael V. Palleschi, Sui Huang, Jingshen Wang, and Zeyu Zheng. 3D-DefectBench: A Controlled Factorial Study of Vision-Language Model Evaluation Pipelines for Fine-Grained 3D Generation Defects. *arXiv preprint arXiv:2607.10826*, 2026.

Yiheng Zhu, Kangle Deng, Jean-Philippe Fauconnier, Inaki Navarro, Daiqing Li, Ava Pun, Yinan Zhang, Peiye Zhuang, Xiaoxia Sun, Maneesh Agrawala, Kiran Bhat, and Tinghui Zhou. CubePart: An Open-Vocabulary Part-Controllable 3D Generator. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12. ACM, 2026. doi: 10.1145/3799902.3811117. URL https://doi.org/10.1145/3799902.3811117.

A Additional Metric Definitions

A.1 Whole-Object Alignment and Global Measurements

Evaluation uses the Y-up coordinate frame of the exported scenes and the reference asset's recorded diagonal D, in metres. For each scene $X \in \{P, G\}$, let b_X^- and b_X^+ be the bounds of the complete object. We center it horizontally, place its lowest point at $y = 0$, and uniformly scale its diagonal to D:

$$o_X = \left(\frac{b_{X,x}^- + b_{X,x}^+}{2},\ b_{X,y}^-,\ \frac{b_{X,z}^- + b_{X,z}^+}{2} \right), \qquad \mathcal{N}_X(x) = \frac{D}{\|b_X^+ - b_X^-\|_2}(x - o_X).$$

The target scale is specific to each reference asset, rather than a common unit diagonal across the benchmark.

The complete prediction is then aligned to the reference using a single global rotation and translation, applied unchanged to all predicted parts. Writing this shared transformation as T, corresponding normalized parts P_i and G_i are compared through $\mathrm{IoU}(T(P_i), G_i)$, with no independently optimized part transformations. The aligned prediction is also used for global and relational measurements, preserving internal geometry and spatial relationships throughout scoring. The reference implementation estimates this transformation from 2000 surface samples per object with fixed seeds. It initializes rigid registration with the identity pose and proper rotations formed by axis permutations and sign choices in both the scene and principal-component frames. Each initialization is refined by bidirectional nearest-neighbor iterative closest point alignment for at most 30 iterations, stopping when the change in symmetric Chamfer distance is below 10^{-7} times the reference sample bounding-box diagonal. The lowest-distance candidate, including the unchanged input, supplies the shared transformation; reflection and additional scale fitting are excluded. This numerical search estimates a global pose without using part correspondences and may remain ambiguous for symmetric shapes. Because the preceding scale normalization uses axis-aligned bounds, its scale factor can also depend on the object's initial orientation.

Let P, Q contain 8000 surface samples each, drawn with fixed random seeds. Chamfer distance sums the two mean nearest-neighbor Euclidean distances:

$$\mathrm{CD}(P, Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p - q\|_2 + \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \|q - p\|_2.$$

CD is expressed in metres: distances are not squared, divided by two, or further divided by D. We report the median across objects to limit the influence of extreme distances; this aggregation does not remove dependence on reference-object size. Geometric F-score uses thresholds of $0.01D$, $0.02D$, and $0.05D$. Global and in-situ part IoU use surface-sample voxel occupancy with voxel size $D/64$ and one dilation step, approximating surface-occupancy overlap. Each matched part pair is compared on a common voxel grid. Sampling density, resolution, and dilation can affect thin-part measurements.

A.2 Relative Layout Score

For scene $X \in \{P, G\}$, let c_a^X and d_a^X denote part a's surface-sample center and axis-aligned bounding-box diagonal. Let R_a estimate the rotation from the centered reference part to its aligned prediction, and let $\theta(\cdot)$ return a rotation's geodesic angle. The errors used in Equation 3 are

$$\begin{aligned} e_t^{ab} &= \frac{\|(c_a^P - c_b^P) - (c_a^G - c_b^G)\|_2}{D}, \\ e_s^{ab} &= \left| \ln \frac{d_a^P / d_b^P}{d_a^G / d_b^G} \right|, \qquad e_R^{ab} = \theta(R_a R_b^\top). \end{aligned} \tag{5}$$

Each part center is estimated from the mean of 2000 surface samples. Rotations R_a are estimated through principal-component initialization followed by ten iterations of iterative closest point alignment. Initialization selects the minimum-Chamfer candidate among four sign choices that preserve

a positive rotation determinant. These per-part rotation estimates measure orientation discrepancies; they are not applied to the geometry used for IoU scoring. Eigenvalue-ratio rules exclude unstable rotation components for near-spherical and near-cylindrical parts; not all symmetries are covered.

The truncated aggregation in Equation 3 is equivalent to a normalized empirical CDF integral. For each valid component q, define

$$F_q(u) = \frac{1}{|\mathcal{E}_q|} \sum_{(a,b)\in\mathcal{E}_q} \mathbf{1}[e_q^{ab} \le u].$$

Then

$$A_q = \frac{1}{h_q}\int_0^{h_q} F_q(u)\,du = \frac{1}{|\mathcal{E}_q|} \sum_{(a,b)\in\mathcal{E}_q} \left[1 - \frac{e_q^{ab}}{h_q}\right]_+, \qquad \Delta\mathrm{AUC} = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} A_q.$$

Thus each valid error component receives equal weight, regardless of its valid pair count. Translation and scale remain usable when rotation is unstable; the layout score is undefined with fewer than two matched parts.

A.3 Excess Penetration

For Equation 4, let V_a^X denote the approximate filled voxel occupancy of part a in scene X, distinct from the surface occupancy used for IoU. The normalized intersection is

$$\rho_{ab}^X = \frac{|V_a^X \cap V_b^X|}{\max(1, \min(|V_a^X|, |V_b^X|))}.$$

All parts within each scene use a common grid. Voxels have side length $D/64$; occupancy is constructed from 4000 surface samples per part, one dilation step, and solid filling. If filling fails or produces a volume exceeding 95% of the bounding box, occupancy uses two dilations of the surface representation instead. The denominator of the task average includes every pair of matched parts, including pairs with zero excess overlap.

A.4 Part Material Fidelity

Equation 2 uses matched pairs with usable base-color, roughness, and metallicity attributes on both sides. For pair a, let d_c^a denote the CIEDE2000 color difference after converting sRGB base colors to CIELAB under the D65 white point, r_a^X roughness, and $\mu_a^X \in [0,1]$ metallicity. With $\bar{r}_a^X = \mathrm{clip}(r_a^X, 0.02, 1)$, the attribute errors are

$$e_c^a = \min\left(\frac{d_c^a}{5}, 1\right), \qquad e_r^a = \min\left(\frac{|\log_2(\bar{r}_a^P/\bar{r}_a^G)|}{2}, 1\right), \qquad e_\mu^a = |\mu_a^P - \mu_a^G|. \tag{6}$$

Color and roughness errors saturate at their respective normalization thresholds. We use the complete CIEDE2000 definition with unit parametric factors (Sharma et al., 2005):

$$d_c^a = \sqrt{\left(\frac{\Delta L'}{S_L}\right)^2 + \left(\frac{\Delta C'}{S_C}\right)^2 + \left(\frac{\Delta H'}{S_H}\right)^2 + R_T \frac{\Delta C'}{S_C}\frac{\Delta H'}{S_H}}.$$

The primed differences incorporate the standard chroma and hue corrections, and S_L, S_C, S_H, R_T are the CIEDE2000 weighting and rotation terms.

Material attributes summarize one representative material per component. The reference uses the first material name recorded for the part in its manifest. For the prediction, we prefer a textured material; otherwise, we select the first material with nondefault parameters, falling back to the first material. Texture-linked reference attributes are resolved from the exported material, preferring the recorded material name. Textures are downsampled to a maximum side length of 128 pixels, and channel means are multiplied by the corresponding material factors: RGB for base color, green for roughness, and blue for metallicity. The same texture reduction updates available attributes of textured predictions; prediction attributes that cannot be resolved retain their material factors. This reduction does not preserve spatial texture patterns. Extraction may use default material factors, so available attributes do not imply that all material parameters were explicitly assigned. The primary score weights valid parts equally; the evaluator additionally records a reference-area-weighted average and separate attribute errors.

A.5 CONTACT CONSISTENCY

The reference contact graph E_G is constructed once from the exported reference geometry. For each part, we draw 1500 surface points using area-weighted triangle selection and uniform within-triangle sampling with random seed zero. An undirected edge joins two parts when the minimum distance between their sampled point sets is below $\epsilon_{\text{contact}} = 0.005D$. The reference graph and its contact threshold are stored in the task manifest. This reference graph remains fixed across all prediction thresholds.

Predicted contact edges use minimum distances between subsampled mesh vertices. We select at most 300 vertices per part at evenly spaced indices in the mesh's vertex ordering. For $\eta \in H = \{0.0025, 0.005, 0.01\}$, an edge belongs to $E_P(\eta D)$ when this minimum distance is below ηD. Both predicted parts must have semantic correspondences, and their edge is expressed using the reference part identities. The evaluator reports

$$\text{ContactF-AUC} = \frac{1}{|H|} \sum_{\eta \in H} \frac{2|E_P(\eta D) \cap E_G|}{|E_P(\eta D)| + |E_G|}. \tag{7}$$

This is a discrete arithmetic mean of edge F1 scores, rather than a continuous area integral. It differs from the auxiliary single-threshold contact F1 computed at the stored reference threshold. The full reference edge set remains in the denominator, so edges incident to missing parts contribute false negatives. Tasks with fewer than five reference edges are excluded from this measure. Finite surface sampling and vertex subsampling make this proximity estimate sensitive to sampling density and mesh tessellation.

B Additional Evaluation Details

B.1 Execution Settings

Single-call evaluations were conducted from September 19 to 21, 2026, through direct model API requests, without an agent tool loop. Each response has a configured output allowance of 65,536 tokens and a two-hour request timeout, with no separately configured reasoning-token or task-level total-token budget. Failed requests are retried up to eight times with exponential backoff; retries do not provide modeling feedback or revise a successfully returned program. Independent tasks use eight concurrent requests per configuration, six concurrent Blender executions, and sixteen concurrent evaluation processes.

The main agent runs were conducted on September 9–10, 2026, using Claude Code, Codex CLI version 0.147.0, and Kimi Code for the respective model families. Reasoning settings are those reported in Table 1. Each task has a shared two-hour wall-clock limit and no explicit token cap; Codex additionally permits up to thirty minutes of stream inactivity. Agent tasks are dispatched independently, without a uniform cap on the number of simultaneous tasks. The visual feedback allowance remains one self-check followed by one correction.

Prompts instantiate the inputs and named-mesh output requirements defined in Section 3.2 for each object.

B.2 Agent Correspondence Diagnostics

Table 4 reports auxiliary diagnostics on the same 185-object agent subset. With k matched part names, N_P predicted parts, and N_G reference parts, the evaluator computes

$$\mathrm{NameRecall} = \frac{k}{\max(1, N_G)}, \qquad \mathrm{RQ} = \frac{2k}{\max(1, N_P + N_G)}, \qquad \mathrm{PQ} = \mathrm{SQ}\,\mathrm{RQ}\,.$$

Name recall measures the presence of a counterpart for each reference name; RQ also penalizes extra predicted parts. Neither applies an IoU threshold, so these correspondence diagnostics must not be interpreted as geometric reconstruction accuracy. PQ is computed within each task before averaging, and its reported mean need not equal the product of the mean SQ and mean RQ. All configurations complete execution and scoring for 185/185 objects in each input setting.

Table 4: Auxiliary agent diagnostics, averaged over the 185 scored tasks per configuration and setting. Name recall and RQ measure semantic correspondence; PQ combines correspondence with in-situ overlap.

		Part diagnostics		
Configuration	Setting	Name recall ↑	RQ ↑	PQ ↑
Kimi-K2.6 (thinking)	Textual	0.985	0.966	0.054
Kimi-K2.6 (thinking)	Visual	0.983	0.967	0.068
Kimi-K3 (max)	Textual	0.982	0.972	0.135
Kimi-K3 (max)	Visual	0.985	0.976	0.185
Claude-Fable-5 (high)	Textual	0.986	0.979	0.146
Claude-Fable-5 (high)	Visual	0.986	0.979	0.205
GPT-5.6-Sol (xhigh)	Textual	0.985	0.977	0.069
GPT-5.6-Sol (xhigh)	Visual	0.985	0.974	0.083
GPT-5.6-Sol (max)	Textual	0.984	0.978	0.134
GPT-5.6-Sol (max)	Visual	0.985	0.974	0.166

C MULTI-AGENT MODELING SKILL

The `partground-multi-agent` skill specifies a coordinator, part builders, and a separate visual reviewer for reference-based Blender modeling. It adapts the planning and separate-review ideas of Procedura (Lin et al., 2026) to PartGround's fixed part specification and restricted feedback protocol. The package contains role instructions, handoff formats, and a plan validator; it relies on the host agent harness for actual delegation and tool execution. Figure 3 summarizes the workflow, and Table 3 reports its evaluation across three model configurations.

C.1 INPUTS, ROLES, AND SHARED PLAN

Every role uses only the task-authorized overall views, part names, and textual or visual guidance. Reference scenes, hidden geometry, and evaluator outputs are excluded from generation and review. The supplied semantic grouping is preserved: a named part may contain repeated or disconnected geometry, and builders may neither split its scored identity nor merge distinct required names. For each experimental configuration, all roles use the same model and reasoning setting; any heterogeneous team or fallback is a separate condition.

The coordinator records the object frame, shared dimensions, estimated part bounds, shape features, material intent, and attachment or clearance relationships in a versioned plan. These estimates come from the visible references rather than ground-truth measurements. Each part has exactly one builder owner, and construction dependencies determine which assignments can proceed independently. Tightly coupled or repeated geometry is grouped into a coherent assignment. The supplied validator checks task identity, exact name coverage, ownership, dependency cycles, and declared worker/review/correction caps against a separately transcribed task contract. It does not verify geometry or visual accuracy.

C.2 CONSTRUCTION AND INTEGRATION

Builders receive the shared plan, allowed overall views, and their relevant part evidence, and write only to assigned artifact directories. Each returns local geometry constructors, node materials, interface locations, and the plan revision it consumed. The coordinator alone writes the assembled program and places each returned part through the shared frame and anchor parameters. An interface change requires a coordinated plan revision; stale handoffs are rejected rather than silently combined. The final program includes its generated helper code when external dependencies are prohibited and saves the task-required scene from a clean Blender state.

Permitted non-render checks cover exact mesh names, nonempty geometry, finite coordinates, material bindings, estimated interface consistency, and agreement between the current script and saved scene. These checks cannot establish reference fidelity. Openings, intentional gaps, articulated clearances, and multi-piece semantic parts remain permissible; the skill does not require a watertight or single connected solid. Suspected penetration is assessed against the visible structure, not inferred solely from overlapping bounding boxes.

C.3 INDEPENDENT REVIEW AND TARGETED CORRECTION

After integration, the coordinator renders the complete object within the task's allowance and sends the preview to a fresh image-capable reviewer context. The reviewer receives the authorized references, part identities, and necessary plan/code excerpts, but not the builders' self-assessments. It returns an ordered issue list covering overall shape, in-situ part geometry, placement and contact, and per-part appearance. Each issue identifies the affected parts, visible evidence, uncertainty, and a bounded correction; it is not a predicted benchmark score.

The coordinator routes the most evident permitted correction to its owner or updates the relevant assembly parameters. The preview and critique are tied to a code revision, with unrelated edits frozen until the selected correction is applied; a later edit invalidates that critique for further changes. Under the current protocol, the entire team shares one visual self-check and one correction, so workers cannot introduce private previews or additional review cycles. After the correction, the object is rebuilt and checked without a second render. All roles share the two-hour task deadline, with no explicit task-level token cap.

C.4 Evaluation and Resource Accounting

The comparison uses Kimi-K3 (max), Claude-Fable-5 (high), and GPT-5.6-Sol (max), each with and without the skill under Visual Part Guidance on the 185-object agent subset. Table 3 reuses the corresponding baseline quality scores from Table 1. Both conditions use the same task inputs, model and reasoning settings, and evaluation protocol; the skill adds coordinated part construction and independent review. Reconstruction metrics use the valid-score aggregation defined in Section 4.1. The held-out evaluator runs only after generation; its scores are not available to any modeling role.

Wall time spans task initialization through final artifact completion, including integration and retries; parallel worker durations are not summed as elapsed time. Total tokens sum unique calls across the coordinator, builders, and reviewer, preserving provider usage fields without counting cached tokens twice. Unavailable usage is recorded as missing rather than zero.