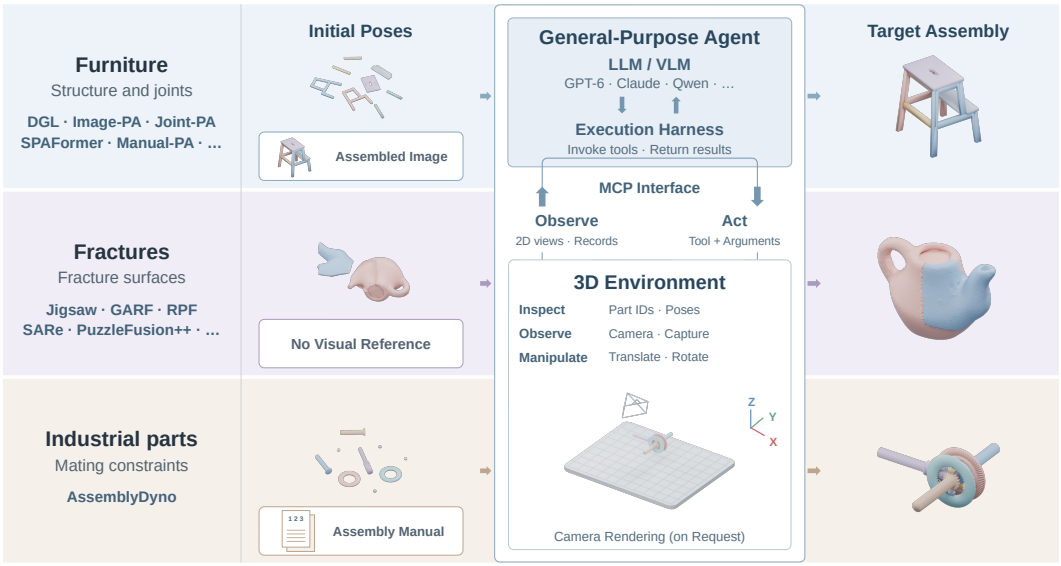
ASSEMBLYWORLD: RETHINKING 3D ASSEMBLY WITH GENERAL-PURPOSE AGENTS

Anonymous authors
Paper under double-blind review



(a) Specialized Methods (b) AssemblyWorld (Ours)

Figure 1: (a) Prior specialized assembly methods across three domains. (b) AssemblyWorld connects a general-purpose model and execution harness to a 3D environment through Model Context Protocol (MCP). Tools return requested views, scene records, or execution status. References guide the agent; initial parts populate the scene. Targets illustrate assembly goals, not agent outputs. Reference conditions, layouts, colors, and the manual icon are illustrative.

ABSTRACT

The task of 3D assembly requires translating an understanding of parts and their relationships into precise spatial arrangements. Can pretrained general-purpose agents assemble objects through visual interaction without additional assembly-specific fine-tuning? To investigate this question, we introduce **AssemblyWorld**, an interactive 3D environment in which agents inspect rendered views and manipulate supplied rigid parts, guided by images or assembly manuals when available. Agents perceive part geometry through 2D views rather than direct access to mesh vertices or faces, while their resulting assemblies are evaluated geometrically. Building on this environment, we construct **AssemblyWorldBench**, comprising 100 assembly tasks across 80 objects spanning furniture, industrial assembly, and fracture reassembly. Evaluating eight agent systems reveals substantial differences in their capabilities. The strongest system achieves 80.9% part accuracy but 59.4% complete-assembly success. The evaluated open-source systems lag substantially behind their stronger closed-source peers in both execution reliability and assembly accuracy. Analyses of visual references, interaction trajectories, and failures show how agents revise assemblies while leaving residual positioning errors. AssemblyWorld provides a common setting for both assessing the capabilities of interactive assembly agents and characterizing the gap between approximate structure recovery and precise reconstruction.

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

Assembling an object requires more than recognizing its parts or understanding what the finished object should look like. A chair may have all its legs beneath the seat and still be incorrectly assembled: connections must align, parts must be correctly oriented, and even small positional errors can leave the object assembly incomplete. These demands arise across domains: furniture assembly (Zhang et al., 2025a), industrial components (Li et al., 2026), and fracture reconstruction (Li et al., 2025), although the available clues range from well-illustrated manuals to just the geometry of the parts alone. The task of *3D assembly* therefore provides a concrete test of whether an intelligent system can translate visual and spatial reasoning into precise transformations in a 3D world.

A substantial body of work studies geometric representations (Huang et al., 2020; Harish et al., 2022; Du et al., 2024) and learned pose prediction (Zhang et al., 2022; Cheng et al., 2023; Xu et al., 2025; Zhao et al., 2025; Zhang et al., 2025b) for 3D part assembly. Reference-guided methods further use assembled-object images or instruction diagrams to inform part placement (Li et al., 2020; Wang et al., 2025a; Zhang et al., 2025a). For example, Zhang et al. (2025a) exploit instruction diagrams in Manual-PA, while Li et al. (2025) model geometric relationships between fragments in GARF. General-purpose agents built on models from the OpenAI GPT and Qwen families (OpenAI, 2023; Yang et al., 2025) offer a complementary approach. Through tool calls, models specify operations for an execution harness to perform and receive their outputs as context for subsequent decisions. In a 3D environment, this allows an agent to inspect a scene, manipulate parts, and use further observations to revise its decisions. This setting requires choosing informative viewpoints, relating observations across views, and translating an assembly hypothesis into numerical pose edits. Because edits change subsequent observations, the agent must also detect and correct its own errors throughout interaction. Success must therefore be assessed from the assembled geometry, since a plausible verbal explanation need not imply accurate part placement. This raises a central question: *can pretrained general-purpose agents assemble objects through visual interaction without additional assembly-specific fine-tuning?* Answering it requires an environment that connects observations to executable geometric actions and supports comparisons across assembly domains, reference conditions, and agent systems.

Towards this end, we introduce *AssemblyWorld*, an interactive 3D environment for assembling supplied rigid object parts. Within this environment, agents inspect rendered 2D views, choose camera viewpoints, and translate or rotate individual object parts and groups. When available, images of the completed object or assembly manuals provide visual guidance. Since object parts are rigid, agents must solve the task by arranging the given object components. As illustrated in Figure 1, the same observation-action interface supports furniture, industrial assembly, and fracture reassembly. Geometric evaluation measures accuracy of the resulting configurations, while recorded states make it possible to examine how an assembly develops throughout interaction.

Building on this environment, we construct *AssemblyWorldBench*, a benchmark comprising 80 objects selected from PartNet (Mo et al., 2019), IKEA-Manual (Wang et al., 2022), AssemblyBench (Li et al., 2026), and Fantastic Breaks (Lamb et al., 2023). Each assembly task consists of an object's parts in a fixed initial configuration and the reference information provided to the agent. We evaluate each of the 20 PartNet objects both with and without an assembled-object image, and each of the remaining 60 objects under its dataset-specific reference setting, yielding 100 assembly tasks across 80 distinct objects. Standardized initial configurations and a common execution budget enable comparisons across eight agent systems. Larger source-level evaluations complement this benchmark by placing agent performance alongside specialized assembly methods. These settings connect cross-domain evaluation with analyses of reference use, interaction dynamics, and residual errors.

Our results show strong assembly capability but a gap between partial progress and complete reconstruction. The strongest system achieves 80.9% part accuracy but only 59.4% complete-assembly success on the common benchmark, while the evaluated open-source systems lag substantially behind their stronger closed-source peers in execution reliability and assembly accuracy. Visual references improve performance for the stronger systems, while trajectories reveal how agents alternate inspection and manipulation, revise intermediate arrangements, and sometimes terminate with unresolved positional errors. Additional experiments examine sensitivity to task variations and the potential for combining interactive agents with specialized geometric models. Together, these findings identify the precision and reliability challenges that remain for general-purpose assembly agents.

2 Related Work

Learning-Based and Instruction-Guided Assembly. Learning-based 3D assembly models relative part poses using geometric and structural cues, including graph- and hierarchy-based reasoning (Huang et al., 2020; Du et al., 2024), generative pose modeling (Cheng et al., 2023), and visual or instructional guidance (Li et al., 2020; Zhang et al., 2025a). Recent language-based methods further incorporate geometric encoding, manual understanding, and constraint reasoning (Jing et al., 2026; Tie et al., 2025; 2026). Fracture reassembly instead relies more heavily on geometric correspondence between fragments (Lu et al., 2023; Li et al., 2025; Sun et al., 2025b). In contrast, we study whether general-purpose agents can perform these assembly tasks through rendered observations and pose-editing tools without additional assembly-specific fine-tuning.

3D Assembly Datasets and Benchmarks. Existing datasets capture complementary aspects of the task of 3D assembly. PartNet provides hierarchically annotated object parts (Mo et al., 2019), IKEA-Manual pairs furniture geometry with real assembly instructions (Wang et al., 2022), and AssemblyBench extends instruction-guided assembly to industrial objects (Li et al., 2026). For fracture reconstruction, Breaking Bad provides simulated fractures (Sellán et al., 2022), while Fantastic Breaks provides scans of physically broken objects (Lamb et al., 2023). AssemblyWorldBench unifies subsets of these datasets in a common interactive environment with standardized initialization and geometric evaluation, enabling cross-domain agent comparison while retaining the reference conditions used by each dataset.

Interactive Agents in 3D Environments. Tool-using agents connect language and visual reasoning to actions in executable environments. Liang et al. (2023) generate action programs in Code as Policies, while Huang et al. (2023) and Huang et al. (2025) use spatial objectives and constraints in VoxPoser and ReKep. Related work in 3D content creation includes procedural modeling in 3D-GPT (Sun et al., 2025a) and iterative scene reconstruction through rendered feedback in Thinking in Blender (He et al., 2026) and VIGA (Yin et al., 2026). In contrast, AssemblyWorld studies interactive assembly of a fixed set of parts through rendered observations and pose-editing tools, with execution-based evaluation following the paradigm of OSWorld (Xie et al., 2024).

3 AssemblyWorld

3.1 Task Formulation

We study the task of interactive 3D assembly of a fixed set of rigid parts. The task setup provides part meshes $\mathcal{M} = \{M_i\}_{i=1}^{n}$ in a separated initial configuration, together with optional reference information r, such as an image of the completed object or an assembly manual. The agent accesses the parts through a 3D environment and uses observations and manipulation actions to determine their final poses $\widehat{T}_i \in \mathrm{SE}(3)$. Part geometry and relative scale remain fixed throughout interaction. The objective is to assemble the supplied parts into a coherent object, following the reference image or instructions when provided.

3.2 Interactive 3D Environment

General-purpose agents interact with external environments through predefined tools. The model selects a tool and generates its arguments; the execution harness invokes the operation and returns its output to the model. In AssemblyWorld (Figure 1), these tools are exposed through Model Context Protocol (MCP): an agent can request a rendered view, choose a part translation or rotation, and then request another view to inspect the change. The returned information becomes part of the context for subsequent decisions.

Observation. The agent observes the current assembly through rendered 2D views and can adjust the camera to inspect connections from different viewpoints. Scene-query tools supplement these observations with structured records: `list_objects` returns part identifiers, while `get_object` reports a selected part's body position, orientation, and mesh-local bounding dimensions. Mesh vertices and faces are hidden; part geometry must be inferred from rendered views.

Manipulation and Interaction. Tool descriptions specify the available operations and their arguments. To translate or rotate parts, the agent selects their identifiers and specifies a translation vector

or rotation angles. It can also choose the coordinate frame and, for rotations, a pivot. Group operations allow several parts to be repositioned together while preserving their relative configuration. The environment executes the requested transformation and returns its status; the agent can then request a rendered view to inspect the result.

The agent determines the sequence of observations and edits and when to finish. We record the operations and resulting states, retaining the final configuration when execution ends.

3.3 Benchmark Construction

3D Assembly Datasets. AssemblyWorldBench covers three assembly domains using four datasets. PartNet (Mo et al., 2019) and IKEA-Manual (Wang et al., 2022) provide furniture with semantic parts, such as seats, legs, and tabletops, whose functional roles and spatial relationships help determine the object structure. AssemblyBench (Li et al., 2026) provides complex industrial assemblies, where components must be arranged according to their geometric fit and mating relationships. Fantastic Breaks (Lamb et al., 2023) contains fractured objects whose fragments provide complementary surface geometry, rather than the semantic part structure (as in furniture assembly).

Reference Conditions. IKEA-Manual provides real-world IKEA assembly instructions, whereas AssemblyBench provides sequences of rendered assembly diagrams shown from a consistent viewpoint. Fantastic Breaks tasks provide no reference. For PartNet, we evaluate each object both without a reference and with an image of the fully assembled furniture, using the same geometry and initial configuration in both conditions. The paired conditions allow us to compare performance with and without a reference image on the same objects. With only a fixed reference image, the agent must infer occluded part placements from the supplied parts and visible structure. We evaluate the final assembly geometry rather than agreement with an illustrated operation sequence.

Standardized Initial Configurations. We apply the same initialization procedure across datasets. All parts of an object use a shared scale to preserve their relative dimensions. Each part is placed in a local coordinate frame defined by its principal axes, assigned a randomized yaw, and scattered into a separated initial layout. The normalization scale depends only on individual part geometry. We fix the initial configuration of each object across all evaluated systems.

Benchmark Composition. We select 20 objects from each dataset through random sampling with part-count stratification where part counts vary. The resulting benchmark provides a common set of assembly tasks for comparing general-purpose agents across domains and reference conditions. The Fantastic Breaks subset contains only two-part objects. Evaluating each PartNet object both with and without a reference image yields 100 assembly tasks across 80 distinct objects. Appendix A provides task construction and execution details.

4 Experiments

4.1 Experimental Setup

Evaluation Settings. We compare eight agents on the 100 assembly tasks of AssemblyWorldBench, constructed from the four datasets described in Section 3.3. We also evaluate GPT-6 Astra on larger subsets of these datasets for comparison with specialized methods.

Agents and Execution Protocol. Table 1 lists the eight agents and their resource use. GPT, Qwen, and DeepSeek use Codex, with provider adapters for the latter two; Claude models use Claude Code. All systems share the environment tools, initial configurations, task instructions, and 60-minute execution limit. We score the final exported configuration, including partial assemblies after errors or timeouts.

Evaluation Metrics. For the common benchmark, predicted assemblies are globally aligned to the targets before evaluation, and geometrically equivalent parts are matched using Hungarian assignment. We report shape Chamfer distance (SCD), part accuracy (PA), and shape success rate (SR). SCD measures the squared bidirectional Chamfer distance between the complete predicted and target assemblies. PA is the fraction of correctly placed parts, where a part is considered correct if its squared bidirectional Chamfer distance to the matched target part is at most 0.01. SR requires all parts of an object to be correct. In our free-space setting, unfinished assemblies may leave parts

Table 1: AssemblyWorldBench results: SR (%) and mean resource use for eight agents. NR, IR, and M denote no reference, an assembled-object image, and a manual. IKEA, AB, and FB denote IKEA-Manual, AssemblyBench, and Fantastic Breaks. Bold marks the highest SR or lowest resource use.

LLM	Harness	PartNet		IKEA	AB	FB	Overall	Time	MCP	Tokens	Cost
		NR	IR	M	M	NR		(min)	calls	(M)	($)
DeepSeek V4.1 Flash	Codex	0.0	0.0	0.0	0.0	0.0	0.0	26.7	186	2.59	**0.16**
Qwen3.8 Max	Codex	5.0	0.0	0.0	10.0	35.0	11.9	54.5	107	3.67	1.67
Claude Sonnet 5	Claude Code	0.0	0.0	0.0	5.0	25.0	7.5	20.6	196	16.45	4.72
Claude Opus 5	Claude Code	10.0	25.0	50.0	40.0	70.0	44.4	37.0	180	16.89	12.38
Claude Fable 5.1	Claude Code	5.0	25.0	50.0	45.0	**90.0**	50.0	22.1	162	4.60	6.84
GPT-5.6 Terra	Codex	0.0	0.0	0.0	0.0	30.0	7.5	**4.0**	**60**	1.32	0.50
GPT-5.6 Sol	Codex	0.0	0.0	5.0	10.0	30.0	11.2	8.7	208	2.77	1.79
GPT-6 Astra	Codex	**15.0**	**40.0**	**65.0**	**55.0**	**90.0**	**59.4**	5.8	94	**1.08**	2.02

far from their targets, so a small number of severe failures can dominate the mean SCD. PA and SR are reported as percentages and SCD as $\times 1000$. Overall PA and SR average the four datasets equally, with the two PartNet reference conditions averaged first. The source-level Fantastic Breaks comparison uses a separate anchor-aligned protocol. Appendices A–B provide evaluation details and supplementary quantitative results.

4.2 Results on AssemblyWorldBench

Assembly Quality. Table 1 compares assembly quality and resource use across the eight systems. Astra achieves the highest overall SR at 59.4%, followed by Fable at 50.0% and Opus at 44.4%. Performance nevertheless varies across domains: Astra and Fable both reach 90% SR on two-part fracture reassembly, but only 40% and 25%, respectively, on image-conditioned PartNet tasks. High success on fragment alignment therefore does not extend uniformly to multi-part furniture assembly.

Qwen reaches 35% SR on two-part fracture reassembly, but its overall SR is 11.9%; DeepSeek completes no task successfully. Of their 100 evaluations, 65 and 17 reach the time limit, while another 14 and 34 end in execution failures. These results reflect the deployed model–harness combinations, including execution reliability as well as assembly accuracy.

Time and Computational Cost. Higher success does not necessarily require more time or tool calls. Astra averages 5.8 minutes and 94 calls per evaluation, versus Sol's 8.7 minutes and 208 calls, while achieving substantially higher SR. Astra also uses less time and recorded cost than Fable and Opus, whereas Terra uses fewer resources but reaches only 7.5% SR.

4.3 Comparison with Specialized Assembly Methods

Tables 2–5 compare Astra with specialized assembly methods on larger source-level evaluation sets, using the reported settings of each method.

Furniture Assembly. PartNet methods commonly train category-specific models, including Manual-PA, Imagine (Wang et al., 2025a), and Assembler (Zhao et al., 2025). Cross-category transfer is studied by Manual-PA and Imagine, and multi-category training by SPAFormer (Xu et al., 2025) and Assembler. We use the same pretrained Astra agent across furniture categories without additional assembly-specific fine-tuning or category-specific models.

With image guidance, Astra exceeds the compared methods in PA and SR across all three PartNet categories (Table 2), reaching 85.35% PA and 63.60% SR on tables. Removing the reference reduces PA in every category. Image-conditioned Storage reaches 63.87% PA but only 20.27% SR, illustrating the gap between correctly placing individual parts and completing an assembly. Free-space failures can nevertheless dominate SCD: 8 of 148 image-conditioned Storage objects (5.4%) contribute 99.3% of summed SCD, with six retaining all initial part poses. Their separated parts yield large squared distances, unlike predictors with bounded translations such as DGL (Huang et al., 2020).

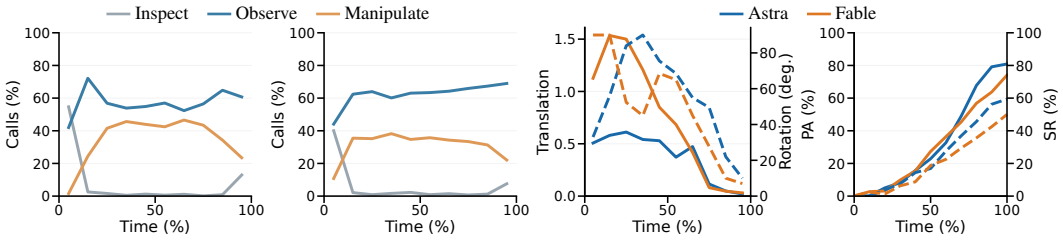
Real manuals add challenges beyond rendered diagrams: Manual-PA uses synthetic stepwise diagrams for PartNet and IKEA-Manual and identifies arrows, close-ups, part labels, and hierarchical

Table 2: Assembly results on PartNet. Astra uses no target reference (–) or an image of the assembled furniture; scene observation is available in both conditions.

Method	Condition	SCD ↓			PA ↑			SR ↑		
		Chair	Table	Storage	Chair	Table	Storage	Chair	Table	Storage
$\text{DGL}_{\text{NeurIPS'20}}$	–	9.1	5.0	–	39.00	49.51	–	–	–	–
$\text{RGL}_{\text{WACV'22}}$	Sequence	8.7	4.8	–	49.06	54.16	–	–	–	–
$\text{IET}_{\text{RA-L'22}}$	–	5.4	3.5	–	62.80	61.67	–	–	–	–
$\text{Score-PA}_{\text{BMVC'23}}$	–	7.4	4.5	–	42.11	51.55	–	8.32	11.23	–
$\text{CCS}_{\text{AAAI'24}}$	–	7.0	–	–	53.59	–	–	–	–	–
$\text{3DHPA}_{\text{CVPR'24}}$	–	5.1	**2.8**	–	64.13	64.83	–	–	–	–
$\text{Joint-PA}_{\text{CVPR'24}}$	Joint	6.0	7.0	–	**72.80**	67.40	–	–	–	–
$\text{SPAFormer}_{\text{3DV'25}}$	Sequence	6.7	3.8	**4.5**	55.88	64.38	**56.11**	16.40	33.50	7.85
$\text{CFPA}_{\text{NeurIPS'25}}$	–	**4.9**	3.3	–	69.24	68.48	–	–	–	–
$\text{Assembler}_{\text{SA'25}}$	–	9.2	8.0	–	61.85	65.04	–	22.59	39.07	–
GPT-6 Astra + Codex	–	69.9	47.0	141.6	55.68	**71.39**	50.06	**22.63**	**48.97**	**12.84**
$\text{Image-PA}_{\text{ECCV'20}}$	Image	6.7	3.7	5.0	45.40	71.60	40.20	–	–	–
$\text{Manual-PA}_{\text{ICCV'25}}$	Image	5.9	3.9	3.7	62.67	70.10	47.79	19.97	32.83	4.730
$\text{Imagine}_{\text{AAAI'25}}$	Image	**4.2**	**2.5**	**2.4**	65.88	65.13	56.35	–	–	–
$\text{Assembler}_{\text{SA'25}}$	Image	7.2	4.7	–	66.17	70.44	–	24.92	43.46	–
GPT-6 Astra + Codex	Image	29.5	21.0	184.8	**76.45**	**85.35**	**63.87**	**43.36**	**63.60**	**20.27**

Table 3: Assembly results on IKEA-Manual. Subset columns use equal-category averages. PA and SR are percentages; SR requires all parts correct, whereas SR^a uses SCD < 0.02 before scaling. † denotes Image-PA retrained on diagrams.

Method	Chair/Table			Bench/Chair/Desk		Full	
	SCD ↓	PA ↑	SR ↑	SCD ↓	SR^a ↑	SCD ↓	SR^a ↑
$\text{3DHPA}_{\text{CVPR'24}}$	36.1	2.971	0.000	–	–	–	–
$\text{Image-PA}_{\text{ECCV'20}}$ †	16.0	27.91	5.265	–	–	–	–
$\text{Manual-PA}_{\text{ICCV'25}}$	8.1	46.12	9.650	–	–	21.6	54.90
$\text{TwoByTwo}_{\text{CVPR'25}}$	–	–	–	221.7	6.50	–	–
$\text{AssemLM}_{\text{arXiv'26}}$	–	–	–	21.3	81.00	61.3	69.60
GPT-6 Astra + Codex	**0.8**	**93.27**	**75.44**	**1.0**	**100.00**	**0.9**	**100.00**



(a) Astra activity (b) Fable activity (c) Manipulation (d) Assembly quality

Figure 2: Interaction dynamics of Astra and Fable across 100 evaluations each. (a–b) Mean call shares within active evaluation-bins. (c) Median per-evaluation translation and rotation magnitudes; translation is normalized by the largest-part diagonal. (d) Mean PA and SR. All four plots use benchmark source weights and normalized interaction time. In (c–d), colors identify systems; solid and dashed lines use the left and right axes, respectively.

subassemblies as obstacles to transfer. Manual2Skill handles real manuals with graph extraction and learned pose estimation. Astra instead uses the original IKEA pages to assemble through interaction, without additional assembly-specific fine-tuning or a dedicated pose predictor.

On IKEA-Manual (Table 3), Astra achieves category-averaged PA/SR of 93.27%/75.44% on Chair/Table, exceeding the compared methods. It also obtains lower SCD and higher SR^a on Full and Bench/Chair/Desk, with full-dataset SCD of 0.9 and 100% SR^a in both settings. These results show that a pretrained general-purpose agent can interpret real instruction manuals and construct furniture geometry through interaction, without an assembly-specific pose predictor.

Industrial Assembly. Industrial assembly introduces substantial differences in relative part size within a single object. The median largest-to-smallest part-size ratio is 4.39 on AssemblyBench, compared with 2.20 on IKEA-Manual and 1.79 on PartNet. The ratio of per-part PCA (principal component analysis) bounding-box diagonals is scale-invariant. Using stepwise rendered instructions, the same pretrained Astra agent achieves 78.32% PA and 55.91% SR across 279 AssemblyBench objects without additional assembly-specific fine-tuning, exceeding the reported results of the specialized methods in Table 4.

Table 4: Assembly results on AssemblyBench. N counts evaluated objects; PA/SR are percentages and SCD is scaled by 1000. Astra is evaluated on 279 of 280 objects, excluding one refusal involving a dangerous item.

Method	N	SCD ↓	PA ↑	SR ↑
Manual-PA$_{\text{ICCV'25}}$	280	4.24	70.04	33.57
AssemblyDyno$_{\text{CVPR'26}}$	280	**3.91**	71.21	34.64
Astra	279	8.54	**78.32**	**55.91**

Fracture Reassembly. In our fracture reassembly setting, agents receive no visual reference of the target object, such as an image of its intact shape. Within AssemblyWorld's interactive 3D environment, the agent cannot directly access mesh vertices, faces, or point clouds; it must infer how fragments fit together and adjust their poses using feedback from rendered 2D views. Despite this observation constraint, Astra achieves 91.67% PA on 150 Fantastic Breaks objects without additional fracture-specific fine-tuning. Table 5 compares its performance with specialized methods using published results reported by Li et al. (2025) for GARF and Jia et al. (2026) for SARe. The higher PA and lower translation errors reported by recent specialized methods highlight their strength in precise fragment placement.

Table 5: Assembly results on Fantastic Breaks. RE/TE are rotation/translation errors. Units: RE in degrees, PA in percent, TE/CD scaled by 100/1000. RE is Euler-angle root mean square error (RMSE) for Jigsaw, PF++, GARF, and Astra; geodesic otherwise. Astra uses anchor alignment on 150 objects.

Method	RE ↓	TE ↓	PA ↑	CD ↓
Jigsaw$_{\text{NeurIPS'23}}$	26.30°	6.43	73.64	10.47
PF++$_{\text{ICLR'25}}$	20.68°	4.37	83.33	6.68
GARF$_{\text{ICCV'25}}$	10.62°	2.10	91.00	2.12
RPF$_{\text{NeurIPS'25}}$	6.32°	2.18	96.90	2.53
TORA-CKA$_{\text{ECCV'26}}$	**3.03°**	0.80	97.28	0.26
SARe-Gen$_{\text{arXiv'26}}$	7.78°	**0.35**	**98.53**	**0.23**
Astra	11.45°	2.54	91.67	6.98

4.4 Analysis of Interactive Assembly

Interaction Dynamics. Agents actively inspect intermediate assemblies and adjust part poses using visual feedback. Figure 3 shows Astra changing viewpoint to examine the bench end frames and correcting their orientation before turning the assembly upright. In the industrial example, it temporarily lifts the perforated plate and retaining ring to inspect and adjust the blade beneath them, then restores the lifted parts. Such inspection can require moving parts away from their target poses, so intermediate geometric accuracy alone does not determine whether an action is useful.

Figure 2 summarizes how these interactions evolve across Astra and Fable trajectories. In Figure 2(a–b), inspection queries are concentrated at the beginning, when agents acquire scene and part information. Manipulation then becomes more frequent alongside sustained visual observation. Toward the end, manipulation decreases while observation remains prominent, consistent with a shift toward checking the resulting configuration. Figure 2(c) shows an accompanying reduction in translation and rotation magnitudes, while Figure 2(d) shows overall increases in mean PA and SR. Manipulation magnitudes also decrease in failed evaluations, so smaller edits alone do not establish convergence to a correct assembly.

Effect of Visual References. Reference images improve assembly accuracy, but their benefit varies across systems. Figure 4 compares paired evaluations with and without an image of the assembled furniture. On the benchmark's 20 PartNet objects, Astra, Fable, and Opus gain 25, 20, and 15 SR points, respectively; the other systems show no gain or a decrease. The larger PartNet evaluations in Table 2 show a similar benefit for Astra. Across 533 tables and 148 storage objects, SR increases by 14.6 and 7.4 points, respectively. These gains extend beyond the benchmark subset and differ between furniture categories.

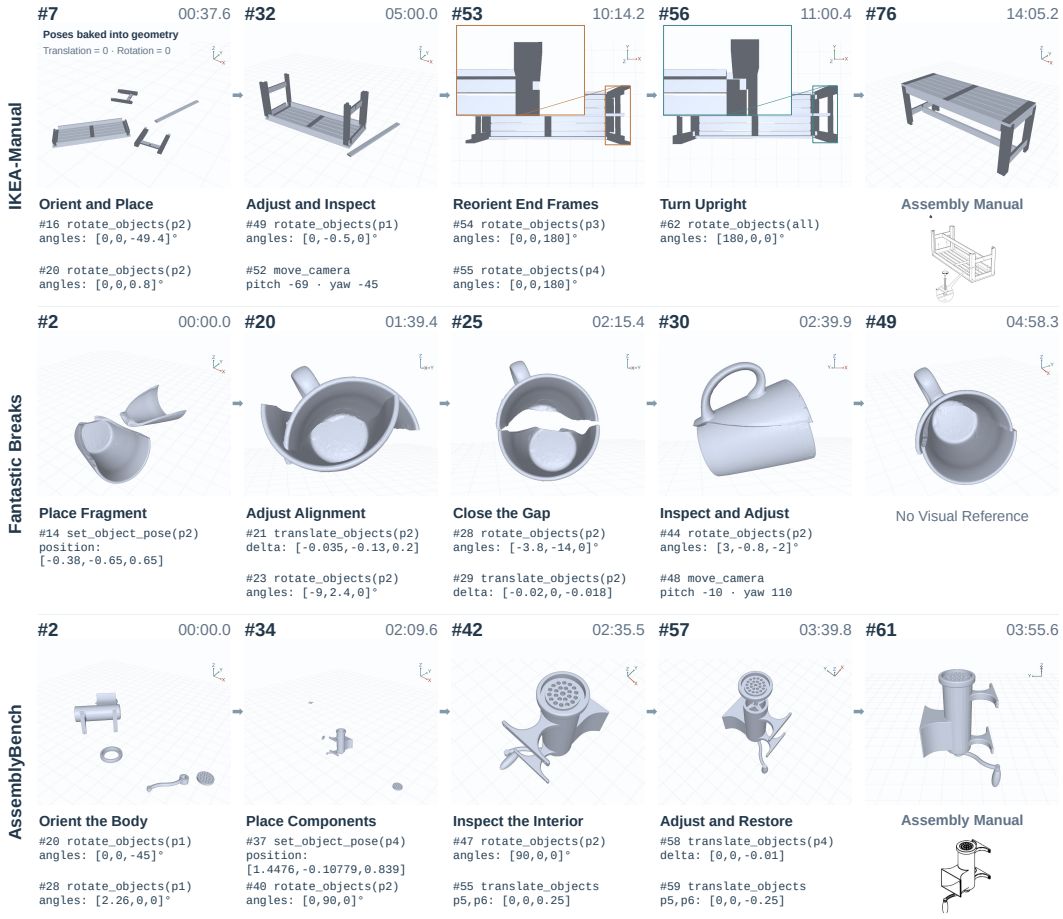


Figure 3: GPT-6 Astra interaction trajectories across three assembly domains. Recorded states are re-rendered from the corresponding camera views. Labels show environment-call numbers (#) and elapsed time; selected MCP calls connect successive views. Insets highlight the IKEA end-frame correction. Part IDs and arguments are abbreviated; angles are in degrees.

Failure Modes. Failures span execution and geometry: Qwen and DeepSeek records contain unsupported-tool and stream-disconnection errors, while all 49 normally terminated DeepSeek runs fail geometrically. Figure 5(a–b) shows Terra grouping 13 dispersed parts and Qwen reporting numerical alignment despite detached components. These cases reveal difficulties in assessing visible geometry from pose readback.

Some Astra and Fable failures retain much of the target structure: translation-dominated errors account for 67.4% and 75.1% of their incorrect parts. These parts pass the correctness threshold after centroid alignment without changing orientation. Figure 5(c) shows an Astra industrial assembly with eight of nine parts correct and one positional error; a Fable case has six of seven correct. Their reports describe unresolved seating or internal fits inferred from rendered views.

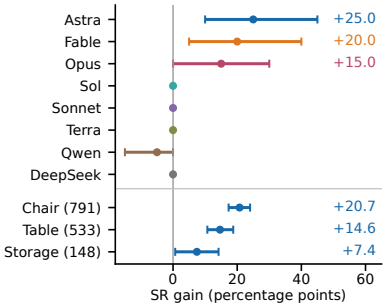
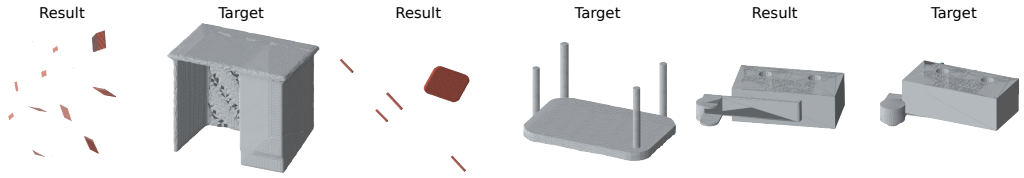


Figure 4: SR gains from reference images, with paired 95% intervals: benchmark (top) and Astra on larger PartNet sets (bottom). Labels give gains in percentage points.

These residual errors motivate combining visual interaction with geometric refinement, while checking whether either composition degrades correct structure. Appendix C extends the interaction and failure analyses.

Combining Agents with Geometric Models. Table 6 compares both composition orders of Astra and GARF on 150 two-fragment Fantastic Breaks objects with shared point samples and metrics.



(a) Terra (b) Qwen (c) Astra

Figure 5: Recorded failure cases and targets. (a) Logical grouping leaves parts dispersed. (b) Numerical pose placement leaves components detached. (c) Eight of nine parts are correct, with one residual positional error. Red marks incorrect parts; teal marks the matched target in (c). Views in (a–b) are fitted independently.

Agent initialization raises GARF PA from 83.67% to 87.00%, below the agent-only 91.67%: refinement reduces rotation error but increases translation error, degrading some correct assemblies. GARF was not trained to refine agent outputs and is applied without retraining.

Conversely, Astra inspects GARF predictions through rendered views and adjusts their part poses, reaching 93.00% PA and reducing CD from 8.82×10^{-3} to 4.35×10^{-3}. All four mean metrics improve over the agent-only baseline. These results support visual correction of geometric predictions and motivate refinement adapted to agent-generated initializations. Appendix D reports the matched protocol and paired changes.

Table 6: Bidirectional refinement on 150 Fantastic Breaks objects with shared point samples and anchor-aligned evaluation. Arrows indicate execution order. RE is in degrees, PA in percent, and TE/CD scaled by 100/1000.

Method	RE ↓	TE ↓	PA ↑	CD ↓
Agent	11.45	2.54	91.67	6.98
GARF	14.78	5.16	83.67	8.82
Agent → GARF	**9.34**	4.24	87.00	6.94
GARF → Agent	10.35	**1.84**	**93.00**	**4.35**

Robustness to Task Variations. We evaluate Astra's sensitivity to initial layouts and part-set changes on five fixed objects per benchmark block, with paired PartNet reference conditions and one run per condition. Aggregate SR changes little across layouts, from 52.5% to 52.5% and 50.0%, but some objects switch between success and failure. Similar aggregate performance does not imply consistent per-object success.

With one part removed, retained-part PA differs substantially across domains: 88.4% on IKEA-Manual versus 43.3% on AssemblyBench. With an added distractor, Astra sometimes leaves it unincorporated and sometimes attempts to assign it a role. Because the unchanged prompt requests that all supplied parts be incorporated, these behaviors cannot be interpreted directly as anomaly-detection ability. Appendix E provides the complete paired results and trajectory audits. These small-sample, single-run findings remain exploratory.

5 Conclusion

We introduced AssemblyWorld, an interactive 3D assembly environment, and AssemblyWorld-Bench, a cross-domain benchmark evaluating general-purpose agents with a common interface and geometric criteria. Across eight systems, Astra achieves the highest overall success rate, while the stronger closed-source systems substantially outperform the evaluated open-source systems. The same pretrained agent can assemble furniture, industrial objects, and fractured objects without additional assembly-specific fine-tuning. The gap between part accuracy and complete-assembly success nevertheless shows that broad capability does not ensure geometric precision.

These findings motivate a division of labor in which general-purpose models provide cross-domain reasoning, while specialized geometric methods provide precise pose estimation and refinement. The dependence on composition order in our refinement study highlights the need to design this interaction jointly. Our evaluation concerns free-space geometry and does not establish collision-free or physically stable assembly. The sampled tasks, single-run robustness tests, and two-fragment, single-seed refinement study limit the scope of our findings; unknown pretraining exposure also prevents claims of contamination-free generalization. Within these bounds, AssemblyWorld provides a common platform for investigating how general-purpose reasoning and specialized geometry can jointly improve assembly across domains.

AI USE STATEMENT

We used Claude and OpenAI Codex to assist with research methodology, implementation and experimental workflows, analysis and interpretation of results, and manuscript and figure preparation. The authors reviewed the AI-assisted text, code, references, and experimental results and take full responsibility for the final content.

ETHICS STATEMENT

This work uses publicly available datasets and involves no human-subject studies or non-public personal data. We have reviewed the source datasets' licenses and will distribute data and derived artifacts in accordance with their terms.

REPRODUCIBILITY STATEMENT

The appendix documents the evaluation protocols, task instructions, and supplementary analyses. Following submission, we will release the code, including dataset adapters and evaluation tools, together with data permitted for redistribution and recorded experimental outputs. The authors reviewed the experimental pipeline, including checks that ground-truth assembly poses and evaluator-only information were not exposed to the evaluated agents.

REFERENCES

Junfeng Cheng, Mingdong Wu, Ruiyuan Zhang, Guanqi Zhan, Chao Wu, and Hao Dong. Score-PA: Score-based 3D part assembly. In *Proceedings of the British Machine Vision Conference*, 2023.

Bi'an Du, Xiang Gao, Wei Hu, and Renjie Liao. Generative 3D part assembly via part-whole-hierarchy message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Abhinav Narayan Harish, Rajendra Nagar, and Shanmuganathan Raman. RGL-NET: A recurrent graph learning framework for progressive part assembly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

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.

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

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3D value maps for robotic manipulation with language models. In *Conference on Robot Learning*, 2023.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *Proceedings of The 8th Conference on Robot Learning*, 2025.

Hanze Jia, Chunshi Wang, Yuxiao Yang, Zhonghua Jiang, Yawei Luo, Shuainan Ye, and Tan Tang. SARe: Structure-aware generative 3D fragment reassembly. *arXiv preprint arXiv:2603.21611v2*, 2026.

Zhi Jing, Jinbin Qiao, Ouyang Lu, Jicong Ao, Shuang Qiu, Huazhe Xu, Yu-Gang Jiang, and Chenjia Bai. AssemLM: A spatial reasoning multimodal large language model for robotic assembly. *arXiv preprint arXiv:2604.08983v2*, 2026.

Nikolas Lamb, Cameron Palmer, Benjamin Molloy, Sean Banerjee, and Natasha Kholgade Banerjee. Fantastic breaks: A dataset of paired 3D scans of real-world broken objects and their complete counterparts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Nahyuk Lee, Zhiang Chen, Marc Pollefeys, and Sunghwan Hong. TORA: Topological representation alignment for 3D shape assembly. In *European Conference on Computer Vision*, 2026.

Danrui Li, Jiahao Zhang, Bernhard Egger, Moitreya Chatterjee, Suhas Lohit, Tim K. Marks, and Anoop Cherian. AssemblyBench: Physics-aware assembly of complex industrial objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Sihang Li, Zeyu Jiang, Grace Chen, Chenyang Xu, Siqi Tan, Xue Wang, Irving Fang, Kristof Zyskowski, Shannon P. McPherron, Radu Iovita, Chen Feng, and Jing Zhang. GARF: Learning generalizable 3D reassembly for real-world fractures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3D part assembly from a single image. In *European Conference on Computer Vision*, 2020.

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

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation*, 2023.

Jiaxin Lu, Yifan Sun, and Qixing Huang. Jigsaw: Learning to assemble multiple fractured objects. In *Advances in Neural Information Processing Systems*, 2023.

Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. URL `https://arxiv.org/abs/2303.08774`.

Yu Qi, Yuanchen Ju, Tianming Wei, Chi Chu, Lawson L. S. Wong, and Huazhe Xu. Two by two: Learning multi-task pairwise objects assembly for generalizable robot manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Silvia Sellán, Yun-Chun Chen, Ziyi Wu, Animesh Garg, and Alec Jacobson. Breaking bad: A dataset for geometric fracture and reassembly. In *Advances in Neural Information Processing Systems*, 2022.

Chunyi Sun, Junlin Han, Weijian Deng, Xinlong Wang, Zishan Qin, and Stephen Gould. 3D-GPT: Procedural 3D modeling with large language models. In *International Conference on 3D Vision*, 2025a.

Tao Sun, Liyuan Zhu, Shengyu Huang, Shuran Song, and Iro Armeni. Rectified point flow: Generic point cloud pose estimation. In *Advances in Neural Information Processing Systems*, 2025b.

Chenrui Tie, Shengxiang Sun, Jinxuan Zhu, Yiwei Liu, Jingxiang Guo, Yue Hu, Haonan Chen, Junting Chen, Ruihai Wu, and Lin Shao. Manual2Skill: Learning to read manuals and acquire robotic skills for furniture assembly using vision-language models. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.150.

Chenrui Tie, Shengxiang Sun, Yudi Lin, Yanbo Wang, Zhongrui Li, Zhouhan Zhong, Jinxuan Zhu, Yiman Pang, Haonan Chen, Junting Chen, Ruihai Wu, and Lin Shao. Manual2Skill++: Connector-aware general robotic assembly from instruction manuals via vision-language models. In *IEEE International Conference on Robotics and Automation*, 2026.

Ruocheng Wang, Yunzhi Zhang, Jiayuan Mao, Ran Zhang, Chin-Yi Cheng, and Jiajun Wu. IKEA-Manual: Seeing shape assembly step by step. In *Advances in Neural Information Processing Systems*, 2022.

Weihao Wang, Yu Lan, Mingyu You, and Bin He. Imagine: Image-guided 3D part assembly with structure knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025a.

Zhengqing Wang, Jiacheng Chen, and Yasutaka Furukawa. PuzzleFusion++: Auto-agglomerative 3D fracture assembly by denoise and verify. In *International Conference on Learning Representations*, 2025b.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, 2024.

Boshen Xu, Sipeng Zheng, and Qin Jin. SPAFormer: Sequential 3D part assembly with transformers. In *International Conference on 3D Vision*, 2025.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Xiuyu Li, Michael J. Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. Vision-as-inverse-graphics agent via interleaved multimodal reasoning. *arXiv preprint arXiv:2601.11109*, 2026.

Jiahao Zhang, Anoop Cherian, Cristian Rodriguez, Weijian Deng, and Stephen Gould. Manual-PA: Learning 3D part assembly from instruction diagrams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025a.

Rufeng Zhang, Tao Kong, Weihao Wang, Xuan Han, and Mingyu You. 3D part assembly generation with instance encoded transformer. *IEEE Robotics and Automation Letters*, 7(4):9051–9058, 2022.

Ruiyuan Zhang, Jiaxiang Liu, Zexi Li, Hao Dong, Jie Fu, and Chao Wu. Scalable geometric fracture assembly via co-creation space among assemblers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Xinyi Zhang, Bingyang Wei, Ruixuan Yu, and Jian Sun. Coarse-to-fine 3D part assembly via semantic super-parts and symmetry-aware pose estimation. In *Advances in Neural Information Processing Systems*, 2025b.

Wang Zhao, Yan-Pei Cao, Jiale Xu, Yuejiang Dong, and Ying Shan. Assembler: Scalable 3D part assembly via anchor point diffusion. In *SIGGRAPH Asia 2025 Conference Papers*, 2025.

A Experimental Details

A.1 Datasets and Task Construction

Table A-1 distinguishes the common benchmark from the larger source-level evaluations. The benchmark contains 20 objects per source. PartNet objects have paired image and no-reference conditions, giving 100 tasks across 80 distinct objects. All systems receive the same initial configurations and references.

Table A-1: Dataset coverage in the reported evaluations. Upper rows are the subsets used for all eight systems; lower rows are the larger source-level evaluations. Counts describe evaluation sets, not full datasets; parts are counted once per object, irrespective of reference condition or system. The paired PartNet benchmark conditions give 40 assembly tasks from 20 objects. AssemblyBench retains 279 of 280 objects.

Dataset / evaluation set	Objects	Total parts	Parts/object	Reference
Benchmark subsets				
PartNet	20	218	5–15	None / assembled-furniture image
IKEA-Manual	20	163	3–19	Real IKEA instructions
AssemblyBench	20	139	3–18	Rendered step diagrams
Fantastic Breaks	20	40	2	None
Larger evaluations				
PartNet chair	791	8,596	2–20	None / assembled-furniture image
PartNet table	533	4,732	2–20	None / assembled-furniture image
PartNet storage	148	1,970	5–20	None / assembled-furniture image
IKEA-Manual	102	754	2–19	Real IKEA instructions
AssemblyBench	279	1,821	3–20	Rendered step diagrams
Fantastic Breaks	150	300	2	None

Sampling. We randomly sample objects with seed zero, balancing low, middle, and high part-count bands defined by source-pool tertiles. PartNet uses category-specific bands and contains seven chairs, seven tables, and six storage objects. IKEA-Manual and Fantastic Breaks use category quotas proportional to their candidate pools, with at least one object for categories containing three or more candidates. Fantastic Breaks contains only two-part objects and is not stratified by part count. The resulting low/middle/high counts are 6/7/7 for PartNet, 7/6/7 for IKEA-Manual, and 7/7/6 for AssemblyBench. These subsets support common-task comparisons across systems; they are not assumed to reproduce full-dataset performance.

Two replacements preserve the original category and part-count band: one AssemblyBench object was replaced after content screening, and one IKEA bench was replaced with the first bench in the source dataset. Two additional AssemblyBench objects were excluded from the candidate pool during screening. The larger AssemblyBench evaluation retains 279 of 280 objects after a dangerous-item refusal and uses a successful retry for one other object. Its reported means are conditional on these choices. The original attempt for that retry is unavailable. These source-level results are distinct from the 100-task benchmark aggregate.

The larger PartNet comparisons cover the same 791 chairs, 533 tables, and 148 storage objects with and without an assembled-object image. IKEA-Manual contains 102 objects: 57 chairs, 19 tables, eight benches, four desks, three shelves, and 11 miscellaneous objects. Chair/Table and Bench/Chair/Desk summaries average category means equally. Fantastic Breaks uses 150 two-fragment objects.

Initialization. Each part is expressed in a right-handed vertex-PCA frame, with its smallest principal axis along local Z. A shared scale, twice the largest vertex radius about any part's centroid, preserves relative part sizes without using assembled poses. Parts receive independent yaw angles sampled uniformly from $[-\pi, \pi)$ and are scattered on the XY plane with non-overlapping bounding boxes and a minimum gap of 0.02 normalized units. Sampling and initialization use seed zero. Initial placements are baked into the mesh vertices, so all initial body transforms are identity transforms. Body origins therefore need not coincide with visual part centers.

Table A-2: Environment tools shared by all evaluated systems. Coordinates use normalized scene units with world up $+Z$; quaternions use (w, x, y, z). Group identifiers select their constituent parts.

Tools	Information or operation
`list_objects`	Lists object and group identifiers.
`get_object`, `get_scene`, `get_state`	Reports body poses, local bounding extents, scene conventions, groups, and camera state. Mesh vertices and faces are unavailable.
`capture_scene`	Returns a 1024×768 image with a 38° vertical field of view.
`move_camera`	Orbits, zooms, or pans using yaw, pitch, zoom, and lateral/vertical offsets.
`translate_objects`	Applies a translation vector in world or camera coordinates.
`rotate_objects`	Applies X, Y, then Z angles in degrees, in world or camera coordinates, about a specified pivot. The default pivot is the mean of selected body origins.
`set_object_pose`	Sets a body's position and/or quaternion directly.
`group_objects`, `ungroup_objects`	Creates or dissolves transformation groups without changing relative part poses or snapping parts together.
`start_episode`	Marks the interaction as active.

Part identifiers are anonymized. Target poses, evaluator point clouds, and equivalence annotations are unavailable to the agent. Reference images are supplied separately: none for no-reference tasks, the final illustrated manual page for PartNet image conditions, all published pages for IKEA-Manual, and one rendered diagram per step for AssemblyBench.

A.2 Agents, Tools, and Instructions

The browser-based 3D environment exposes the same MCP tools to every system. Table A-2 summarizes observations and pose edits. The model selects operations and arguments, while its harness executes them and returns the outputs. Numeric queries provide body transforms and bounding extents but not mesh vertices or faces; visual geometry is accessed through rendered images. Pose edits return execution status, and a separate observation is required to inspect their effect.

Astra, Sol, and Terra use Codex CLI 0.155.1 with medium reasoning effort. Fable, Opus, and Sonnet use Claude Code 2.1.272 with medium effort. Qwen uses Codex 0.154.0 and DeepSeek uses Codex 0.155.0, both with the recorded xhigh setting and provider adapters. The comparison thus evaluates deployed model–harness combinations, including their differences in tool and image handling.

Execution. The benchmark allows 60 minutes of agent execution, excluding preparation, startup, and export. Available final states are evaluated after normal termination, errors, and timeouts. Documented server-capacity interruptions were retried in the OpenAI runs; retained states correspond to the latest executed attempt. Separate source-level runs do not all share this limit: the larger no-reference PartNet runs did not record a timeout setting. Execution status and geometric success are treated separately.

Task Instructions. All systems receive the following task text, with the reference lines substituted according to the condition. Harness defaults are retained, while project instructions and stored memories are disabled. The task text is shared across systems; their internal system prompts need not be identical.

```
Assemble the supplied parts into a coherent object.
[Reference-specific lines below.]
Inspect connections from multiple camera views using
capture_scene, correct gaps, orientation and obvious
interpenetration, and report uncertainties.  Do not claim
computed accuracy or physical stability without evidence.
Before finishing, check whether all supplied parts have been
incorporated into the assembly.
```

The reference-specific lines are:

- **No reference:** "No manual or reference image is supplied. Infer the assembly from the part geometry alone."
- **Assembled-object image:** "One reference image of the finished assembly is attached. Assemble the parts so that the result matches that image."
- **Manual:** "The attached images are the pages of the assembly manual, in order. Read them in order and follow them."

An additional execution instruction restricts interaction to the supplied environment tools, prohibits access to source geometry, ground truth, and other experiments, and prohibits resetting the scene or spawning other agents. Supplied references may be reread. The final report contains a completion status, summary, checked connections, and uncertainties. This execution instruction is summarized here; the quoted task text is verbatim.

Resource Measurements. Resource columns report arithmetic per-evaluation means, whereas assembly quality uses equal source weights. Elapsed run time includes intervals between calls and is not a measure of internal reasoning time. Claude Code reports cost directly; Codex cost is estimated from the recorded price schedule. These are usage estimates rather than subscription charges. Input-token accounting includes cache reads and, for Claude, cache creation; output tokens are added once. Records cover all 100 evaluations for Astra, Sol, Terra, Fable, and Sonnet, but only 88 for Opus, 78 for DeepSeek, and 34 for Qwen. Means exclude missing records and may therefore reflect outcome-dependent coverage.

A.3 Geometric Evaluation

Each part contributes 1,000 farthest-point samples selected from 4,096 area-weighted surface candidates. Evaluation normalizes the largest part's vertex-PCA bounding-box diagonal to one, independently of the initialization scale. For point sets X and Y, squared bidirectional Chamfer distance is

$$d_{\mathrm{CD}}(X,Y) = \frac{1}{|X|}\sum_{x\in X}\min_{y\in Y}\|x-y\|^2 + \frac{1}{|Y|}\sum_{y\in Y}\min_{x\in X}\|y-x\|^2. \tag{1}$$

SCD applies this distance to the assembled point sets. A single rigid registration aligns the complete prediction to the target while preserving relative part placement. We select the lowest SCD found from 24 proper PCA initializations and same-ID part-pose initializations, each refined by at most 100 symmetric ICP (iterative closest point) iterations with improvement tolerance 10^{-9}.

Geometrically equivalent parts form connected components of pairwise proper-rigid shape matches with CD $\leq 10^{-4}$. Hungarian assignment within each group minimizes the sum of part CDs. Given matched errors e_i,

$$\mathrm{PA} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}[e_i \leq 0.01], \qquad \mathrm{SR} = \mathbf{1}[\max_i e_i \leq 0.01]. \tag{2}$$

No part is independently repositioned to improve its scored pose. Registration is approximate, and equivalence uses the transitive closure of pairwise matches. These definitions are fixed across systems. For source means $\bar{s}$, aggregation is

$$\mathrm{Overall} = \tfrac{1}{4}\left[\tfrac{1}{2}(\bar{s}_{\mathrm{PN,none}} + \bar{s}_{\mathrm{PN,image}}) + \bar{s}_{\mathrm{IKEA}} + \bar{s}_{\mathrm{AB}} + \bar{s}_{\mathrm{FB}}\right]. \tag{3}$$

PA and benchmark quality curves use the same weighting. All benchmark tasks have scored final states; under the benchmark rule, a missing outcome would receive zero PA and SR, while SCD requires finite geometry.

Source-Level Comparisons. The PartNet comparison includes joint-guided assembly in Joint-PA (Li et al., 2024) and co-creation-space assembly in CCS (Zhang et al., 2024). The Fantastic Breaks evaluation samples 5,000 area-weighted surface points across parts, with at least 20 per part. Its scale is $\max(1, e_{\max})$, where $e_{\max}$ is the largest source per-part axis-aligned extent. A shared transform aligns the largest sampled part to its target, with that anchor included in the average and fixed part identities. For wrapped intrinsic-XYZ Euler-angle differences $\Delta\phi_i$ and sampled-centroid differences Δc_i,

$$\mathrm{RE} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{\|\Delta\phi_i\|^2}{3}}, \qquad \mathrm{TE} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{\|\Delta c_i\|^2}{3}}. \tag{4}$$

RE is in degrees and differs from geodesic rotation error. PA uses strict part CD < 0.01; shape CD uses global nearest neighbors, averaged within parts and then equally across parts. TE and CD are displayed at $\times 100$ and $\times 1000$. Published Jigsaw, PuzzleFusion++ (PF++) (Wang et al., 2025b), and GARF results follow GARF Table 3; RPF, TORA-CKA (Lee et al., 2026), and SARe-Gen follow

SARe v2 Table 3. Their original evaluation implementations are retained. Our anchor alignment and sampled inputs do not constitute a matched reproduction of those published runs; the four-way refinement experiment instead uses matched inputs throughout.

IKEA Chair/Table baselines follow Manual-PA Table 1. Full and Bench/Chair/Desk comparisons, including TwoByTwo (Qi et al., 2025), follow AssemLM v2 Tables 5–6 and Table 2. The alternative SR^{a} uses SCD < 0.02 before the display multiplier, with each method's normalization. It measures whole-shape proximity rather than requiring every part to be correct.

B Additional Quantitative Results

B.1 Uncertainty and Reference Effects

Tables A-3–A-5 report benchmark quality, paired system differences, and image-reference effects. Percentile bootstrap intervals use 10,000 draws with seed 20260921, resampling objects within each source. Both PartNet conditions remain paired, and system comparisons use identical resampled objects and the benchmark weights. These pointwise intervals describe object-sampling variation within the recorded runs, not repeated-run reliability. An all-zero interval does not establish zero population success probability.

Pairwise comparisons use two-sided randomization tests with 100,000 draws and the add-one correction. System labels are exchanged within objects, jointly for both PartNet conditions. Holm correction covers 56 tests: 28 system pairs and two metrics. Confidence intervals remain pointwise. Astra's descriptive advantage over Fable is not significant under this family-wise correction for either metric.

B.2 Geometry, Complexity, and Metric Sensitivity

Relative Part Size. Table A-6 measures each object's largest-to-smallest part PCA bounding-box diagonal, using unique mesh vertices. Whole-object normalization preserves this ratio. AssemblyBench has a higher median than the furniture sets, although the difference is not uniform across categories: PartNet Storage has a 90th percentile of 17.68, exceeding AssemblyBench's 15.73. These descriptive differences do not isolate the effect of part scale on performance.

Part Count. Table A-7 reports PartNet results in fixed bands of 2–5, 6–10, and 11–20 parts. Category and reference conditions remain separate. Table-category PA and SR decline across bands, whereas Chair/no-reference is not monotonic. Storage's lowest band contains only one object. The comparisons reflect joint changes in geometry, part count, and the all-parts success criterion, rather than a controlled manipulation of complexity.

SCD Concentration. Separated parts can remain far apart in free space even after whole-object registration, producing large squared distances. Table A-8 reports the contribution of the largest $\lceil 0.05N \rceil$ SCD values in the five analyzed PartNet conditions. All objects remain in the reported means. For Storage/image, eight objects contribute 99.32% of summed SCD; the median is 0.53 despite a mean of 184.80. Six of these eight retain every initial part pose, and the other two retain most of them. Normal execution termination does not exclude geometric failure. This source of

Table A-3: Source-weighted benchmark quality and pointwise paired-object bootstrap intervals (percent). All 100 tasks per system contribute, including partial assemblies after errors or timeouts.

System	SR	95% CI	PA	95% CI	Errors/timeouts
GPT-6 Astra	59.4	[50.0, 68.1]	80.9	[75.0, 86.3]	0
Claude Fable	50.0	[41.2, 58.8]	74.1	[67.8, 80.0]	0
Claude Opus	44.4	[34.4, 54.4]	63.1	[55.1, 70.8]	12
GPT-5.6 Sol	11.2	[5.0, 17.5]	30.6	[23.5, 37.7]	0
Claude Sonnet	7.5	[2.5, 12.5]	14.4	[9.0, 20.0]	0
GPT-5.6 Terra	7.5	[2.5, 12.5]	11.2	[6.2, 16.2]	0
Qwen Max	11.9	[5.6, 18.1]	15.9	[9.8, 22.3]	79
DeepSeek Flash	0.0	[0.0, 0.0]	0.0	[0.0, 0.0]	51

Table A-4: All system differences (A minus B, percentage points), with pointwise paired-object 95% intervals. Randomization-test p values use Holm correction across all 56 system-pair/metric comparisons; intervals remain pointwise.

A	B	ΔSR [95% CI]	p_{Holm}	ΔPA [95% CI]	p_{Holm}
Astra	Fable	+9.4 [0.6, 18.1]	0.581	+6.8 [1.3, 12.3]	0.263
Astra	Opus	+15.0 [5.6, 24.4]	0.064	+17.8 [10.5, 25.4]	< 0.001
Astra	Sol	+48.1 [38.1, 58.1]	< 0.001	+50.3 [41.8, 58.7]	< 0.001
Astra	Sonnet	+51.9 [41.9, 61.9]	< 0.001	+66.5 [59.2, 73.6]	< 0.001
Astra	Terra	+51.9 [40.6, 62.5]	< 0.001	+69.7 [61.5, 77.5]	< 0.001
Astra	Qwen	+47.5 [37.5, 57.5]	< 0.001	+65.0 [57.4, 72.5]	< 0.001
Astra	DeepSeek	+59.4 [50.0, 68.1]	< 0.001	+80.9 [75.0, 86.3]	< 0.001
Fable	Opus	+5.6 [-2.5, 13.8]	1.000	+11.0 [4.5, 17.8]	0.021
Fable	Sol	+38.8 [28.7, 48.1]	< 0.001	+43.5 [35.3, 51.5]	< 0.001
Fable	Sonnet	+42.5 [33.1, 51.9]	< 0.001	+59.7 [52.1, 67.1]	< 0.001
Fable	Terra	+42.5 [32.5, 52.5]	< 0.001	+62.9 [55.3, 70.2]	< 0.001
Fable	Qwen	+38.1 [28.7, 48.1]	< 0.001	+58.2 [50.4, 65.7]	< 0.001
Fable	DeepSeek	+50.0 [41.2, 58.8]	< 0.001	+74.1 [67.8, 80.0]	< 0.001
Opus	Sol	+33.1 [23.1, 43.8]	< 0.001	+32.5 [23.6, 41.2]	< 0.001
Opus	Sonnet	+36.9 [25.6, 48.1]	< 0.001	+48.7 [39.2, 57.9]	< 0.001
Opus	Terra	+36.9 [25.6, 48.1]	< 0.001	+51.9 [43.2, 60.5]	< 0.001
Opus	Qwen	+32.5 [22.5, 42.5]	< 0.001	+47.2 [38.7, 55.6]	< 0.001
Opus	DeepSeek	+44.4 [34.4, 54.4]	< 0.001	+63.1 [55.1, 70.8]	< 0.001
Sol	Sonnet	+3.8 [-1.2, 10.0]	1.000	+16.2 [8.4, 24.3]	0.003
Sol	Terra	+3.8 [-3.8, 11.2]	1.000	+19.4 [12.3, 26.6]	< 0.001
Sol	Qwen	-0.6 [-6.9, 5.6]	1.000	+14.7 [8.1, 21.6]	0.002
Sol	DeepSeek	+11.2 [5.0, 17.5]	0.064	+30.6 [23.5, 37.7]	< 0.001
Sonnet	Terra	+0.0 [-6.2, 7.5]	1.000	+3.2 [-3.8, 10.2]	1.000
Sonnet	Qwen	-4.4 [-10.6, 1.9]	1.000	-1.4 [-8.7, 5.6]	1.000
Sonnet	DeepSeek	+7.5 [2.5, 12.5]	0.407	+14.4 [9.0, 20.0]	< 0.001
Terra	Qwen	-4.4 [-11.2, 1.9]	1.000	-4.7 [-11.4, 1.7]	1.000
Terra	DeepSeek	+7.5 [2.5, 12.5]	0.407	+11.2 [6.2, 16.2]	0.002
Qwen	DeepSeek	+11.9 [5.6, 18.1]	0.035	+15.9 [9.8, 22.3]	< 0.001

Table A-5: Paired image-conditioned minus no-reference results on the same 20 PartNet objects. Up/down counts exclude ties; SR counts are discordant successes. PA intervals are pointwise percentile bootstrap intervals.

System	ΔPA (pp)	95% CI	PA up/down	SR up/down
GPT-6 Astra	+20.6	[9.5, 32.6]	12/2	5/0
Claude Fable	+19.3	[5.5, 34.3]	13/5	4/0
Claude Opus	+15.6	[2.0, 30.0]	13/4	3/0
GPT-5.6 Sol	+2.7	[-5.1, 11.2]	6/4	0/0
Claude Sonnet	-3.9	[-11.7, 0.0]	0/1	0/0
GPT-5.6 Terra	+0.0	[0.0, 0.0]	0/0	0/0
Qwen Max	-12.5	[-24.1, -4.0]	0/8	0/1
DeepSeek Flash	+0.0	[0.0, 0.0]	0/0	0/0

Table A-6: Within-object part-size ratios on the evaluated source-level sets. Each ratio divides the largest part PCA bounding-box diagonal by the smallest. PartNet assigns equal weight to objects across all three categories.

Dataset	N	Median	Interquartile range	90th percentile	Ratio > 10 (%)
AssemblyBench	279	4.39	2.29–8.90	15.73	20.1
IKEA-Manual	102	2.20	1.67–3.28	4.52	3.9
PartNet	1472	1.79	1.00–3.22	8.32	8.6

Table A-7: Part-count bands within complete PartNet category–reference conditions. Each cell shows sample count and mean PA/SR in percent. Bands keep objects with equal part counts together; category and reference conditions are not pooled.

Condition	2–5 parts	6–10 parts	11–20 parts
Chair / None	40: 51.5/25.0	358: 61.9/28.5	393: 50.4/17.0
Table / None	136: 85.8/79.4	237: 73.5/49.4	160: 55.9/22.5
Table / Image	136: 93.7/86.8	237: 85.3/66.7	160: 78.3/39.4
Storage / None	1: 60.0/0.0	42: 58.5/28.6	105: 46.6/6.7
Storage / Image	1: 100.0/100.0	42: 69.7/33.3	105: 61.2/14.3

Table A-8: Concentration of PartNet SCD for Astra. The largest $k = \lceil 0.05N \rceil$ values define the upper tail; its contribution is the fraction of summed SCD. Mean and median use all objects; the last column excludes the upper tail only as a diagnostic. SCD uses the same scale as Table 2.

Setting	N	Mean	Median	k	Contribution (%)	Remaining mean
Chair / NR	791	69.90	2.88	40	94.62	3.96
Table / NR	533	46.98	1.21	27	94.53	2.71
Storage / NR	148	141.60	0.58	8	98.87	1.70
Table / IR	533	20.95	0.49	27	96.02	0.88
Storage / IR	148	184.80	0.53	8	99.32	1.33

Table A-9: IKEA-Manual results by category, before the equal-category averaging in Table 3. † denotes Image-PA retrained on diagrams.

Method	Condition	SCD ↓		PA ↑		SR ↑	
		Chair	Table	Chair	Table	Chair	Table
3DHPA$_{\text{CVPR'24}}$	–	34.3	37.8	1.914	4.027	0.000	0.000
Image-PA$_{\text{ECCV'20}}$ †	Diagram	17.3	14.7	19.07	36.74	0.000	10.53
Manual-PA$_{\text{ICCV'25}}$	Manual	11.4	4.8	42.51	49.72	3.509	15.79
GPT-6 Astra + Codex	Manual	**1.2**	**0.4**	**91.07**	**95.47**	**71.93**	**78.95**

error differs from bounded translation predictors such as DGL, whose released pose head uses $\tanh$; input normalization alone does not impose an output bound.

Category Results and Thresholds. Table A-9 provides the separate IKEA Chair and Table results underlying the equally weighted summary. Figure A-1 varies the correctness threshold with registration and correspondence fixed. This isolates cutoff sensitivity from changes in alignment or equivalence grouping.

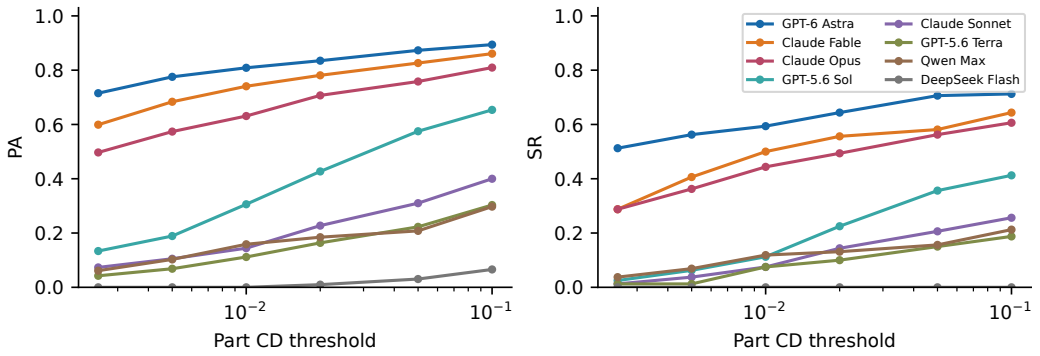


Figure A-1: Correctness-threshold sensitivity for all eight systems. PA/SR use equal source weights; saved registration and Hungarian correspondence remain fixed. The input-shape equivalence threshold remains fixed.

C Interaction and Failure Analysis

C.1 Interaction Dynamics

Each evaluation is divided into ten equal-width bins from its first to last environment call. Activity shares are computed within each active evaluation-bin and then averaged with benchmark source weights. Inspect includes object and state queries; Observe includes camera moves and captures; Manipulate includes pose edits. Grouping and episode-control calls remain in the denominator. Bins without calls are omitted and available weights renormalized. Thus, evaluations with more calls do not automatically dominate the activity curves.

For each bin, manipulation magnitude is first summarized by its median within an evaluation and then by the weighted median across evaluations. Translation measures part-centroid displacement normalized by the largest-part diagonal; rotation measures angular displacement. Operations below $0.5°$ enter translation summaries, and other operations enter rotation summaries. Group edits contribute the maximum magnitude among affected parts. Missing actions are not assigned zero magnitude. Table A-10 applies the same procedure to the first and last thirds of each trajectory. Smaller late-stage motions occur in both successful and failed runs, so they do not by themselves establish successful refinement.

Quality and Time Budgets. For quality curves, we score the state after the last completed operation at each normalized checkpoint or absolute time budget. The initial state applies before the first operation and the final state persists after termination; scores are not interpolated between checkpoints. PA and SR use all evaluations with the benchmark weights. Figure A-2 reports retrospective absolute-time truncation for all eight systems. Five episodes have edits after 60 minutes of end-to-end time, which includes startup outside the agent-execution budget; their untruncated outcomes are shown separately. PA and SR nevertheless agree at 60 minutes and termination for all systems.

Table A-10: Early-to-late interaction changes. Each entry is a source-weighted median of per-evaluation action medians within the first or last third of the recorded interaction interval. Translation is normalized by the largest-part diagonal; rotation is in degrees. Revisit is the median fraction of edits affecting a previously edited part.

System	Trans. early	Late	Rot. early	Late	Revisit (%)
Astra	0.573	0.051	81.7	44.0	81.7
Fable	1.072	0.053	72.5	19.0	87.5
Opus	1.529	0.332	73.8	45.0	88.1
Sol	0.827	0.273	90.0	85.0	88.9
Sonnet	1.984	0.916	90.0	90.0	90.0
Terra	0.386	0.390	90.0	90.0	75.0
Qwen	0.819	0.806	90.0	90.0	87.5
DeepSeek	0.532	0.143	90.0	90.0	94.3

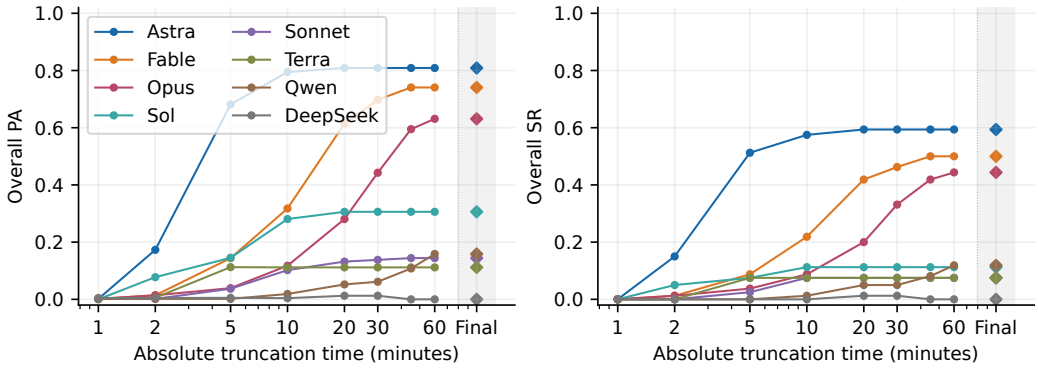


Figure A-2: Recorded-state truncation at absolute budgets. Earlier termination retains the final state. Diamonds in the shaded Final column show untruncated outcomes, not an additional time budget; five episodes have pose changes after 60 minutes of end-to-end time. Terminal values agree with the result tables. These are retrospective curves, not new budget-conditioned agent runs.

Table A-11: Geometric diagnostic denominators and translation-dominated error fractions. Part pooling weights incorrect parts equally; task pooling first averages within each failed task. Categories follow the ordered classifier.

System	Failed tasks	Wrong parts	Part pooled (%)	Task pooled (%)
GPT-6 Astra	47	236	67.4	68.5
Claude Fable	57	309	75.1	75.3
Claude Opus	61	400	67.8	64.7
GPT-5.6 Sol	91	631	46.4	50.9
Claude Sonnet	94	740	41.4	42.2
GPT-5.6 Terra	94	757	34.9	38.6
Qwen Max	90	730	34.4	36.8
DeepSeek Flash	100	778	32.9	38.2

A drop in PA can reflect a changed global registration or interchangeable-part assignment. Among 239 decreases between sampled checkpoints, 96 change at least one assignment; 140 decreases persist with alignment fixed, and 137 persist with both alignment and matching fixed. Intermediate accuracy is therefore interpreted alongside the actual manipulation, as in the inspection example in Figure 3.

Completion Reports. Astra succeeds in 43 of 74 declared completions, and Fable in 39 of 88. Among unsuccessful completion claims, median incorrect-part fractions are 33.3% and 42.9%, respectively. Missing reports are excluded. These counts compare categorical declarations with geometric success; they do not measure probabilistic confidence calibration.

C.2 Execution and Geometric Failures

Execution and geometric outcomes are distinct. Astra and Fable terminate normally in all 100 benchmark tasks. Qwen has 65 timeouts and 14 execution failures; DeepSeek has 17 and 34, respectively. All 49 normally terminated DeepSeek runs fail geometrically. Unsupported-tool and stream-disconnection errors occur in both systems, sometimes within the same run. These observations establish failures of the evaluated model–harness combinations, without attributing every error to the model's geometric reasoning.

The cases in Figure 5 distinguish logical grouping from assembly, numerical pose agreement from visible alignment, and a residual positioning error in an otherwise assembled object. They were selected to illustrate these mechanisms, not their prevalence. In another DeepSeek case, the agent reports receiving no viewable image despite image payloads in the recorded tool responses; whether those images reached the model through its provider remains unresolved.

Geometric Classification. We apply the saved global alignment and equivalent-part matching before assigning the first applicable category: correct (CD ≤ 0.01); never moved; far displaced (centroid offset > 3 largest-part diagonals); translation-dominated (centered CD ≤ 0.01 and offset > 0.1); near-threshold (centered CD ≤ 0.01 and offset ≤ 0.1); orientation-dominated (centered CD > 0.01 and offset ≤ 0.1); or mixed. Centering removes translation while preserving orientation and accommodates geometric symmetry through shape distance.

Table A-11 gives denominators and compares part-pooled with task-pooled translation-error fractions. Varying the offset cutoff over $0.05, 0.1, 0.2$ and centered-CD cutoff over $0.005, 0.01, 0.02$ yields translation-dominated fractions of 48.3–78.0% for Astra and 50.5–86.1% for Fable. Their default values are 67.4% and 75.1%. Positioning error is a useful diagnostic, but the majority classification is not invariant to the thresholds.

C.3 Qualitative Comparisons and Manual Use

Figures A-3 and A-4 compare all eight systems on four common objects. Objects are selected from Astra's benchmark blocks by proximity to median PA, breaking ties by proximity to median part count and then object identifier. These examples supplement the quantitative comparison rather than estimate failure frequencies. Predictions are globally aligned to the targets; view directions and part colors are shared, but each rendering is fitted to its geometry.

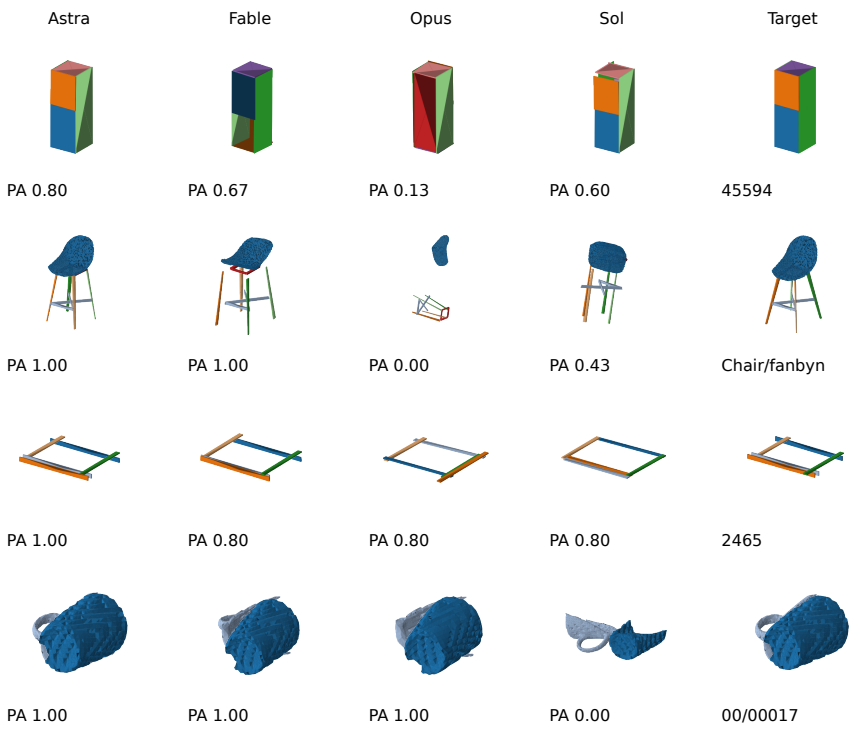


Figure A-3: Four common objects selected near Astra's median PA in each source, evaluated by Astra, Fable, Opus, and Sol. Columns retain a shared view direction and per-part colors; each view is fitted to its own geometry, so apparent size is not a shared physical scale. Numbers are PA under the common benchmark protocol. The other four systems appear in Figure A-4.

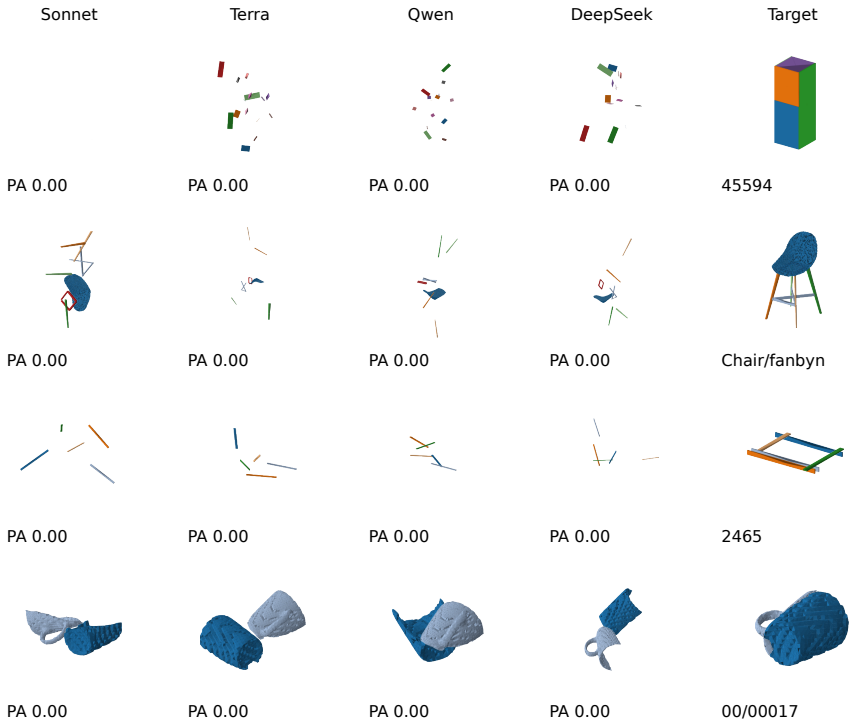


Figure A-4: The same four objects evaluated by Sonnet, Terra, Qwen, and DeepSeek, with the same rendering conventions as Figure A-3. The target column repeats the object identities to make both comparisons independently readable.

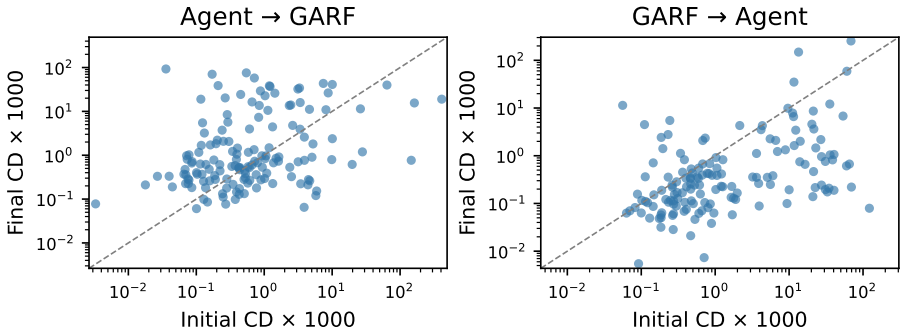


Figure A-5: Per-object changes in shape CD for the two composition orders. Each point is one of the same 150 objects; points below the diagonal improve. Both axes are logarithmic.

Figure 3 instead uses cases selected for visible intermediate manipulations. Its offline renderings preserve recorded part poses and camera directions with centered framing; they are not the original images observed by the agent. Display grids and axes aid interpretation. Reference images correspond to the supplied source materials.

In the IKEA bench example, the agent places the seat and both end frames before adding the stretcher, whereas the manual introduces the stretcher with the first end frame. In the industrial example, provisional placements are revised before and after inspecting the covered blade. These trajectories show correspondence to instructional relationships without reproducing every illustrated step. A different placement order alone does not establish failure to use a manual. The two selected successful cases do not establish a population-level instruction-following rate.

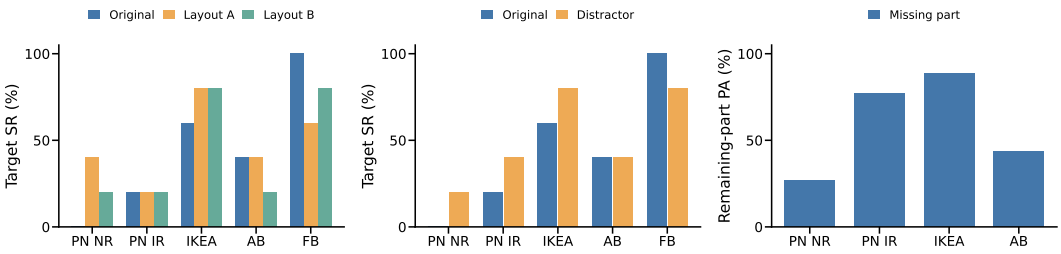
D Combining Agents with Geometric Models

Matched Comparison. All four configurations use the same 150 two-fragment Fantastic Breaks objects, fixed point samples, part identities, normalization, and anchor-aligned evaluator. Each object has 5,000 area-weighted surface samples with at least 20 per part and sampling seed 42. The largest sampled part is the anchor. Standard GARF is rerun on these inputs; its matched result in Table 6 is distinct from the published GARF result in Table 5.

Both GARF configurations use the released GARF-mini checkpoint without training or tuning. Standard inference uses random initialization and a one-step initialization stage, followed by 20 refinement steps. Agent-initialized inference replaces the initialization stage with the agent poses and retains the same refinement schedule. A deterministic per-object flow seed is derived from seed 42 and the object identifier. One rigid transformation aligns the agent anchor to its target while preserving relative part poses; the anchor remains fixed during inference. No non-anchor target pose is supplied.

For GARF-to-Agent, the predicted poses initialize the original meshes with no preceding interaction history. Astra receives the original no-reference prompt and uses medium effort through Codex, without target geometry or the GARF trajectory. These runs follow the original source-level fracture setting without a wall-time cutoff. All 150 terminate normally. The reconstructed initial states and final states use the same point samples and evaluator.

Paired Outcomes. Relative to standard GARF, agent initialization improves PA on ten objects and leaves it unchanged on 140. Relative to the unrefined agent, Agent-to-GARF improves PA on 15 objects, degrades it on 29, and leaves it unchanged on 106. Relative to standard GARF, GARF-to-Agent improves PA on 33 objects, degrades it on five, and leaves it unchanged on 112; shape CD improves on 125, worsens on 23, and is unchanged on two, using tolerance 10^{-8}. Figure A-5 shows both improvements and regressions. These single-run, two-fragment results do not establish seed-averaged superiority or transfer to multi-part industrial assemblies.



(a) Initial layouts (b) Distractor parts (c) Missing parts

Figure A-6: GPT-6 Astra under task variations on five fixed samples per benchmark block, with one run per condition. PartNet reference conditions share objects and perturbations. Missing-part evaluation measures only retained parts; Fantastic Breaks is excluded from this condition.

E Robustness Experiments

Conditions. Five fixed samples per benchmark block are selected using part-count strata, without selection on performance. PartNet uses two chairs, two tables, and one storage object, paired across reference conditions. The study contains 20 distinct objects and 25 baseline conditions. Each object receives two alternative layouts, one randomly removed part except on Fantastic Breaks, and one added part from a different object in the same source. PartNet conditions share the perturbations. Geometry, references, targets, and normalization remain fixed; inventory changes preserve unaffected parts' initial poses.

Selection and inventory changes use seed 20260924, with separate deterministic streams; layouts A and B use seeds 20260925 and 20260926. Each of the 95 new conditions is run once with medium-effort Astra, the original prompt, and a one-hour limit. All runs terminate normally. The unchanged prompt requests incorporation of all supplied parts and does not identify the perturbation. The 25 baselines are rescored using the same evaluator as the variants.

Evaluation. Layout variants use benchmark PA and SR. Distractor evaluation excludes the added part from registration, correspondence, and scoring. Missing-part evaluation scores retained parts against their targets at the original scale. Retained-part completeness is not complete-object success, and retained-part PA has a different denominator from full-object PA. Figure A-6 summarizes the results; Table A-12 preserves the object-level pairing.

Responses to Inventory Changes. We examine actions, final geometry, and reports for all 20 missing-part and 25 distractor runs. Four IKEA and three AssemblyBench missing-part reports identify an absent component. A completion declaration can coexist with a reported omission. Under distractors, four IKEA and three AssemblyBench reports identify an uninstalled part, although several such parts were never moved. On Fantastic Breaks, two runs attempt to incorporate all three fragments, two try the extra fragment and then move it aside, and one reports leaving it unincorporated. These behaviors distinguish recognition, attempted use, and non-incorporation without equating them with successful anomaly detection.

The unchanged all-parts instruction influences distractor handling. Without a reference, the intended inventory may also be underdetermined. The small paired sample and single run per condition support exploratory observations, not repeated-run reliability or robustness to larger numbers of missing or extra parts.

Table A-12: Paired task-variation results. Each cell reports PA (%) / SR (0 or 1); the missing-part column instead reports retained-part PA / retained-part completeness. A dash indicates an untested condition. Each condition is run once.

Source	Object	Original	Layout A	Layout B	Distractor	Missing
PartNet NR	40074	81.8/0	100.0/1	81.8/0	72.7/0	30.0/0
	2738	0.0/0	0.0/0	0.0/0	0.0/0	0.0/0
	23814	18.2/0	18.2/0	18.2/0	18.2/0	10.0/0
	22320	76.9/0	100.0/1	100.0/1	100.0/1	75.0/0
	46475	54.5/0	54.5/0	54.5/0	27.3/0	20.0/0
PartNet IR	40074	81.8/0	81.8/0	81.8/0	100.0/1	60.0/0
	2738	76.9/0	53.8/0	69.2/0	76.9/0	75.0/0
	23814	36.4/0	9.1/0	36.4/0	9.1/0	60.0/0
	22320	100.0/1	100.0/1	100.0/1	100.0/1	91.7/0
	46475	45.5/0	72.7/0	72.7/0	90.9/0	100.0/1
IKEA-Manual	Bench/applaro	100.0/1	100.0/1	100.0/1	100.0/1	100.0/1
	Chair/voxlov	66.7/0	66.7/0	66.7/0	66.7/0	80.0/0
	Chair/jokkmokk	100.0/1	100.0/1	100.0/1	100.0/1	85.7/0
	Table/bjorkudden	100.0/1	100.0/1	100.0/1	100.0/1	87.5/0
	Shelf/laiva	57.9/0	100.0/1	100.0/1	100.0/1	88.9/0
AssemblyBench	4451	100.0/1	100.0/1	100.0/1	100.0/1	0.0/0
	2465	100.0/1	100.0/1	60.0/0	100.0/1	0.0/0
	7780	60.0/0	60.0/0	80.0/0	40.0/0	100.0/1
	4492	42.9/0	42.9/0	57.1/0	71.4/0	50.0/0
	8243	46.2/0	46.2/0	46.2/0	46.2/0	66.7/0
Fantastic Breaks	01/01007	100.0/1	100.0/1	100.0/1	100.0/1	–
	02/02002	100.0/1	100.0/1	100.0/1	100.0/1	–
	02/02007	100.0/1	0.0/0	0.0/0	0.0/0	–
	09/09028	100.0/1	100.0/1	100.0/1	100.0/1	–
	18/18002	100.0/1	50.0/0	100.0/1	100.0/1	–