SPARC: COMPILING VISUAL ASSEMBLY INTENT INTO CAD ASSEMBLIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Automatically assembling a set of existing CAD parts from a single reference image requires both visual reasoning and geometric verification. A vision-language model can infer part roles, relative layouts, and likely connections, but these predictions may not be supported by the actual part geometry. Conversely, B-Rep geometry can verify and realize assembly relations, yet often admits multiple plausible interface assignments. We present SPARC, a framework that requires no task-specific model training and formulates reference-guided CAD assembly as a compilation process from visual assembly intent to executable CAD constraints. Given an unordered set of B-Rep parts and a reference image, SPARC first generates a structured visual assembly proposal. Relation-specific geometric routines resolve feature references and construct candidate assembly plans. Assembly-level selection compares these plans using geometric evidence and cross-part consistency checks. Pre-execution preparation may further revise relations and interface bindings before compiling the resulting state into native CAD constraints. FreeCAD executes the program and returns execution and interference diagnostics for bounded local repair. We further introduce SPARC-Bench, a benchmark of 310 assemblies constructed from two public CAD assembly datasets. Experiments evaluate visual consistency, executed assembly-graph validity, interference, and end-to-end assembly success. SPARC achieves a 94.84% end-to-end assembly success rate on SPARC-Bench. The benefit of plan replacement increases across the evaluated interface-ambiguity groups, supporting the value of resolving local geometric choices at the assembly level.

1 INTRODUCTION

Recent CAD generation methods produce command sequences, constrained sketches, and modeling programs (Wu et al., 2021; Seff et al., 2022; Khan et al., 2024; Zhang et al., 2025). Planning and execution feedback further support the generation and refinement of individual CAD parts (Alrashedy et al., 2025; Guo et al., 2026; Gong et al., 2026). Assembling existing parts, however, poses a different challenge: the geometry is given, but its intended connections must be inferred. Image-guided methods recover configurations of supplied 3D parts (Li et al., 2020; Wang et al., 2025; Zhao et al., 2025), while assembly-generation systems can synthesize parts together with their connection specifications (Shui et al., 2026; Dong et al., 2026). We consider a fixed set of B-Rep parts and a single reference image, seeking the relations and geometric interfaces needed to express the target assembly as an executable CAD constraint program.

A reference image provides evidence of the intended assembly. Vision-language models (VLMs) can infer part roles, likely connections, and relative layouts (Cho et al., 2025; Chen et al., 2024a). These cues, however, do not directly establish which B-Rep entities should be connected. A shaft may be visually associated with a housing while several cylindrical faces remain plausible mating interfaces. Reported limitations in localization, orientation, and multi-object spatial reasoning further motivate explicit geometric verification (Covert et al., 2025; Ma et al., 2025).

Geometry does not eliminate this ambiguity by itself. AutoMate predicts mating entities and mate types (Jones et al., 2021), while JoinABLe predicts pairwise joints and searches for relative part placements (Willis et al., 2022). Repeated holes, compatible axes, and similar planar faces can support multiple locally feasible bindings. More importantly, individually compatible choices may

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

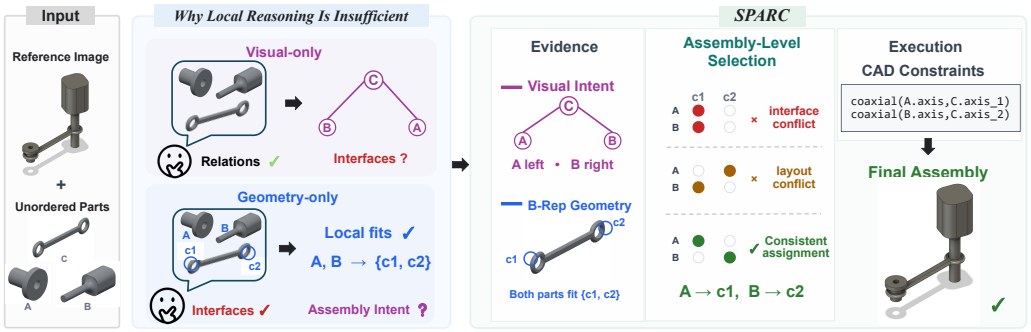


Figure 1: **Motivation for assembly-level selection.** Visual reasoning may leave interfaces unresolved, while geometry alone may admit multiple assignments. SPARC combines visual intent and geometric evidence to resolve conflicting assignments before CAD execution.

compete for the same interface or imply conflicting placements of a shared part. The central difficulty is therefore not finding a compatible interface in isolation, but selecting relations and bindings that jointly realize the reference assembly. Figure 1 illustrates this distinction between local compatibility and assembly-level consistency.

This observation motivates SPARC, which treats reference-guided assembly as compilation from visual intent to executable CAD constraints without task-specific model training. A VLM provides a structured proposal of typed part relations, and relation-specific geometric routines resolve candidate interfaces and construct alternative assembly plans. These alternatives may revise the proposed relations as well as their interface bindings. SPARC compares complete plans using geometric evidence and shared-part, interface-reuse, and repeated-site consistency checks, applying a replacement only when the candidate meets the required evidence and score conditions. The retained plan undergoes pre-execution preparation before compilation into native CAD joints. Execution and interference diagnostics then guide bounded local repair of interface bindings and relation parameters.

To evaluate this setting, we introduce SPARC-Bench with 310 assemblies: 230 from AutoMate (Jones et al., 2021) and 80 from the Fusion 360 Gallery Assembly dataset (Willis et al., 2022). Each case provides an unordered set of sanitized B-Rep parts and a single reference image, without exposing source mates, assembly hierarchy, instance poses, or interface annotations to inference. SPARC achieves a 94.84% end-to-end assembly success rate. Component ablations examine the contributions of plan replacement and CAD-feedback repair, while ambiguity analysis shows that the benefit of plan replacement is greatest in the high-ambiguity group.

Our contributions are threefold: (i) We formulate reference-guided assembly of fixed B-Rep parts as compilation into solver-executable CAD constraints. (ii) We develop assembly-level plan selection that compares alternative relation assignments and interface bindings through geometric evidence, cross-part consistency checks, and conditional replacement. (iii) We introduce SPARC-Bench, comprising 310 assemblies and an evaluation protocol covering assembly-graph validity, visual consistency, interference, and end-to-end assembly success.

2 Related Work

2.1 Structured and Executable CAD Generation

CAD generation methods increasingly represent designs through structured representations and executable programs. DeepCAD models parametric CAD as sequences of sketch and extrusion commands (Wu et al., 2021), while Vitruvion generates sketches with geometric primitives and explicit constraints (Seff et al., 2022). Text2CAD conditions CAD command sequences on natural-language descriptions (Khan et al., 2024), and FlexCAD supports controllable generation across CAD modeling hierarchies (Zhang et al., 2025). Recent methods further introduce intermediate planning representations: Plan then Act separates high-level operation planning from low-level command generation (Guo et al., 2026), while Graph-CAD uses a hierarchical, geometry-aware graph to guide

action planning and code generation (Gong et al., 2026). CADCodeVerify uses visual feedback from rendered outputs to iteratively verify and refine generated CAD code (Alrashedy et al., 2025). These approaches construct or modify geometry through structured modeling programs. SPARC instead preserves the supplied B-Rep geometry and infers the connections and placements needed to realize a reference assembly. Its task is to identify compatible interfaces in existing parts, rather than generate the geometry that realizes a design specification.

2.2 Geometry-Based CAD Assembly Reasoning

AutoMate learns B-Rep representations to predict mating entities and mate types from annotated CAD assemblies (Jones et al., 2021), while JoinABLe predicts joint axes between B-Rep part pairs and searches for their relative placements (Willis et al., 2022). These methods use explicit CAD geometry for local interface reasoning without a reference image specifying the target configuration. Linkify augments assembly graphs with local contact geometry for context-aware part retrieval (Jignasu & Grandi, 2026), but its interface information is extracted from existing assemblies. SPARC instead infers which interfaces should connect the supplied parts from visual and geometric evidence. Repeated holes, symmetric axes, and similar planar faces can yield multiple locally feasible connections that conflict at the assembly level. SPARC therefore separates typed relation proposals from B-Rep interface grounding, enabling local compatibility checks and assembly-level consistency reasoning.

2.3 Image-Guided Assembly and Multimodal CAD Design

Image-guided assembly arranges supplied 3D parts to match a reference image. Li et al. (2020) combine image–part correspondences with graph reasoning to infer relationships and poses, while Imagine progressively refines an image-derived structure knowledge graph (Wang et al., 2025). Assembler uses anchor-point diffusion for mesh-part assembly, including repeated parts and symmetric configurations (Zhao et al., 2025). These methods primarily recover part configurations rather than ground mating interfaces in native B-Rep geometry. Recent multimodal models have also improved spatial reasoning from visual inputs (Chen et al., 2024a), while benchmarks such as 3DSRBench reveal remaining limitations in orientation, relative position, and multi-object 3D reasoning (Ma et al., 2025). These observations motivate explicit geometric verification when visual predictions must be converted into executable CAD relations. OmniCAD evaluates component selection, pose estimation, mate-graph prediction, and tool-assisted refinement from multimodal assembly evidence (Wang et al., 2026). Other recent systems integrate structured connection specifications with CAD execution. ArtiCAD generates articulated assemblies from connector and joint specifications (Shui et al., 2026), while AssemCAD uses typed parts, geometry-backed ports, and executable mates followed by deterministic CAD verification (Dong et al., 2026). SPADA uses executable tests for iterative CAD verification and repair (Zheng et al., 2026), and Embodied CAD combines hierarchical skills with solver feedback for assembly planning (Liu et al., 2026). SPARC differs in its input and grounding setting. It takes a fixed inventory of B-Rep parts and a single reference image, without target poses or mating annotations. The method must infer the intended part relations, identify compatible interfaces in the supplied geometry, compare alternative assembly plans, and compile the retained bindings into executable CAD constraints.

3 SPARC

3.1 Problem Formulation

Figure 2 summarizes the overall SPARC pipeline. Given an unordered set of B-Rep parts $\mathcal{P} = \{p_i\}_{i=1}^{n}$ and a reference image I, SPARC seeks an executable CAD constraint program that realizes the target assembly while preserving the supplied part geometry. The VLM produces a structured proposal whose typed part relations form a graph $G_{\text{prop}} = (V, E_{\text{prop}})$. Each node represents a part instance, and each edge $e = (i, j, \tau)$ proposes a relation of type τ between parts p_i and p_j. Geometric processing resolves the proposal into an initial assembly plan H_0. Each plan specifies a base instance, placement steps, part relations, interface bindings, relation parameters, and candidate placements. The main stages are summarized as

$$(\mathcal{P}, I) \rightarrow G_{\text{prop}} \rightarrow H_0 \rightarrow H_{\text{out}} \rightarrow A \rightarrow \Pi \rightarrow R.$$

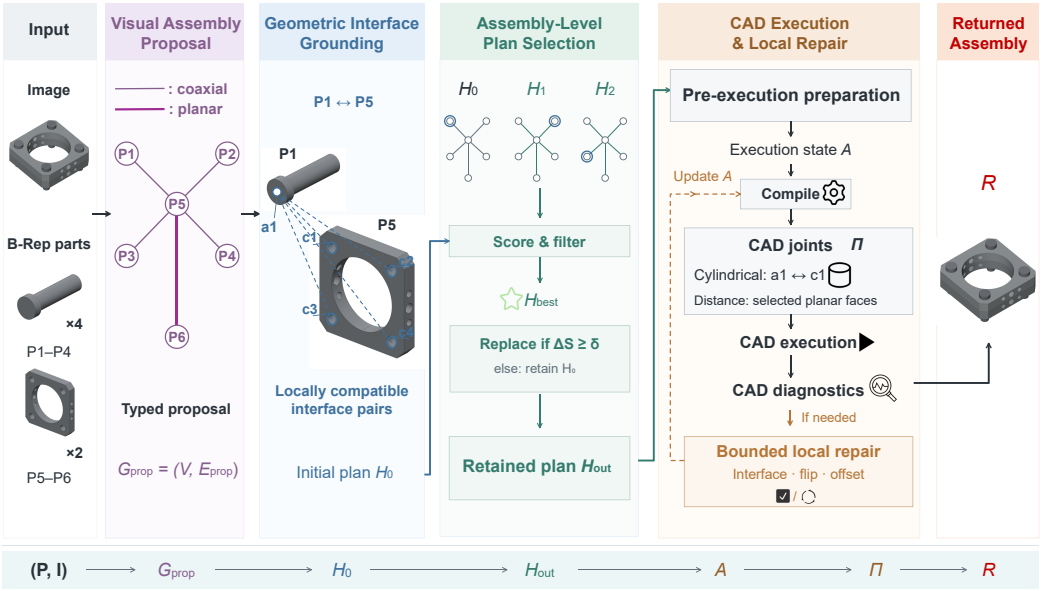


Figure 2: Overview of SPARC. A visual assembly proposal is grounded to B-Rep interfaces and resolved through assembly-level plan selection. The retained plan undergoes pre-execution preparation and compilation into native CAD joints. Execution diagnostics guide bounded local repair.

Here, H_{out} is the plan retained by assembly-level selection, A is the state obtained after pre-execution preparation, Π is the compiled CAD constraint program, and R is the result returned by CAD execution. Candidate construction and pre-execution preparation may revise the base, placement steps, part relations, and interface bindings. The relations used for execution therefore need not coincide with E_{prop}.

3.2 Visual Assembly Proposal

In the default planning call, the VLM receives the assembly reference image, geometric descriptions, and instance information, together with at most four representative part collages. Each collage contains four normalized views of a representative part. Parts are grouped using geometric summaries to reduce repeated prompt content, while all instance IDs, group memberships, and instance counts remain in the text input. The VLM returns a structured assembly proposal containing a base instance, ordered placement steps, relation types, and coarse spatial hints. Each step identifies a moving instance and an anchor and may reference supplied feature labels. The proposed part relations are represented as a typed graph $G_{\text{prop}} = (V, E_{\text{prop}})$. Geometric routines resolve and check the feature references, and assembly-level selection determines the retained plan. The VLM does not directly predict exact part transforms or generate native CAD constraints. Planner inputs and output fields are detailed in Appendix A.1.

3.3 Geometric Interface Grounding

The visual proposal expresses intended connections, but its feature references remain hypotheses until checked against the supplied geometry. Geometric grounding converts these references into locally compatible B-Rep interface candidates.

For each part p_i, we extract a set of interface candidates

$$\mathcal{A}_i = \{a_i^q\}_{q=1}^{N_i}.$$

An interface candidate can be a B-Rep face or a geometric reference derived from the B-Rep model, such as an axis or reference point. Each candidate is described by its type, position, orientation, dimensions, and local geometric context. For a relation $e = (i, j, \tau)$ considered during plan construction or preparation, the relation type τ determines the admissible interface types and relation-specific

local geometric compatibility tests. Candidate matches are generated as

$$\widetilde{\mathcal{C}}_e = \left\{ (a_i, a_j, \theta_e) \mid a_i \in \mathcal{A}_i,\ a_j \in \mathcal{A}_j,\ \chi_\tau^{\mathrm{local}}(a_i, a_j) \right\}.$$

This set contains locally compatible interface hypotheses before routine-specific ranking and candidate reduction. For example, coaxial relations check cylindrical dimensions and axis compatibility, whereas planar relations check face orientation and support geometry. After relation-specific ranking and candidate limits, the retained candidates are

$$\mathcal{C}_e = \left\{ c_e^k = (a_i^k, a_j^k, \theta_e^k) \right\}_{k=1}^{K_e},$$

where a_i^k and a_j^k are the matched interfaces and θ_e^k contains the relation-specific parameters required for the CAD constraint. The retained candidates are ranked by the corresponding relation-specific geometric routine. For example, cylindrical matching uses clearance, length ratio, and optional position hints. Each routine retains a finite candidate set $\mathcal{C}_e \subseteq \widetilde{\mathcal{C}}_e$, where K_e denotes the number of retained candidates. Ranking rules and candidate limits vary across routines. These local scores guide candidate exploration; they are not summed to score a complete assembly plan. Assembly-level selection is described in Section 3.4. Detailed ranking rules, geometric thresholds, and candidate limits are provided in Appendix A.2. These routines support both the initial proposal and the relations introduced or revised during candidate construction and pre-execution preparation.

3.4 Assembly-Level Plan Selection

Local compatibility does not resolve conflicts between connections that share parts or compete for interfaces. SPARC therefore evaluates complete assembly plans rather than treating each local binding as a final decision. Relation-specific geometric routines construct alternatives through finite, family-specific searches. These plans may differ in their base instance, placement order, part-pair relations, relation types, interface bindings, and parameters. The assembly-level score $S_{\mathrm{asm}}(H)$ considers coverage, geometric and semantic evidence, underconstrained placements, and conflicts. It evaluates the complete plan rather than summing local interface-ranking scores. Let H_0 denote the current plan. SPARC scores the current plan and the alternatives, then removes alternatives that fail validity or evidence checks. The evidence check $Q_{\mathrm{ev}}(H, H_0)$ requires an alternative H to contain at least as many geometrically supported steps as H_0, unless H_0 is rejected. The highest-scoring eligible plan, denoted by H_{best}, is selected, with score ties resolved by the lexicographic order of candidate names. It replaces H_0 only when

$$S_{\mathrm{asm}}(H_{\mathrm{best}}) - S_{\mathrm{asm}}(H_0) \geq \delta(H_{\mathrm{best}}, H_0).$$

Here, $\delta(H_{\mathrm{best}}, H_0)$ is the context-dependent replacement margin, and the score difference is denoted by ΔS in the figures. If no eligible plan exists or H_{best} fails this condition, H_0 is retained. Lower-scoring plans are not reconsidered. The retained plan, denoted by H_{out}, undergoes pre-execution preparation before CAD compilation and execution. Figure 3(a) illustrates the conditional replacement of an assembly plan. Detailed search procedures, validity checks, scoring terms, and replacement rules are provided in Appendix A.3.

3.5 CAD Execution and Bounded Local Repair

Pre-execution preparation. The retained plan H_{out} is converted into an execution state. Geometry-based preparation may revise the base, placement order, anchors, relation types, interface bindings, and parameters. These changes are distinct from feedback-driven repair after CAD execution. The resulting state A contains the relations and geometric bindings used for the first main CAD execution.

Constraint compilation and execution. Relation-specific mappings compile the relations, interface bindings, and parameters in A into native FreeCAD joints:

$$\Pi = \mathrm{Compile}(A), \qquad R = \mathrm{Execute}_{\mathrm{FreeCAD}}(\Pi).$$

Shaft–hole insertion uses a Cylindrical joint between the selected cylindrical faces, with an auxiliary Distance joint for axial seating when applicable. Planar seating uses a Distance joint between the selected planar faces. Geometric routines determine the interfaces and joint specifications, while the

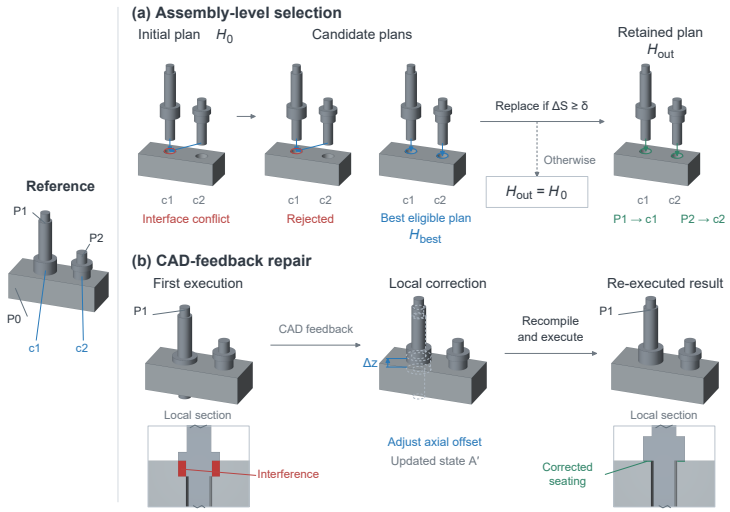


Figure 3: **Schematic examples of plan selection and CAD-feedback repair.** (a) The best eligible plan replaces H_0 only when its score gain ΔS meets the replacement margin δ; otherwise, H_0 is retained. (b) CAD diagnostics guide an axial-offset correction, followed by recompilation and CAD re-execution.

FreeCAD assembly solver adjusts part poses under these constraints. Execution returns solver and interference diagnostics.

Bounded local repair. A plan that passes selection checks may still fail CAD solving or produce excessive interference. SPARC therefore uses execution diagnostics to guide bounded local repair of the affected relations. Repair considers alternative interface bindings or revised parameters, including face or hole selection, orientation, flip, insertion depth, and offset. The standard direct repair operations preserve the current base, placement order, part-pair relations, and relation types. Repair may repeat geometric solving for the affected steps. Within the same assembly run, the updated execution state is recompiled and executed without repeating the full pre-execution preparation. CAD diagnostics determine whether the trial satisfies the configured recovery criteria. Repair stops when recovery succeeds or the configured budget is exhausted. Accepting an intermediate improvement does not establish final assembly success, which requires the evaluation criteria in Section 4.1. Preparation rules, relation-specific execution, repair budgets, and trial acceptance criteria are provided in Appendix A.4.

4 Experiments

4.1 Experimental Settings

(1) SPARC-Bench. We evaluate on SPARC-Bench, which contains 310 assemblies: 230 from AutoMate (Jones et al., 2021) and 80 from the Fusion 360 Gallery Assembly dataset (Willis et al., 2022). Each case provides an unordered set of B-Rep parts and a single reference image. Original mates, assembly hierarchy, instance poses, and interface annotations are not provided to the inference pipeline. Dataset construction and input-isolation details are provided in Appendix B.

(2) Metrics. We use four case-level metrics. **Executed Assembly-Graph Validity (EAGV)** combines instance coverage and connectivity of the execution-plan relation graph corresponding to the returned assembly, CAD execution evidence, and structural topology consistency. **Visual Assembly Consistency (VAC)** evaluates visual agreement between each method's final returned assembly and the reference image. Three VLMs assess each result independently, and the final decision is determined by majority vote. **Interference-Free Rate (IFR)** measures whether all pairwise solid overlaps satisfy the shared FreeCAD/OCCT interference criterion. For all methods, final IFR is

recomputed using the same FreeCAD/OCCT evaluator described in Appendix C.3. **End-to-End Assembly Success Rate (EASR)** requires all three checks together with the clean-input condition.

For case i, let e_i, v_i, f_i, and q_i denote the corresponding binary decisions. We compute

$$\mathrm{EASR} = \frac{100}{N} \sum_{i=1}^{N} \mathbf{1}[e_i \wedge v_i \wedge f_i \wedge q_i], \tag{1}$$

where $N = 310$. All metrics are reported as pass rates (%) over the complete case set. Detailed evaluation rules are provided in Appendix C.

(3) Comparison Methods. We compare SPARC with representative geometry-based and image-guided assembly methods, including JoinABLe (Willis et al., 2022) and Assembler (Zhao et al., 2025). JoinABLe predicts geometric joints between B-Rep part pairs and searches for their relative placements, whereas Assembler predicts the spatial arrangement of supplied parts from a reference image. All methods use the same input part sets and target assemblies whenever their input requirements permit. Because JoinABLe does not infer a global assembly sequence from the reference image, we provide the ground-truth part-pair sequence and report this setting as *JoinABLe (Oracle Seq.)*. Assembler receives the same reference image and input parts as SPARC. We evaluate it using VAC and IFR because it returns part placements rather than an executable B-Rep constraint program. The JoinABLe results should therefore be interpreted as an oracle-assisted reference rather than a strictly input-matched comparison.

(4) Implementation Details. Unless otherwise stated, SPARC uses Claude Opus 4.8 as the default VLM. The reference image and normalized multi-view renderings of the input parts are provided for structured proposal generation. We also evaluate Gemini 3.1 Flash Image to examine sensitivity to the VLM backbone. Geometric grounding and CAD execution operate on the supplied STEP/B-Rep parts. Rendering settings, geometric search parameters, assembly-level scoring, and recovery details are provided in Appendix A.

4.2 Experimental Results

(1) Main Results. We compare SPARC with existing image-guided and geometry-based assembly methods on the full 310-case SPARC-Bench. We report SPARC with Gemini 3.1 Flash Image and Claude Opus 4.8. Table 1 summarizes the main results.

Table 1: Comparison on SPARC-Bench.

Method	EAGV ↑	VAC ↑	IFR ↑	EASR ↑
Assembler	–	36.45	24.19	–
JoinABLe (Oracle Seq.)	83.23	17.10	10.32	5.16
SPARC (Gemini 3.1 Flash Image)	94.52	93.87	95.16	93.55
SPARC (Claude Opus 4.8)	**95.16**	**94.84**	**96.13**	**94.84**

SPARC improves both reference consistency and interference-free assembly, achieving higher VAC and IFR than both baselines. Its EASR reaches 94.84% with Claude Opus 4.8 and 93.55% with Gemini 3.1 Flash Image. The gap between graph validity and end-to-end assembly success is particularly pronounced for JoinABLe. Even with an oracle part-pair sequence, JoinABLe achieves 83.23% EAGV but only 5.16% EASR. Its lower VAC and IFR indicate that structural validity does not necessarily imply the intended spatial configuration or interference-free placement. Assembler likewise remains below SPARC on both shared metrics. Together, these results highlight the importance of resolving connections in the context of the target assembly and checking their geometric realization.

(2) Component Ablation. We evaluate three variants. *Geometry-Only* removes reference-image and VLM guidance while retaining geometry-based planning, assembly-level scoring, and CAD-feedback recovery. *w/o Assembly Sel.* reuses the same VLM proposal and initial plan H_0 as full

SPARC for each case. It disables selector-based plan replacement while retaining pre-execution preparation and CAD-feedback repair. *w/o CAD Repair* reuses the same VLM proposal as full SPARC for each case. Geometric grounding, assembly-level selection, and pre-execution preparation remain unchanged. The variant disables feedback-driven repair after the first main CAD execution. Detailed configurations are provided in Appendix D.2.

Visual guidance remains important even when geometric planning and CAD recovery are available: Geometry-Only achieves 20.00% EASR (Table 2). Disabling assembly-level plan replacement reduces EASR from 94.84% to 81.61%, supporting the benefit of comparing alternative relations and interface bindings before execution. CAD-feedback repair provides a further improvement after the plan has been prepared. With the same VLM proposal and pre-execution settings, enabling repair raises EASR from 87.10% to 94.84%. The paired results include 26 failure-to-success transitions and two success-to-failure transitions, yielding a net gain of 24 successful cases (Appendix D.3).

Table 2: Component ablation on SPARC-Bench. All metrics are pass rates (%).

Method	EAGV	VAC	IFR	EASR
Geometry-Only	22.26	20.97	32.26	20.00
w/o Assembly Sel.	82.26	81.61	83.23	81.61
w/o CAD Repair	91.29	90.00	92.26	87.10
SPARC	**95.16**	**94.84**	**96.13**	**94.84**

(3) Search and Repair Strategies. We compare full SPARC with Local Greedy and two Static Beam variants. The three baselines reuse the same VLM proposal for each case and keep its part-pair relations and relation types fixed. Local Greedy independently selects the highest-ranked interface match for each relation. Static Beam expands partial assemblies and ranks complete plans: the single variant considers only the top-ranked plan, whereas the multi variant considers up to three pre-ranked plans. All three baselines disable post-execution repair. SPARC additionally supports relation revision before execution and CAD-feedback repair afterward. This comparison evaluates complete configurations with different search and repair settings (Appendix D.2).

Exploring alternative plans improves over independent interface selection in these configurations (Table 3). Local Greedy achieves 15.81% EASR, whereas Static Beam reaches 40.65% when considering one complete plan and 71.61% when considering up to three. The full SPARC configuration reaches 94.84% EASR. Unlike the fixed-relation baselines, it can revise the proposed relations before execution and use CAD feedback for local repair. The comparison therefore shows the advantage of the full pipeline over committing to the initial relation structure and searching only its interface assignments.

Table 3: Comparison of search and repair strategies on SPARC-Bench. All metrics are pass rates (%).

Method	EAGV	VAC	IFR	EASR
Local Greedy	18.39	15.81	18.71	15.81
Static Beam (single)	41.61	40.65	46.13	40.65
Static Beam (multi)	72.58	71.61	76.45	71.61
SPARC	**95.16**	**94.84**	**96.13**	**94.84**

(4) Interface Ambiguity Analysis. We compare SPARC with the variant without assembly-level plan replacement to examine performance across different levels of local interface ambiguity.

For case i, we define the ambiguity score as $A_i = \frac{|\{e \in E_i : n_{ie} > 1\}|}{|E_i|}$. Here, E_i is the edge set of the recovered VLM proposal graph before assembly-level plan selection and pre-execution preparation. The count n_{ie} is the number of distinct physical interface pairings in $\widetilde{\mathcal{C}}_e$ before routine-specific ranking and candidate reduction. Fallback returns that fail the applicable local geometric checks are excluded from this count. Thus, A_i measures the fraction of relations with multiple locally compatible interface pairings. Cases are assigned to low-, medium-, and high-ambiguity groups using the fixed intervals $0 \le A_i \le 1/3$, $1/3 < A_i \le 2/3$, and $2/3 < A_i \le 1$, respectively. The benefit of assembly-level plan replacement grows with interface ambiguity. As shown in Table 4, the EASR gap widens from 7.77 percentage points in the low-ambiguity group to 19.23 points in the high-ambiguity group, where SPARC achieves 90.38% compared with 71.15% without plan replacement. This pattern supports the role of complete-plan comparison in resolving competing interface hypotheses.

Table 4: EASR (%) across interface-ambiguity groups.

Ambiguity	Cases	SPARC	w/o Sel.
Low	103	99.03	91.26
Medium	103	95.15	82.52
High	104	90.38	71.15

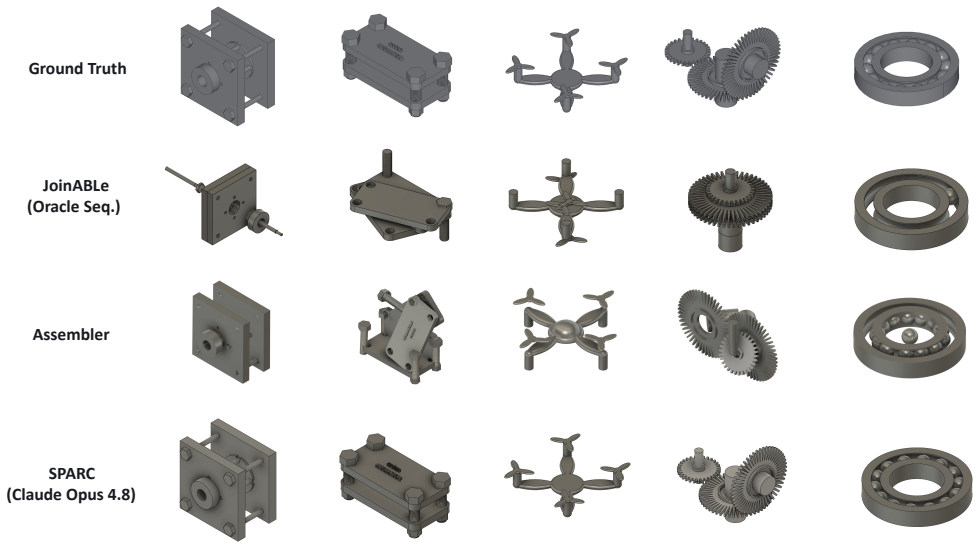


Figure 4: **Qualitative comparison on five assembly examples.** Each column shows one assembly. Rows show the ground truth, JoinABLe (Oracle Seq.), Assembler, and SPARC (Claude Opus 4.8), from top to bottom. JoinABLe receives the ground-truth base and ordered part-pair sequence, but no target interfaces or poses.

(5) Qualitative Comparison. Figure 4 compares SPARC with JoinABLe and Assembler on five assembly examples. SPARC more closely reproduces the reference configurations. In the second column, it preserves the layered plate arrangement and the placement of repeated fasteners, whereas the baseline outputs show displaced or tilted components. In the fourth column, the baseline gear arrangements differ from the reference, while SPARC better preserves the relative gear positions and orientations. These examples illustrate the difficulty of recovering the target configuration from locally plausible connections and placements, even when the part-pair sequence is provided. They complement the quantitative comparison by showing the resulting spatial errors.

5 Discussion and Limitations

The results on SPARC-Bench support treating interface grounding as an assembly-level decision rather than a collection of independent matches. SPARC connects visually proposed relations to geometric plan comparison and CAD execution while preserving the supplied part geometry.

The current system relies on predefined relation types and geometric routines. Occluded or symmetric interfaces may remain ambiguous from a single reference image. Candidate construction and pre-execution preparation can revise relations, whereas standard local repair adjusts interfaces and parameters without structural replanning. Our evaluation concerns the realization of the target configuration, not recovery of the source mating program or verification of articulated motion. Manufacturing feasibility and physical assemblability require additional checks, and VLM-based evaluation remains subject to bias and inconsistent judgments (Chen et al., 2024b). Future work will expand interface coverage and investigate broader feedback-driven structural replanning.

AI use statement. Generative AI was used solely for language editing and writing refinement during manuscript preparation.

Reproducibility statement. The appendix provides implementation details, benchmark construction procedures, baseline configurations, and evaluation protocols.

REFERENCES

Kamel Alrashedy, Pradyumna Tambwekar, Zulfiqar Haider Zaidi, Megan Langwasser, Wei Xu, and Matthew Gombolay. Generating CAD code with vision-language models for 3D designs. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=BLWaTeucYX.

Boyuan Chen, Zhuo Xu, Sean Kirmani, B. Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chen_SpatialVLM_Endowing_Vision-Language_Models_with_Spatial_Reasoning_Capabilities_CVPR_2024_paper.html.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. MLLM-as-a-judge: Assessing multimodal LLM-as-a-judge with vision-language benchmark. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6562–6595. PMLR, 2024b. URL https://proceedings.mlr.press/v235/chen24h.html.

Jang Hyun Cho, Boris Ivanovic, Yulong Cao, Edward Schmerling, Yue Wang, Xinshuo Weng, Boyi Li, Yurong You, Philipp Krähenbühl, Yan Wang, and Marco Pavone. Language-image models with 3D understanding. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yaQbTAD2JJ.

Ian Covert, Tony Sun, James Y. Zou, and Tatsunori Hashimoto. Locality alignment improves vision-language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=qssVptHTPN.

Yurui Dong, Shu Zou, Siqi Li, Nianchen Deng, Hongbin Zhou, Xuemeng Yang, Pinlong Cai, Licheng Wen, Xinyu Cai, and Botian Shi. ASSEMCAD: Production-ready CAD assembly generation from natural language. *arXiv preprint arXiv:2607.05123*, 2026. URL https://arxiv.org/abs/2607.05123.

Shengjie Gong, Wenjie Peng, Hongyuan Chen, Gangyu Zhang, Yunqing Hu, Huiyuan Zhang, Shuangping Huang, and Tianshui Chen. Learning hierarchical and geometry-aware graph representations for text-to-CAD. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=oKMomThD6n.

Qiangya Guo, Gang Dai, Zhuoman Liu, Shuangping Huang, Yunqing Hu, Huiyuan Zhang, and Tianshui Chen. Plan then Act: Bi-level CAD command sequence generation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LhE03r8X8z.

Anushrut Jignasu and Daniele Grandi. Linkify: Learning from interface-augmented assembly graphs. *Computer-Aided Design*, 199:104135, 2026. doi: 10.1016/j.cad.2026.104135.

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

Mohammad Sadil Khan, Sankalp Sinha, Talha Uddin Sheikh, Didier Stricker, Sk Aziz Ali, and Muhammad Zeshan Afzal. Text2CAD: Generating sequential CAD designs from beginner-to-expert level text prompts. In *Advances in Neural Information Processing Systems*, volume 37, pp. 7552–7579, 2024. doi: 10.52202/079017-0242.

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, 2020. doi: 10.1007/978-3-030-58539-6_40.

Fumin Liu, Haoyu Zhou, Fei Hao, and Lin Yang. Embodied CAD: Solver-grounded LLM agents for parametric B-Rep assembly modeling. *arXiv preprint arXiv:2606.31252*, 2026. URL https://arxiv.org/abs/2606.31252.

Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Jieneng Chen, Celso de Melo, and Alan Yuille. 3DSRBench: A comprehensive 3D spatial reasoning benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6924–6934, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Ma_3DSRBench_A_Comprehensive_3D_Spatial_Reasoning_Benchmark_ICCV_2025_paper.html.

Ari Seff, Wenda Zhou, Nick Richardson, and Ryan P. Adams. Vitruvion: A generative model of parametric CAD sketches. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Ow1C7s3UcY.

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

Mingjia Wang, Taiting Lu, Ziwei Dong, Sisong Bei, Jingying Zeng, Runze Liu, Kaiyuan Lin, Hongxing Pan, Kai Zhang, Yizheng Hou, Yangshoudu Zheng, Chenchen Guo, Weiyuan Meng, Shubin Lyu, Zhijun Zheng, Dexu Wang, Xinyu Bai, Shurui Qian, Zhangzixin, Mengyu Pan, Guoliang Shi, Ling Ma, Yifan Yang, Qi He, Yi-Chao Chen, Yincheng Jin, Sung-Liang Chen, and Mahanth Gowda. OmniCAD: A large-scale benchmark for 3D spatial reasoning in robotics assemblies. *arXiv preprint arXiv:2608.22637*, 2026. URL https://arxiv.org/abs/2608.22637.

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*, volume 39, pp. 7889–7897, 2025. doi: 10.1609/aaai.v39i8.32850.

Karl D. D. Willis, Pradeep Kumar Jayaraman, Hang Chu, Yunsheng Tian, Yifei Li, Daniele Grandi, Aditya Sanghi, Linh Tran, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. JoinABLe: Learning bottom-up assembly of parametric CAD joints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15849–15860, 2022.

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

Zhanwei Zhang, Shizhao Sun, Wenxiao Wang, Deng Cai, and Jiang Bian. FlexCAD: Unified and versatile controllable CAD generation with fine-tuned large language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Z0eiiV3Yyh.

Wang Zhao, Yan-Pei Cao, Jiale Xu, Yuejiang Dong, and Ying Shan. Assembler: Scalable 3D part assembly via anchor point diffusion. *arXiv preprint arXiv:2506.17074*, 2025. URL https://arxiv.org/abs/2506.17074.

Keyou Zheng, Xuyang Su, and Jiewu Leng. SPADA: A verifiable test-driven agent for controllable parametric CAD assembly generation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=hnV1ku7cd0.

Table 5: Main fields of the visual assembly proposal.

Field group	Description
Base and steps	Specifies the base instance and the ordered moving–anchor placement steps (*base*, *steps*).
Role and relation	Describes the functional role of a part and the proposed semantic assembly relation (*role*, *assembly type*).
Feature references	Identifies candidate geometric features on the moving and anchor parts, including members of repeated feature arrays.
Contact pairs	Specifies the primary contact and optional secondary locating interfaces, together with their normal-alignment conditions.
Spatial hints	Provides coarse insertion direction, axial seating, and relative layout cues for downstream geometric reasoning.
Structural conditions	Records cross-part structural relations when they are supported by the visual evidence.
Rationale	Provides a short explanation of the proposed assembly structure.

A Implementation Details

A.1 Planner Inputs and Output Fields

Inputs and part grouping. The planner receives the assembly reference image, representative part images, instance information, and geometric descriptions. Each input instance has a concrete ID, such as P0 or P1. Geometric descriptions include geometry classes, bounding-box dimensions, feature types, radii, and axis directions.

Parts are grouped by matching geometric summaries computed from the input geometry, rather than by reading source-assembly duplication labels or inferring repeated instances from the reference image. The summaries contain sorted bounding-box dimensions rounded to two decimal places, face counts by surface type, cylinder radii and inner/outer surface labels, and extracted feature types with radii and lengths rounded to two decimal places. Each group has one representative instance; non-repeated parts form singleton groups. Grouping reduces repeated prompt content without merging instances. The text input retains all instance IDs, group memberships, and instance counts.

Representative part images. Each part collage contains four fixed views with elevation–azimuth pairs $(25°, 45°)$, $(25°, -135°)$, $(90°, -90°)$, and $(0°, 0°)$. The default main planning call selects at most four image files from group representatives with available collages. Together with the reference image, the call therefore receives at most five image files. All image files preserve their aspect ratios and have a maximum long-edge resolution of 1280 pixels. Instances without an attached representative image remain listed in the text input.

Structured output. The prompt requests a JSON object containing a base instance, ordered placement steps, structural conditions, and a short rationale. Each step identifies a moving instance, an anchor, a relation type, and relevant feature references. The planner selects from the supplied feature labels and provides coarse spatial hints; it does not compute exact part transforms. Table 5 summarizes the output fields.

Inference settings. The main planner uses Claude Opus 4.8.

Parsing and validation. The parser attempts to remove code fences and trailing commas, extract the JSON object, close recoverable brackets, and recover complete step objects from truncated output. Recovered proposals must still pass the general-path instance checks. For n input instances, a complete general-path candidate must select a valid base, contain $n - 1$ placement steps, and use every non-base instance exactly once as moving. The base, moving instances, and all anchors must map to input instance IDs. Missing instances, unknown IDs, and repeated moving IDs cause rejection. Multiple steps may use the same anchor. The $n - 1$ requirement counts placement steps, not individual contact pairs. If parsing and validation do not yield a qualified plan, and no specialized structure-compilation branch recovers one, the main path makes one additional planning call with a compact prompt.

A.2 Geometry and Local Ranking

Geometric interface extraction. Planar features record face IDs, normals, areas, centroids, bounding boxes, and adjacency information. Default extraction retains planar faces covering at least 5% of the part's total surface area and selects representative faces with the same normal direction. Largest-plane selection and explicitly specified faces are processed separately. Cylindrical features distinguish inner hole surfaces from outer shaft surfaces and record radii, axes, positions, and estimated lengths. Lengths are estimated from bounding-box projections rather than measured insertion depths. Cylinders of the same type can form an array when their relative radius difference is at most 5%, the absolute dot product of their axis directions is at least 0.99, and at least two members are present. Each member retains its own face ID.

Shaft–hole matching. Feature pairing distinguishes shaft and hole surfaces and checks their radii using

$$-0.05\,\mathrm{mm} \le r_h - r_s \le 0.5 r_h, \tag{2}$$

where r_h and r_s are the hole and shaft radii.

Individual-hole ranking uses the clearance $c = r_h - r_s$ and length ratio $q_L = \min(L_s, L_h)/\max(L_s, L_h)$. At this stage, a candidate is marked feasible when $-0.5\,\mathrm{mm} \le c \le 5.0\,\mathrm{mm}$ and $q_L \ge 0.05$. If either length is missing or nonpositive, q_L is set to one. The ranking score is

$$S_{\mathrm{cyl}} = \frac{0.5}{1 + |c - 0.05|/1.0} + 0.5 q_L + \mathbf{1}_{\mathrm{hint}} \frac{0.15}{1 + d_{xy}/10}, \tag{3}$$

where c and d_{xy} are expressed in millimetres. Here, d_{xy} is the XY distance between the candidate position and the coarse position hint, and $\mathbf{1}_{\mathrm{hint}}$ indicates whether that hint is available. The feature-pair and instance-ranking tests are distinct stage-specific checks. When no candidate passes the instance-level checks, some matching routines return the first candidate with an ambiguity flag. Such fallback returns indicate unresolved matching attempts and are not counted as admissible bindings in the ambiguity analysis.

Planar seating. Without explicit interface references, candidate pairs are formed from the largest planar features on each part. Explicitly referenced faces are resolved directly. Pairs are ranked first by fewer inward-facing faces, with summed face area as the secondary criterion. There is no shared area-ratio threshold, and the initial normals of the independently positioned input parts need not be opposite. A secondary locating plane may be added when its normal $\mathbf{n}_{\mathrm{secondary}}$ satisfies $|\mathbf{n}_{\mathrm{primary}}^{\top}\mathbf{n}_{\mathrm{secondary}}| < 0.92$ relative to the primary plane of the same part. Without a secondary plane, some paths use a cylinder approximately perpendicular to the seating plane for auxiliary localization. After pose computation, a candidate must have a seating gap of at most $0.05\,\mathrm{mm}$ and a dimensionless directional residual of at most 0.02. Finite seating uses projected face bounding boxes for center or edge alignment. Some coplanar support-set paths additionally use a $0.5\,\mathrm{mm}$ containment tolerance.

Multi-pin and plate-on-pins matching. Matching requires an extracted hole array and expands the corresponding shaft instances. Radius pairing follows Equation 2. All declared holes must be matched by default. When partial matching is enabled, the default minimum is two pairs, unless another minimum is specified by the prompt. Array spacing and correspondence are evaluated through two-dimensional rigid registration of the axis origins in a cross-sectional plane. Correspondences are enumerated within the size limits in Table 6; larger problems use an initial-correspondence strategy. Pairing scores combine transverse registration RMS, rotation costs, unmatched-hole penalties, and position-hint penalties rather than radius agreement alone.

Gear placement. Local gear routines identify gear-like structures and estimate pitch radii from cylindrical arrays, bounding boxes, and related geometric features. Candidate placements are checked using axis alignment and center-distance conditions. For external meshing with parallel axes, the target center distance is $d = r_m + r_a$, where r_m and r_a are the estimated pitch radii of the moving and anchor gears. Internal meshing requires the anchor to be identified as an internal ring gear and $r_a > 1.35 r_m$. It uses $d = r_a - r_m$ and additionally requires $d > \max(1\,\mathrm{mm}, 0.2 r_m)$. Orthogonal lateral configurations require an orthogonal relation hint and a radius ratio in $[0.85, 1.15]$. For the external and internal parallel-axis configurations, the constructed pose is checked for an

Table 6: Candidate limits for local geometric matching.

Routine	Candidate limit
Planar pairing	Up to four planes per part and 16 face pairs when interfaces are unspecified. Returns the highest-ranked match.
Cylindrical ranking	Candidate counts vary across routines, without a shared top-K limit.
Multi-pin matching	Enumerates correspondences when both the pairing size and remaining anchor count are at most eight; otherwise uses the initial correspondence.
Local pattern	Evaluates at most the three highest-ranked pattern matches.
Gear placement	Three placement strategies, each returning at most one candidate pose.

Table 7: Finite search procedures for assembly candidates.

Family	State and extension	Pruning and completion
Mixed cylinders	Tracks placement steps, used interfaces, and partial assignments; extends with compatible interface pairs.	Prunes capacity violations and enumerates truncated candidate pools. The best complete assignment is checked for geometric error.
Coaxial chains	Tracks used parts, the last part, and occupied endpoints; extends with compatible unused parts.	Prevents endpoint reuse and retains a bounded number of paths per state. Full-coverage chains undergo geometric checks.
External gears	Tracks two-dimensional gear centers and direction costs; extends partial layouts with additional placements.	Checks pitch-circle distances and nonadjacent overlap, retaining a bounded number of states per layer. Complete layouts are rechecked; a search with no successor fails.

absolute axis-direction dot product of at least 0.98 and a radial center-distance error of at most $\max(1.5\,\mathrm{mm}, 0.04d)$. Local strategies are ranked using recognition confidence and validation errors and warnings.

Candidate limits. Table 6 summarizes the routine-specific limits. Local scores guide candidate generation; assembly-level evaluation follows Appendix A.3.

A.3 Assembly-Level Selection

Initial plan and candidate generation. The current plan is obtained from the VLM output through instance mapping, relation normalization, and interface resolution. Upstream recovery may request additional output, merge compatible partial plans from the same run, or apply an eligible geometric fallback. Alternative plans are generated by finite, family-specific searches. Table 7 summarizes their states, extensions, and checks. These searches are separate from the local ranking routines in Appendix A.2. Candidate constructors may change the base instance, placement sequence, part-pair relations, and relation types, in addition to interface bindings and parameters. For example, a constructor may replace a star-shaped connection structure with a serial chain. The selector compares the resulting plans and applies the conditional replacement rule; it is not restricted to rebinding interfaces on the original proposal graph.

Assembly-level scoring. We manually set the scoring weights to favor instance coverage and explicit geometric interface evidence, while penalizing underconstrained placements, interface conflicts, and structural inconsistencies. Let H denote an assembly plan, n the number of input in-

Table 8: Terms used in the assembly-level score.

Term	Definition
C	Fraction of expected non-base instances covered by the plan, counted by distinct instance IDs. The denominator is $n-1$, not the number of steps.
N_g	Number of steps with at least one resolved geometric primitive or a supported geometric certificate.
N_s	Number of steps with an evidence record but no such geometric evidence, including steps supported only by visual interface hints.
$N_r,\ N_t$	Numbers of steps whose specific role labels and specialized relation types satisfy the respective scoring rules.
N_w	Number of steps without any evidence record.
$U_m,\ U_p,\ U_a$	Numbers of multi-pin relations with insufficient anchors, lateral planar relations without locating evidence, and axial-only stacks without finite-seating evidence, respectively.
$K_{\mathrm{area}},\ K_{\mathrm{pos}},\ K_{\mathrm{cyl}}$	Numbers of conflict records from planar-area capacity, finite-position capacity, and cylindrical-interface capacity checks.
K_{topo}	Number of planetary-gear visual-topology records explicitly marked inconsistent.
B	Number of steps whose anchors include the base.
$N_{\mathrm{miss}},\ N_{\mathrm{dup}}$	Number of missing expected moving instances and number of repeated ordinary moving-instance IDs, respectively.
$N_{\mathrm{anchor}},\ N_{\mathrm{unreachable}}$	Number of distinct unknown anchor IDs and number of planned moving instances unreachable from the base through directed placement dependencies, respectively.

stances, and T the number of placement steps in H. The assembly-level score is

$$\begin{aligned} S_{\mathrm{asm}}(H) = {} & 120C + 10N_g + 2N_s + 5N_r + 4N_t - 4N_w \\ & - 80U_m - 18U_p - 30U_a - 100E_r \\ & - 120(K_{\mathrm{area}} + K_{\mathrm{pos}} + K_{\mathrm{cyl}}) - 250K_{\mathrm{topo}} \\ & - 3\max(B-1,0) - 70I_{\mathrm{star}} - 80I_{\mathrm{base}} \\ & - 45N_{\mathrm{miss}} - 60N_{\mathrm{dup}} - 55N_{\mathrm{anchor}} - 30N_{\mathrm{unreachable}} \\ & - 120I_{\mathrm{cycle}} - 250I_{\mathrm{rejected}}. \end{aligned} \tag{4}$$

All terms are evaluated on H. Table 8 defines the counts. The score evaluates the complete plan rather than summing local interface-ranking scores.

For each applicable cylindrical step, E_r takes the smallest normalized radius mismatch among its resolved shaft–hole pairs and sums these values across steps:

$$E_r = \sum_{s \in \mathcal{T}_{\mathrm{cyl}}} \min_{(r_s, r_h) \in \mathcal{R}_s} \frac{|r_h - r_s|}{\max(|r_h|, \epsilon)}, \qquad \epsilon = 10^{-6}\,\mathrm{mm}, \tag{5}$$

where $\mathcal{T}_{\mathrm{cyl}}$ contains the applicable steps and $\mathcal{R}_s$ contains their resolved radius pairs.

The indicator I_{star} equals one when $T \geq 5$, $B \geq \max(5, \lceil T/2 \rceil)$, and $(N_g + N_s)/T < 0.45$. The indicator I_{base} equals one when $n \geq 5$ and the base bounding-box volume is less than 15% of the largest input-part bounding-box volume. The indicators I_{cycle} and I_{rejected} mark a directed dependency cycle and a rejected plan, respectively. Each indicator is zero otherwise.

Validity filtering. An alternative is rejected if its base is invalid or used as a moving instance, an ordinary moving-instance ID is unknown or repeated, an anchor is unknown, or the plan contains a self-connection or dependency cycle. Strict coverage checking also rejects missing input instances. The three interface-capacity conflicts and the explicit planetary-gear topology conflicts defined above prevent adoption. Multi-pin relations with insufficient anchors are also rejected. Candidates with at least three steps and no geometric primitive evidence are rejected, subject to branch-specific exceptions, including bearing relations.

These checks define $V_{\mathrm{asm}}(H)$: a rejected plan has value zero. Conflict penalties remain in the score for diagnostic comparison, while the rejection flag determines eligibility. A favorable score cannot override a hard rejection.

Evidence filtering and candidate selection. Let H_0 be the current plan and $\widehat{\mathcal{H}}$ the generated alternatives. Each alternative is scored, and rejected alternatives are removed. If H_0 is not rejected, an alternative must also satisfy $N_g(H) \geq N_g(H_0)$. This evidence requirement is waived when H_0 is rejected. We denote this check by $Q_{\mathrm{ev}}(H, H_0)$ and form

$$\mathcal{H}_{\mathrm{eligible}} = \left\{ H \in \widehat{\mathcal{H}} \mid V_{\mathrm{asm}}(H) = 1,\ Q_{\mathrm{ev}}(H, H_0) = 1 \right\}. \tag{6}$$

If this set is empty, H_0 is retained. Otherwise, the highest-scoring eligible candidate is selected:

$$H_{\mathrm{best}} = \underset{H \in \mathcal{H}_{\mathrm{eligible}}}{\arg\max}\ S_{\mathrm{asm}}(H). \tag{7}$$

Score ties are resolved by the lexicographic order of candidate names. This maximization is over the generated eligible candidates, not all possible assembly plans.

Replacement margin. The replacement margin discourages plan changes based on small score differences. We manually set the default margin $\delta(H, H_0)$ to 35 and reduce it under the following predefined conditions. It is reduced to 5 when the current plan is rejected, the caller marks it suspicious, or $I_{\mathrm{star}}(H_0) = 1$. The margin becomes zero in either of two cases. First, the alternative has complete coverage, $N_g(H) \geq n - 1$, and $N_s(H) = 0$, while the current plan has $N_s(H_0) > 0$ and fewer geometrically supported steps. Second, the caller marks the current plan suspicious, the alternative does not decrease the score, coverage, or geometric-evidence step count, and $N_s(H) = 0$. The margin is capped at 5 for candidates containing bearing relations and for designated gantry candidates satisfying the complete-geometric-evidence conditions. An already zero margin remains zero.

Final replacement decision. The replacement margin is tested only after H_{best} has been selected:

$$H_{\mathrm{out}} = \begin{cases} H_{\mathrm{best}}, & S_{\mathrm{asm}}(H_{\mathrm{best}}) \geq S_{\mathrm{asm}}(H_0) + \delta(H_{\mathrm{best}}, H_0), \\ H_0, & \text{otherwise.} \end{cases} \tag{8}$$

If the highest-scoring eligible candidate fails its margin, the current plan is retained. Lower-scoring candidates are not reconsidered even if they have smaller margins. Thus, one selector call compares all alternatives against the same current plan rather than updating it after each candidate. Retaining a plan does not establish CAD feasibility; execution and final validation follow.

Parameter setting. The scoring weights and replacement margins are manually specified. Geometric rules, candidate construction, scoring parameters, and VLM prompts were not adjusted based on SPARC-Bench results. No case-specific manual adjustments were made during evaluation.

A.4 CAD Execution and Local Repair

Constraint execution. Geometric routines resolve feature labels to B-Rep faces, check radius compatibility, and replace candidate features when needed. They select auxiliary seating faces and generate joint specifications from the current execution state A.

Each selected face carries descriptors including surface type, area, orientation, position, and radius where applicable. After STEP import, the execution layer uses these descriptors to match the selected faces to FreeCAD `Face#` references. The matched references define the two joint connectors.

Table 9 summarizes the mappings for shaft–hole insertion and planar seating. Both `Distance` mappings use the default distance setting. FreeCAD fixes the anchor instance, creates the native joint objects and their connector frames from the resolved B-Rep references, and invokes the assembly solver to adjust part poses. If a required joint cannot be created or the solver does not return a successful solve, the CAD execution is treated as failed. The resulting assembly is then saved and evaluated for interference.

Table 9: Native joint mappings for shaft–hole insertion and planar seating.

Relation	Joint type	Geometric references
Shaft–hole insertion	`Cylindrical`	Selected shaft and hole cylindrical faces.
Axial seating, when applicable	`Distance`	Selected shaft-shoulder and hole-opening planar faces.
Planar seating	`Distance`	Selected planar face on each participating part.

Table 10: Example of native joint compilation from selected B-Rep references.

Selected B-Rep references	Joint	Connector 1	Connector 2
`shaft.Face4`, `sleeve.Face2`	Cylindrical (`Joint1_primary`)	`shaft`: Face4, Vertex1	`sleeve`: Face2, Vertex1
`shaft.Face3`, `sleeve.Face3`	Distance (`Joint2_auxiliary`), $0\,\mathrm{mm}$	`shaft`: Face3, Vertex1	`sleeve`: Face3, Vertex1

Example of native joint compilation. For a representative shaft–sleeve assembly, the selected B-Rep references are compiled into two native FreeCAD joints, as shown in Table 10. The primary cylindrical joint aligns the shaft and sleeve axes, while the auxiliary distance joint specifies axial seating.

FreeCAD reports the resulting assembly as `solved`. For this example, the final measured interference volume is $0\,\mathrm{mm}^3$. The solve status contributes to the CAD-execution component of EAGV, whereas interference is evaluated separately by IFR.

Pre-execution preparation. Before the first main CAD execution in an assembly run, geometry-based preparation may select a different base, reorder or group placement steps, reassign anchors, revise relation types, and update interface bindings and parameters. Applicable branches may expand a single-anchor placement into multi-pin support or replace a cylindrical relation with planar seating. Preparation also checks local geometric convergence, interface compatibility, direction, insertion, and seating conditions, and applies the configured corrections.

Execution diagnostics. After execution, SPARC inspects the run status, created connections, interfering part pairs, and interference volumes, together with the local geometric diagnostics. Repair usually targets the pair with the greatest interference and identifies the corresponding placement step. If no step can be localized this way, the procedure also checks planar steps with unresolved positioning or failed seating. When the only remaining issue requires reference-image confirmation, the outer loop stops rather than applying further geometric adjustments. Completion of the backend program alone does not establish assembly success.

Local repair. Repair operations depend on the relation type. Table 11 lists their order. Standard direct repair updates interface bindings and pose parameters and may repeat geometric solving for the affected steps. VLM diagnosis requests local actions, such as selecting another hole feature, changing direction, or revising planar contact pairs, rather than a new complete assembly plan. A correction may propagate to dependent descendants or be reused for equivalent repeated instances, so a local repair can affect multiple parts. Within the same assembly run, CAD re-execution uses the updated execution state without repeating the full pre-execution preparation.

Repair budgets. The default outer loop permits an initial execution and at most two retries. Some gear and specialized structural branches can extend this budget. The three-attempt limit applies only to the outer loop: local candidate tests can invoke additional CAD executions. Table 12 distinguishes these limits from infrastructure recovery.

Trial acceptance. For the general CAD trial branch, let V_0 and V' denote the interference volumes before and after a trial. The intermediate acceptance condition is

$$V' < 0.9V_0 \quad \text{or} \quad V' < 1.0\,\mathrm{mm}^3. \tag{9}$$

Table 11: Relation-specific repair sequences. Operations are attempted in the listed order when applicable.

Relation	Repair sequence
Planar seating	Opposing-axis candidates; annular end-face candidates; primary contact-face pairs; auxiliary locating-face pairs; center or edge alignment; VLM local diagnosis.
Cylindrical relations	Keyed or equivalent-instance orientations when applicable; bounded direction enumeration; axial-depth candidates; rotation about the axis; flipping; axial shifts; alternative hole openings; VLM local diagnosis.
Gear relations	Joint treatment of multiple colliding relations when needed, followed by bounded center-distance and phase adjustments. Gear repair uses a separate strategy from cylindrical repair.

Table 12: Limits at different levels of execution and repair.

Level	Limit or handling
Pre-execution correction	At most three internal attempts per invocation of the correction routine, not per assembly case.
Outer execution loop	One initial execution and at most two retries by default, with extensions in some specialized branches.
Local candidate testing	Branch-specific limits, including at most four axial candidates and three planar alignment modes by default. These tests may call the CAD backend.
Infrastructure recovery	On timeout, first check for a completion record. Reconnect and resend if needed. These retries are distinct from algorithmic repair.

Thus, a trial must reduce interference by more than 10% or fall below the branch's small-volume threshold. Additional checks depend on the branch. Cylindrical trials first verify that the revised interfaces still satisfy the geometric conditions. Planar trials must satisfy the applicable seating and execution conditions; some candidates awaiting visual confirmation may be retained as intermediate states. Direct gear trials may evaluate interference on the target connection pair while limiting the increase in total assembly interference to $\max(1.0\,\mathrm{mm}^3, 0.05V_{\mathrm{tot},0})$, where $V_{\mathrm{tot},0}$ is the pre-trial total interference volume. The general acceptance branch for VLM-proposed actions requires a reduction greater than 10%, without the small-volume alternative in Equation 9. These conditions govern intermediate repair decisions. The $1.0\,\mathrm{mm}^3$ threshold is not the final IFR threshold. Final assembly success requires the evaluation criteria in Appendix C.

A.5 Complete Assembly Procedure

Algorithm 1 summarizes the default SPARC procedure. Proposal parsing, recovery, and validation follow Appendix A.1. Candidate selection follows Appendix A.3, and execution and repair follow Appendix A.4. Local repair returns an updated state only when a trial satisfies the applicable acceptance rule. Local CAD probes and infrastructure recovery are separate from the outer execution limit. Final evaluation follows Appendix C.

B SPARC-Bench Construction and Input Isolation

B.1 Data Sources and Case Selection

SPARC-Bench contains 310 assembly cases constructed from two public CAD assembly datasets: 230 from AutoMate (Jones et al., 2021) and 80 from the Fusion 360 Gallery Assembly dataset (Willis et al., 2022). Each case is standardized into a single assembly reference image and a set of per-instance STEP part files. Case selection combines programmatic filtering and random sampling based on assembly type, part count, and reference-image criteria. We focus on mechanical assemblies with identifiable mating relationships, excluding configurations dominated by complex free-form surfaces without clear mechanical connections. Cases in which the part instances are at least partially visible and distinguishable in the reference image are prioritized. Corrupted files and

Algorithm 1 SPARC assembly and evaluation procedure

Require: Sanitized part instances $\mathcal{P}$, reference image I, and configuration Θ
Ensure: Assembly result R, evaluation decisions, and final status

Extract geometric summaries and group repeated instances
Prepare representative part views and textual planner inputs
Query the VLM with the reference image and prepared inputs
Parse, recover, and validate the VLM proposal
if no usable proposal is available **then**
    return failure with available diagnostics
end if
Map instances, normalize relations, and resolve interfaces to construct plan H
if plan construction fails **then**
    return failure with available diagnostics
end if
Generate finite family-specific alternatives $\mathcal{H}$
Compute S_{asm} for H and all alternatives
Remove alternatives failing V_{asm}
if H is not rejected **then**
    Remove alternatives with fewer geometrically supported steps than H
end if
if $\mathcal{H} \neq \varnothing$ **then**
    Select the highest-scoring alternative H_{best}, breaking ties by lexicographic candidate-name order
    Determine the replacement margin $\delta(H_{\mathrm{best}}, H)$
    if $S_{\mathrm{asm}}(H_{\mathrm{best}}) \geq S_{\mathrm{asm}}(H) + \delta(H_{\mathrm{best}}, H)$ **then**
        $H \leftarrow H_{\mathrm{best}}$
    end if
end if
Initialize execution state A from the retained plan H
Apply configured pre-execution preparation to A, including structural updates, interface resolution, and geometric corrections
Set the outer execution limit K_{out} from Θ
for $t = 1, \ldots, K_{\mathrm{out}}$ **do**
    $\Pi \leftarrow \textsc{Compile}(A)$
    $R \leftarrow \textsc{ExecuteFreeCAD}(\Pi)$
    Read execution status, created connections, and interference diagnostics
    if execution passes and no further recovery is requested **then**
        break
    end if
    if only reference confirmation remains, or $t = K_{\mathrm{out}}$ **then**
        break
    end if
    Identify the affected relations from the diagnostics
    $(A', a) \leftarrow \textsc{LocalRepair}(A, R, \Theta)$
    if $a = \mathrm{false}$ **then**
        break
    end if
    $A \leftarrow A'$
end for
Evaluate EAGV, VAC, and IFR on the same returned result R
Check the input-isolation condition q
Set final status to success only if $e \wedge v \wedge f \wedge q$
return R, (e, v, f, q), and final status

cases with missing parts are excluded. Source assembly identifiers are used for deduplication to prevent repeated selection of the same assembly record.

Development and evaluation. Cases used during development do not overlap with the 310 assemblies in SPARC-Bench.

B.2 Reference Generation and Input Preparation

Reference images. For Fusion cases, we use the reference images and STEP part files provided by the dataset. The reference images are not re-rendered. Each image is paired with the corresponding input part instances, which undergo the same coordinate sanitization and metadata isolation as the AutoMate cases. For AutoMate cases, reference images are generated by reconstructing the source assemblies from assembly metadata and the corresponding STEP geometry. Each part instance is resolved to its geometry, with repeated instances retained separately. We apply the source assembly transforms, orthogonalize their rotation matrices, and convert translation values from metres to millimetres. For rendering, the STEP geometry is triangulated and the vertices are transformed into the assembly configuration. The complete assembly is framed using its overall bounding box and rendered from configured viewpoints. Candidate views are assessed using foreground coverage, image-space extent, centering, and projection-based visibility diagnostics. An accepted view is retained as the case reference image. Source assembly transforms are used for reference generation, not as inference inputs.

Coordinate sanitization. STEP preprocessing uses CadQuery 2.8.0 with OCCT. Imported geometry is represented in millimetres, and the processed STEP files are exported in millimetres. Separating an assembly into individual STEP files alone does not ensure input isolation, because the exported geometry may still encode the original assembly placement in its coordinates. We therefore translate each part so that its axis-aligned bounding-box center lies at the origin, then apply rotations successively about the x, y, and z axes.

For a point $\mathbf{x}_i$ on part instance i, the sanitized coordinates are

$$\mathbf{x}_i^{\text{in}} = R_z(\theta_{i,z})R_y(\theta_{i,y})R_x(\theta_{i,x})\left(\mathbf{x}_i - \mathbf{c}_i\right),$$

where $\mathbf{c}_i$ is the bounding-box center before rotation and each angle satisfies $\theta_{i,k} \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$. The angles are assigned separately for each instance using a deterministic hash of the sanitization seed, case identifier, filename, and source-file checksum. This procedure preserves part geometry and physical dimensions while disrupting the original shared assembly placement. It is a discrete rotation procedure, not uniform sampling from $SO(3)$.

Metadata isolation and instance preservation. The planning pipeline receives the sanitized per-instance STEP files, one reference image, and part views generated from the sanitized geometry. Original mates, assembly hierarchy, source instance transforms, and source interface annotations are excluded from the inference inputs. Repeated geometry is preserved at the instance level rather than merged by shape. For example, four identical bolts remain four distinct input instances and are assigned separate instance IDs. This preserves the component multiplicity required by the target assembly.

B.3 Benchmark Statistics

SPARC-Bench contains 310 case records, each with one reference image and its corresponding STEP parts, for a total of 1,683 per-instance STEP files. The part count ranges from 2 to 33, with a median of 4 and a mean of 5.43. These counts refer to actual input instances rather than distinct geometric shapes; identical components are counted separately when they occur as separate assembly instances. Table 13 summarizes the recorded case families and assembly sizes. The collection primarily consists of small and medium-sized mechanical assemblies, with planar-seating and shaft–hole cases forming the largest subsets. Gear and bearing assemblies provide additional, less frequent configurations.

C Evaluation Protocol

C.1 Executed Assembly-Graph Validity

Executed Assembly-Graph Validity (EAGV) combines instance coverage, graph connectivity, CAD execution, and topology consistency. For case i, let $\mathcal{I}_i$ denote the set of input instance IDs and $\widehat{G}_i^u = (\widehat{V}_i, \widehat{E}_i^u)$ the undirected graph constructed from the part-pair relation records of the execution

Table 13: Composition of SPARC-Bench.

Recorded family	2 parts	3–7	8–12	13+	Total
Planar seating	13	125	24	6	168
Shaft–hole	11	89	8	12	120
Gear	6	10	2	2	20
Bearing	0	0	1	1	2
Total	30	224	35	21	310

plan corresponding to the final assembly returned by the system. The vertex set $\widehat{V}_i$ contains the distinct instance IDs reported for that plan. For SPARC, this graph need not coincide with the initial VLM proposal graph G_{prop}. It is constructed from the execution-plan records, not inferred from geometric contacts in the returned CAD model.

Instance coverage. Coverage is checked by instance identity rather than part count. The check passes only when $\widehat{V}_i = \mathcal{I}_i$: all input instances must be represented, and no unknown instance may appear. Repeated parts with identical geometry retain distinct IDs and are checked separately. Repeated occurrences of the same ID are counted once and cannot compensate for a missing instance.

Graph connectivity. The execution-plan relation graph $\widehat{G}_i^u$ must be connected. Together with instance coverage, this requires all input instances to belong to a single connected component of the reported graph. **CAD execution.** The execution check passes when the CAD backend successfully creates the required native joint objects from the selected B-Rep references and FreeCAD reports a successful solve for the returned assembly. Failure to create a required joint or failure of the CAD solve causes this check to fail. An execution pass does not establish final geometric correctness; solid interference is evaluated separately by IFR.

Topology consistency. The three evaluators specified in Appendix C.2 independently assess whether the returned assembly matches the reference connection topology. Each produces a topology judgment separate from its VAC judgment. This check passes when at least two evaluators return *match*; *indeterminate* does not count as a match. The assessment concerns missing or incorrect connections, whereas VAC also considers relative placement and orientation.

Let c_i, ℓ_i, d_i, and t_i denote the binary outcomes of the coverage, connectivity, CAD execution, and topology checks, respectively. A case passes EAGV only when all four checks pass:

$$e_i = \mathbf{1}[c_i \wedge \ell_i \wedge d_i \wedge t_i], \tag{10}$$

$$\text{EAGV} = \frac{100}{N} \sum_{i=1}^{N} e_i, \tag{11}$$

where $N = 310$ and the result is reported as a percentage.

C.2 Visual Assembly Consistency

Audit inputs and scope. Claude Opus 4.8, GPT-5.6, and Gemini 3.1 Flash Image independently evaluate each method's final returned assembly. For each case, all evaluators receive the same reference image, the same input-part collage, and five predefined views of the evaluated assembly. All output views use the same renderer, resolution, materials, background, lighting, and camera-framing rules.

The audit checks: (i) matching part count and presence; (ii) matching connections; (iii) matching relative positions, orientations, and ordering; (iv) no obvious interpenetration or detachment; and (v) agreement with the assembly intent shown in the reference.

Permitted differences. Differences in viewpoint, projection, color, material, lighting, and minor tessellation artifacts are allowed. The complete assembly may undergo a common translation or rotation. Geometrically identical parts may exchange instance IDs. Hidden reference details are not imposed as constraints, and a horizontal-to-vertical change in image space alone does not indicate an error. For components between repeated panels, global mirroring is allowed, but each component must remain in the same gap between panels.

Independent evaluation and aggregation. Each evaluator returns *match*, *mismatch*, or *indeterminate* based only on the shared visual evidence and the criteria above. Evaluators do not receive the other models' judgments.

A case passes VAC when at least two of the three evaluators return *match*; otherwise, it fails. An *indeterminate* judgment does not count as a match. Self-reported confidence does not affect the decision. CAD execution and numerical interference are evaluated separately by EAGV and IFR.

For case i, let $v_i = 1$ when the aggregated decision is a pass and $v_i = 0$ otherwise. We report

$$\mathrm{VAC} = \frac{100}{N} \sum_{i=1}^{N} v_i, \tag{12}$$

where $N = 310$. All cases remain in the denominator.

C.3 Interference-Free Rate

Shared evaluation setup. Final interference is evaluated for all methods using the same fixed FreeCAD/OCCT environment. Lengths are expressed in millimetres and volumes in cubic millimetres. The evaluator checks every unordered pair of distinct input instances in the returned assembly, including geometrically identical instances with different IDs. The checked pairs are not restricted to predicted connections or oracle part-pair sequences.

Pairwise overlap criterion. For instances a and b, let S_a and S_b denote their placed B-Rep solids, with volumes V_a and V_b. Their overlap volume is $v_{ab} = \mathrm{Vol}(S_a \cap S_b)$. A pair passes when

$$v_{ab} \le \tau_{ab}, \qquad \tau_{ab} = \max\!\left(10^{-6}\,\mathrm{mm}^3, 10^{-8}\min(V_a, V_b)\right). \tag{13}$$

Face, line, and point contacts with zero overlap volume are allowed. Positive overlap is accepted only within the pairwise tolerance. Shaft–hole, planar-seating, and gear relations receive no additional exemptions. A method's own expected-contact labels do not affect the evaluation.

Acceptance and aggregation. A case passes only when all input instances are present, all pairwise checks complete with valid volume results, and every pair satisfies Equation 13. Missing instances and failed, missing, or invalid checks cause the case to fail.

Let $f_i = 1$ when case i passes these conditions and $f_i = 0$ otherwise. We report

$$\mathrm{IFR} = \frac{100}{N} \sum_{i=1}^{N} f_i, \tag{14}$$

where $N = 310$ and IFR is expressed as a percentage.

Scope. The same criterion applies to intentional interference fits, deformable assemblies, and simplified gear geometry. Such configurations fail when their overlap exceeds the tolerance, even if the overlap is intentional. IFR therefore measures solid overlap in the represented geometry, not the physical feasibility of deformation or interference-fit assembly.

C.4 End-to-End Success and Result Aggregation

End-to-End Assembly Success Rate (EASR) requires the EAGV, VAC, and IFR checks to pass together with the clean-input condition. For case i, let e_i, v_i, and f_i denote the binary decisions defined in Sections C.1–C.3. The input indicator q_i equals one when the inference inputs satisfy the coordinate-sanitization and metadata-isolation requirements in Appendix B.2, and zero otherwise.

A case succeeds only when all four indicators equal one. We compute

$$\mathrm{EASR} = \frac{100}{N} \sum_{i=1}^{N} e_i v_i f_i q_i, \tag{15}$$

where $N = 310$ and EASR is reported as a percentage.

The EAGV, VAC, and IFR decisions for each case must refer to the same final assembly returned by the system. Passing checks from different execution attempts are not combined. Each case contributes one final result, regardless of the number of repair attempts or VLM judgments. For VAC, the final check is the majority-vote decision defined in Appendix C.2. A failed, missing, or indeterminate required final check makes the case unsuccessful. All cases remain in the denominator, including unsuccessful executions and incomplete evaluations.

D Additional Experimental Details and Results

D.1 Baseline Configurations

We use the official implementations and released weights of JoinABLe (Willis et al., 2022) and Assembler (Zhao et al., 2025), without training or fine-tuning. Additional adapters handle input conversion, instance tracking, multi-part composition, and STEP replay.

JoinABLe. We extend the official implementation with STEP input adaptation, sequential multi-part composition, CAD evaluation, and dependency-compatibility modifications. Candidate probabilities are averaged arithmetically across the five official checkpoints, last_run_0.ckpt through last_run_4.ckpt. The model receives B-Rep graphs extracted from STEP parts, including entity types, edge lengths, face and edge directions, and face–edge adjacency. Parts are independently centered and normalized. The reference image is not provided during prediction. The oracle input specifies only the base instance and an ordered sequence of moving–anchor pairs, without relation types, target interfaces, target poses, or offsets. The model predicts candidate face and edge entities.

The 20 highest-probability candidates from the ensemble are passed to the official simplex pose search. The recorded search configuration uses 4,096 samples, seed 42, and at most 100 optimization iterations per candidate. The selected face or edge references are used to align the corresponding geometric primitives and create native joints in the CAD backend. Failure to resolve a required reference, create a joint, or obtain a successful CAD solve causes the case to fail. The base remains fixed, and the remaining instances are placed in the supplied order. Each predicted relative transform is composed with the anchor's current transform to obtain the moving instance's global pose. Graph-construction or inference failure terminates the case. If a required pairwise pose search fails, the case does not proceed to complete CAD replay. Ground-truth poses are not substituted, and failed parts are not skipped to obtain a successful result. The resulting part placements are applied to the corresponding STEP instances to materialize the assembly for evaluation. The evaluation wrapper checks entity placement, rigid-transform validity, and required part-pair contact. Five fixed views are rendered for VAC assessment against the reference image. Because the base and pair sequence are provided, the graph structure used in EAGV is partly oracle-specified rather than fully inferred by JoinABLe.

Assembler. We use the official implementation with the released objaverse_500k.pt checkpoint from TencentARC/Assembler. Additional wrappers perform data conversion, record coordinate transforms, and replay predicted placements on STEP geometry. Each STEP instance is triangulated into OBJ using FreeCAD with a deflection of $0.1\,\mathrm{mm}$, then converted to GLB. Repeated parts remain separate instances. The official preprocessing samples 20,000 surface points with normals per part and selects 4,096 points. A total of 1,024 anchors is allocated across the assembly. The single reference image is preprocessed and passed to the visual encoder as an inference condition. The recorded configuration uses seed 2025, 250 diffusion steps, and classifier-free guidance scale 4.0. The model predicts anchors, and registration recovers the part transforms. Its outputs include an assembled GLB and an anchor point cloud. Saved instance correspondences and preprocessing transforms are used to map the predicted placements back to the original STEP coordinate system. The resulting rigid transforms are applied to the original B-Rep entities, and the assembly is exported as STEP.

Shared interference evaluation. Final assemblies from SPARC, JoinABLe, and Assembler are evaluated offline using the same fixed-version FreeCAD/OCCT evaluator described in Appendix C.3. Lengths are measured in millimetres and volumes in cubic millimetres. The evaluator checks every unordered pair of distinct instances, including repeated parts, rather than only predicted or oracle-connected pairs. All methods use the same pairwise overlap thresholds, contact

rules, and failure handling. No additional overlap exemption is granted based on a relation type or a method's expected-contact label. Method-specific collision criteria used during search or repair do not determine final IFR.

D.2 Ablation and Search Configurations

Component ablations.

Geometry-Only constructs assemblies from STEP geometry and extracted interface features, without reference-image input or VLM calls during planning, interface selection, or repair. Geometric compatibility rules generate part pairs and relation types through relation-specific candidates and a generic interface-graph branch. The generic branch constructs a maximum-weight spanning tree, selects the base using weighted connectivity and part volume, and traverses the tree to determine the placement order. Specialized candidates use their own construction order. Assembly-level scoring and validity filtering remain enabled, and the highest-scoring valid plan is selected. The variant retains geometric preparation, pose correction, and configured CAD-feedback recovery. Repair uses geometric rules and CAD diagnostics, without VLM-based diagnosis or visual guidance.

w/o Assembly Sel. reuses the same VLM proposal and initial plan H_0 as full SPARC for each case. The selector cannot replace H_0, which is passed directly to pre-execution preparation. Alternative plans may still be generated and scored for diagnostic logging, but cannot affect the retained plan.

Pre-execution preparation and CAD-feedback repair remain enabled with the same settings as full SPARC. Preparation may still revise the base, placement order, anchors, relation types, interface bindings, and parameters. This ablation removes selector-based plan replacement, not all structural changes to the initial proposal.

w/o CAD Repair reuses the same VLM proposal as full SPARC for each case. Geometric grounding, assembly-level conditional replacement, and pre-execution preparation remain unchanged. Pre-execution processing retains direction correction, axial insertion and seating adjustments, repeated-instance coordination, geometry-triggered local corrections, and execution-graph preparation. The variant disables feedback-driven repair after the first main CAD execution, with zero repair retries. Disabled operations include diagnostic-driven interface reselection, flipping, depth, offset and phase adjustments, VLM local diagnosis, repair probes, and corrected re-execution. Cases passing pre-execution checks undergo one main CAD execution and return its result. Cases failing these checks may terminate before execution. The returned status, diagnostics, and available outputs are preserved without automatically trying another assembly plan. Final evaluation follows Appendix C.

Search configurations. For each case, Local Greedy, Static Beam (single), and Static Beam (multi) reuse the same VLM proposal. The base instance, moving–anchor pairs, relation types, placement order, and direction hints are fixed across these configurations.

Local Greedy selects the highest-ranked interface match for each relation by local matching confidence. It retains single-relation pre-CAD processing, including planar-seating stabilization and axial-position selection, but disables CAD-feedback recovery and repair retries.

Static Beam extends partial assemblies step by step by enumerating compatible feature and instance matches and computing candidate placements. The beam width is eight, with at most 24 local candidates per incoming beam state for each relation. Ranking is lexicographic: fewer non-converged placements, fewer direction-side conflicts, and lower pre-execution axis-aligned bounding-box overlap take priority. Further criteria include cumulative matching confidence, geometric scores, and position and direction errors.

Static Beam (single) considers only the highest-ranked complete plan. If the plan passes pre-execution checks, it undergoes one main CAD execution without feedback repair. Failure does not trigger an attempt with another plan.

Static Beam (multi) freezes the complete hypotheses and their order before the first CAD execution. It considers at most the first three hypotheses. A hypothesis rejected by pre-execution checks still counts toward this limit. Each admitted hypothesis is executed without feedback repair. The method returns the first result passing the runtime CAD gate, which requires a passing system execution decision, successful backend execution, and satisfaction of the configured unexpected-interference

check. If none passes, it returns the last actually executed result and its status. If all considered hypotheses are rejected before execution, the case fails. The limit therefore applies to considered hypotheses, not to a guaranteed number of CAD executions.

Shared evaluation. All variants are evaluated using the protocol in Appendix C. EAGV, VAC, and IFR are computed for the same final assembly returned by each configuration. Final IFR is recomputed using the shared FreeCAD/OCCT evaluator, independently of the collision criteria used during candidate search or repair. Each benchmark case contributes one result. Failed executions and incomplete evaluations remain in the denominator. Passing checks from different candidates or repair attempts are not combined.

D.3 Paired Analysis of CAD Repair

We compare full SPARC with *w/o CAD Repair* on all 310 cases using the same VLM proposal and pre-execution settings for each case. Each configuration's final returned assembly is evaluated using the EASR criteria in Appendix C.4.

Table 14: Paired success outcomes with CAD repair disabled and enabled on SPARC-Bench. Success is determined by the EASR criteria.

Without repair	With repair	Cases
Failure	Success	26
Success	Failure	2
Success	Success	268
Failure	Failure	14
Total		310

Table 14 shows that enabling repair changes 26 cases from failure to success and two from success to failure. The net gain of 24 successful cases increases EASR from 87.10% (270/310) to 94.84% (294/310). Success or failure status remains unchanged in 282 cases.