RIGOR: CONSTRAINT-AWARE LAYOUT PLANNING AND GUI GROUNDING FOR INDUSTRIAL SCENES

Anonymous authors
Paper under double-blind review

ABSTRACT

Constructing an industrial scene through a graphical interface requires both a coherent spatial target and actions that realize its device parameters. Accurate interaction cannot compensate for an unsuitable layout, while a suitable layout can still be misapplied to the wrong interface field. We introduce RIGOR, a dataset and framework for studying these complementary requirements in static Unity workcells. The dataset links 1,050 rule-validated layouts from 291 task seeds to 222,306 GUI action units from 938 verified expert trajectories. The framework connects a supervised coordinate planner, relation-informed candidate generation, and bounded local search to a separately trained action model through explicit device-level targets. Refinement uses deterministic task feedback at inference time. Stage-wise evaluation separates layout quality from next-action correctness on frozen GUI states with shared targets. RIGOR achieves 87.04 in layout quality and 79.69 in relation satisfaction, compared with 54.14 and 29.17 for the best respective baselines, and 81.14% step success under the given-target protocol. Planning ablations improve layout quality from 25.85 with coordinate supervision to 87.04 with both refinement stages. These results support explicit constraint coordination and distinguish action recognition from parameter-grounded progress in GUI-based construction.

1 INTRODUCTION

Vision-language models increasingly operate software through screenshots and native interface actions (Hong et al., 2024; Lin et al., 2025; Qin et al., 2025). In scene-construction software, however, selecting the right action is only part of the problem: the resulting arrangement must satisfy the task's spatial requirements. An agent may faithfully enter every coordinate of a poor layout, or enter a correct coordinate into the wrong object's field. This distinction motivates our core question: *how should constrained spatial targets be prepared and connected to the GUI actions that realize them?*

We study the static scene-construction stage of industrial digital twins using kitting-to-assembly workcells in Unity Editor. Devices must be arranged under aisle, equipment-clearance, boundary, and parent-placement requirements. Several layouts may satisfy the same requirements, but realizing any one of them requires specific device–field–value bindings. For example, a click intended for Position X can instead select Rotation X, changing the meaning of the subsequent numerical input. The task therefore exposes two complementary requirements: **constraint-aware target quality** in world coordinates and **parameter fidelity** in interface coordinates. Planning admits multiple acceptable solutions; execution must remain faithful to the selected one.

Structured scene generation addresses spatial organization, and SceneGenAgent combines industrial layout planning, verification, and refinement through generated code (Feng et al., 2023; Yang et al., 2024; Xia et al., 2025). GUI agents study screenshot-grounded interaction, including point-precise geometric construction in PAGER (Qin et al., 2025; Wei et al., 2026). Our focus is the interface between these capabilities: linking a constrained industrial layout to the object identities, fields, and values that must be realized through a visible editor. Rather than treating a plausible action sequence as evidence of a suitable scene, we measure target quality and action fidelity separately.

To support this study, we introduce the RIGOR dataset, containing 1,050 rule-validated layouts from 291 task seeds and 222,306 recorded action units from 938 verified expert trajectories. Shared seed and layout identifiers connect requirements, device configurations, screenshots, and action histories.

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.

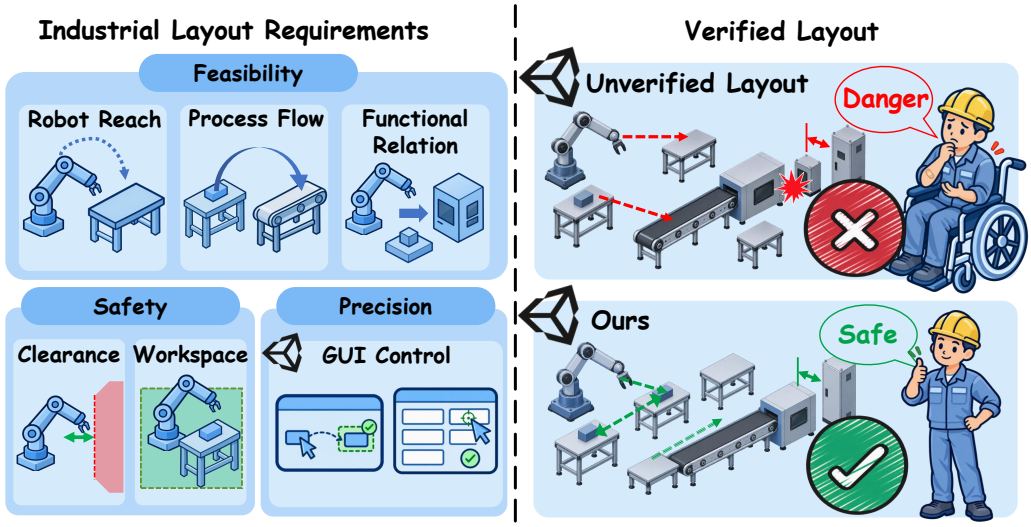

Figure 1: Industrial scene construction exposes a capability gap hidden by semantics-first GUI evaluation. Layouts must account for spatial, functional, and GUI-control constraints. Placement errors degrade these requirements, whereas RIGOR refines layout using task feedback before execution.

We also develop RIGOR, a framework in which a supervised planner proposes a layout; relation-informed candidate generation and bounded search refine it using deterministic task feedback; and a separate action model grounds the selected target into GUI operations. The structured layout is an explicit interface between the two learned stages, not an implicit plan embedded in an action history.

The experiments diagnose the two requirements under a controlled, stage-wise protocol. The strongest baseline by composite score reaches 94.54% action-type accuracy but only 40.07 in layout quality; the strongest baseline layout score, 54.14, coexists with 0.31% navigation success. For RIGOR, refinement raises layout quality from 25.85 to 87.04, while the action model reaches 81.14% step success on frozen states with shared targets. These measurements do not establish autonomous end-to-end construction, but they show why neither action recognition nor layout quality alone adequately characterizes the task.

Our contributions are threefold:

- We introduce the RIGOR dataset and a stage-specific evaluation protocol that support the study of constrained layout quality and parameter-grounded GUI action correctness.
- We propose a planning and action framework connected by explicit device, field, and value targets, combining supervised coordinate prediction, relation-informed refinement, bounded local search, and visual next action prediction.
- We conduct extensive baseline comparisons and planning ablations that reveal complementary gaps between layout planning and GUI grounding, quantify the contribution of each refinement stage, and identify remaining structural and interaction limitations.

2 Related Work

Structured and industrial scene generation. Text-to-scene methods use lexical grounding, layouts, and scene representations to connect language with spatial organization (Chang et al., 2015; Feng et al., 2023; Yang et al., 2024; Hu et al., 2024; Ling et al., 2026). Related systems generate interactive 3D worlds through programs, JSON, and constraint-expressive representations (De La Torre et al., 2024; Chen et al., 2025; Li et al., 2026b; Fan et al., 2026). SceneGenAgent is the most closely related: it generates industrial scenes through C# code, using structured planning, layout verification, and iterative refinement (Xia et al., 2025). RIGOR studies the additional connection between a spatial target and its realization through visible Unity GUI states, with device-level targets linking layout supervision to action supervision.

Visual GUI agents and precise interaction. GUI agents predict actions from screenshots, task context, and interaction histories, with progress in grounding, visual action modeling, and interface-oriented training data (Hong et al., 2024; Cheng et al., 2024; Gou et al., 2025; Wu et al., 2025; Xie et al., 2026; Lin et al., 2025; Xu et al., 2024; Qin et al., 2025). Ponder & Press, ShowUI-π, and BIM-gent further explore reasoning–execution organization and professional-software interaction (Wang et al., 2025; Hu et al., 2026; Deng et al., 2025). PAGER examines dependency-structured geometric drawing with point-level canvas operations (Wei et al., 2026). Our action space includes field clicks, focus transitions, exact parameter entry, and scene operations; the associated target is an industrial layout whose spatial quality is evaluated independently.

Process evidence and outcome evaluation. WebArena, OSWorld, AndroidWorld, and ScienceBoard evaluate agents in interactive environments with task-oriented completion criteria (Zhou et al., 2024; Xie et al., 2024; Rawles et al., 2025; Sun et al., 2026). AgentBoard, WindowsWorld, WorkArena++, and AgentTrek provide complementary settings for analyzing or acquiring multi-step interaction evidence (Ma et al., 2024; Li et al., 2026a; Boisvert et al., 2024; Xu et al., 2025). Professional-software benchmarks also study native artifacts and longer workflows (Ai et al., 2026; Man et al., 2026; Wang et al., 2026a). RIGOR contributes paired layout and GUI evidence, verifies expert trajectories against their saved scenes during dataset construction, and uses a stage-wise protocol to diagnose learned planning and next-action prediction.

3 METHOD

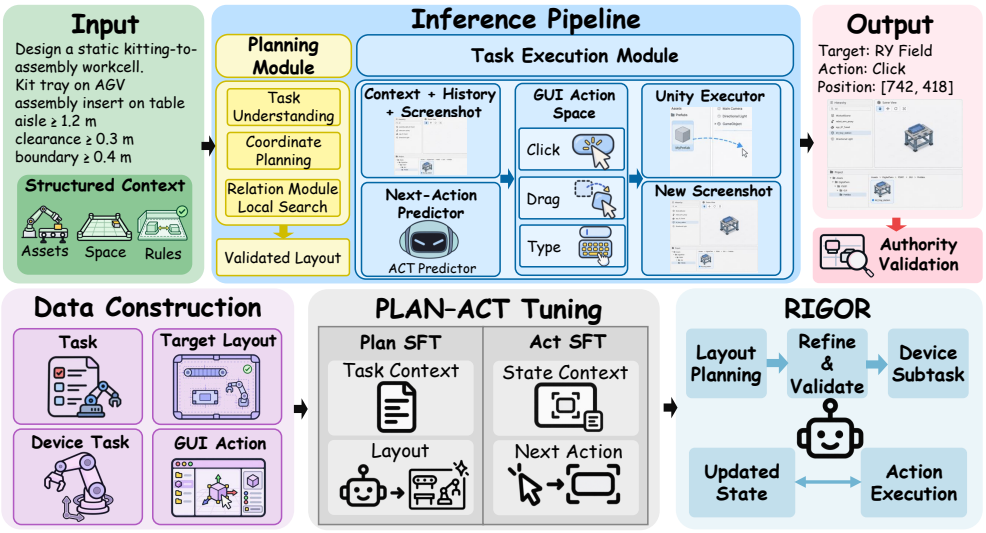

Figure 2: Overview of RIGOR. Planning predicts and refines structured layouts, execution grounds device targets into GUI actions, and training connects PLAN and ACT through layout supervision.

3.1 TASK AND STRUCTURED INTERFACE

We study industrial scene construction through GUI interaction. Given an input $X = (Q, E, W, \mathcal{R})$, where Q is the natural-language requirement, E the public asset information, W the workspace specification, and $\mathcal{R}$ the spatial and structural relations, the planner predicts a structured layout Y containing device identities, transforms, and specified parent relations.

The task requires both a feasible spatial target and faithful GUI realization. Let $\mathcal{H}_X$ denote the geometric and structural checks associated with X. We define

$$\mathcal{Y}_{\text{feas}}(X) = \left\{ Y \;\middle|\; \bigwedge_{h \in \mathcal{H}_X} h(Y) \right\}, \qquad \mathcal{T}_{\text{faith}}(Y) = \{\tau \in \mathcal{T}_{\text{GUI}} \mid \text{Scene}(\tau) \models Y\}. \tag{1}$$

Here, $\mathrm{Scene}(\tau) \models Y$ indicates that the scene produced by executing GUI trajectory τ realizes the device configuration specified by Y.

Multiple layouts may satisfy the same requirement, whereas GUI execution is grounded to the particular target selected for realization. Accordingly, the planner first predicts $Y^{(0)} = f_\theta(X)$, refinement selects $\widetilde{Y}$, and a deterministic compiler converts the selected layout into device-level execution targets for the action model.

3.2 Direct-Coordinate Planning

The planner directly predicts a structured layout $Y^{(0)} = f_\theta(Q, E, W, \mathcal{R})$. Supervised fine-tuning maps each task context to a retained reference layout, with eleven device records containing position, rotation, scale, and optional parent fields (Appendix D).

To evaluate spatial relations, each device's public local collision box is transformed into a world-space axis-aligned bounding box (AABB):

$$\mathbf{v}_{i,k}^{w} = R_y(\theta_i)\mathbf{c}_{i,k} + \mathbf{p}_i, \qquad \mathcal{B}_i(Y^{(0)}) = \left[\min_{k=1,\dots,8} \mathbf{v}_{i,k}^{w},\ \max_{k=1,\dots,8} \mathbf{v}_{i,k}^{w}\right], \tag{2}$$

where $\mathbf{c}_{i,k}$ is the k-th local box corner, $\mathbf{p}_i$ and θ_i are the predicted anchor position and yaw, and R_y denotes rotation about the vertical axis. The resulting AABBs support boundary, clearance, aisle, collision, and placement checks.

Direct coordinate supervision provides an initial task-conditioned layout, but matching a single reference does not explicitly enforce dependencies among device placements. Relation-informed refinement therefore evaluates alternative coordinate assignments against the shared task relations.

3.3 Relation-Informed Refinement

Given the initial prediction $Y^{(0)}$ and fixed task specification X, the refinement stages rank candidate layouts using the deterministic *Authority* score $S_{\mathrm{Auth}}(Y; X)$.

Coordinated candidate generation (R). R generates relation-informed alternatives from $\mathcal{R}$ and selects the highest-scoring layout together with the original proposal:

$$\mathcal{C}_R = \mathcal{G}_R\Big(Y^{(0)}, \mathcal{R}\Big), \qquad Y^{(R)} \in \operatorname*{arg\,max}_{Y \in \{Y^{(0)}\} \cup \mathcal{C}_R} S_{\mathrm{Auth}}(Y; X). \tag{3}$$

Bounded local search (L). Starting from $Y^{(R)}$, L performs bounded beam search over the neighborhood $\mathcal{N}(Y)$ with fixed beam width K. At each depth d, it forms $\mathcal{P}_{d+1} = \bigcup_{Y \in \mathcal{B}_d} \mathcal{N}(Y)$ and retains the highest-scoring candidates:

$$\mathcal{B}_{d+1} = \mathrm{TopK}_{K, S_{\mathrm{Auth}}(\cdot; X)}\left(\mathcal{P}_{d+1}\right), \qquad \widetilde{Y} \in \operatorname*{arg\,max}_{Y \in \bigcup_{d=0}^{D^\star} \mathcal{B}_d} S_{\mathrm{Auth}}(Y; X), \tag{4}$$

where $\mathcal{B}_0 = \{Y^{(R)}\}$ and $D^\star \le D$ is the last completed search depth.

R coordinates placements using explicit relations, while L performs local adjustment; both use deterministic task feedback at inference time without updating the learned planner.

3.4 Layout-Grounded GUI Action Prediction

The selected layout specifies what the scene should contain. A deterministic compiler converts it into execution targets. For a device-level target, the interface binds the device identity to its parameter fields and requested values:

$$\mathcal{T} = \Gamma(\widetilde{Y}) = (T_1, \dots, T_M), \qquad T_j^{\mathrm{device}} = \left(\mathrm{id}_j, \{(f, \nu_{j,f})\}_{f \in \mathcal{F}_j}\right). \tag{5}$$

where $\mathcal{F}_j$ is the set of fields for the device and $\nu_{j,f}$ is the requested value of field f.

For state t, let $o_t = (I_t, G_t, \widetilde{Y}, H_t, C_t)$ denote the action-model context, where I_t is the current screenshot, G_t the current device or scene objective, H_t the preceding recorded history, and C_t

the interface and device context. The model autoregressively generates a serialized action $z_t = (z_{t,1}, \ldots, z_{t,L_t})$, which is parsed under the action protocol $\mathcal{A}$:

$$p_\phi(z_t \mid o_t) = \prod_{\ell=1}^{L_t} p_\phi(z_{t,\ell} \mid o_t, z_{t,<\ell}), \qquad \widehat{a}_t = \mathrm{Parse}_{\mathcal{A}}(\widehat{z}_t). \tag{6}$$

where $\widehat{z}_t$ is the generated output sequence. PLAN and ACT are trained separately; the layout provides their shared semantic interface.

The action representation preserves native Unity interaction semantics. Mouse actions use integer coordinates in the original screenshot, text actions retain the requested string and target field, and keyboard transitions retain the key, transition count, source field, and destination field. For example, `TAB_FIELD_TRANSITION` explicitly represents a focus change between Transform fields. This separation matters because a correct numeric value advances the construction only when entered into its intended device field. Appendix D gives the complete prediction protocols.

4 Dataset

4.1 Construction and Validation

The dataset links a constrained layout problem to recorded actions that realize a reference scene (Figure 3). Its scope is a static kitting-to-assembly workcell with exactly eleven required devices. Condition tuples systematically vary the aisle width, equipment clearance, boundary clearance, and aisle orientation within the specified factory environment.

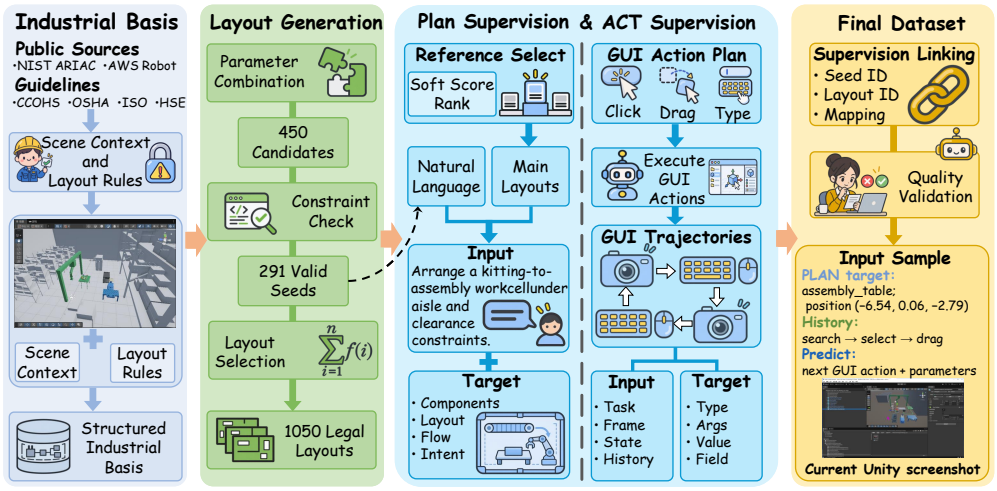

Figure 3: Dataset construction. Public scene information and task rules produce feasible layouts, PLAN supervision, and recorded Unity trajectories. Shared identifiers link the two supervision streams, while validation checks the encoded rules and recorded scene states.

Layout generation. Public asset resources and safety guidance inform the scenario specification. Deterministic enumeration produces 450 candidate tasks; hard-constraint screening retains 291 task seeds across 97 condition groups. Feasible candidates provide 1,050 layouts. For each condition tuple, a fixed search selects the first available $7.5\,\mathrm{m} \times 7.0\,\mathrm{m}$ work area and derives its reserved aisle from the conditions. Appendix C records the device inventory and screening template.

Planning supervision. One primary reference layout per seed supplies the PLAN target; other feasible layouts remain candidate references. The stored records contain `process_flow`, `scene_intent`, `component_config`, and `resolved_layout`. The fixed PLAN evaluation requests the `PLAN_RESOLVED` schema, with the full device layout as its output. This distinction separates the richer dataset record from the prediction contract used in evaluation.

Table 1: Key layout and process statistics of the RIGOR dataset.

Layout inventory	Count	Process inventory	Count
Condition groups	97	Verified expert trajectories	938
Task seeds	291	Raw action units	222,306
Validated layouts	1,050	Mean units per trajectory	237
Train / test layouts	938 / 112	Native TAB actions	61,908

GUI supervision. An automated expert maps layout targets to native mouse and keyboard operations in a visible Unity Editor. The recorder saves screenshots, states, actions, and preceding histories, producing next-action supervision. Seed IDs and Layout IDs connect these trajectories to their source layouts. Independent checks verify action–state continuity, screenshot correspondence, native action semantics, and consistency of the saved scene with its target. The retained non-test process corpus contains 938 verified expert trajectories.

4.2 Corpus Composition

Table 1 distinguishes task seeds, layouts, and recorded action units. The corpus contains 1,050 valid layouts derived from 291 task seeds across 97 condition groups, with a 938/112 Train–Test split. The 938/112 split refers to the layout inventory; multiple layouts can instantiate the same task seed, providing structured spatial variation under shared task conditions. The process corpus contains 222,306 raw action units from 938 verified non-Test trajectories, averaging 237 per trajectory, including 61,908 native TAB actions. Its navigation, parameter-input, and scene-interaction groups occupy 34.18%, 34.18%, and 31.65%, respectively, up to rounding (Appendix G), indicating a balanced coverage of navigation, parameter editing, and scene operations. These counts describe the stored supervision corpus; PLAN questions and ACT evaluation units are specified separately in the evaluation protocol.

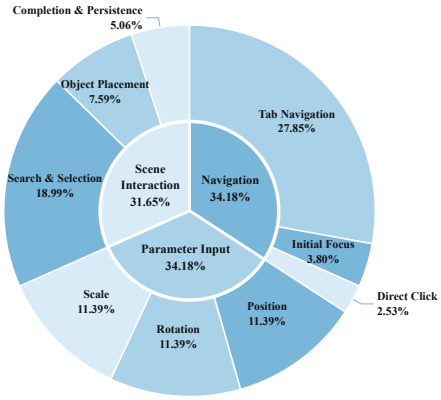

Figure 4: Dataset Analysis.

5 Experiments

5.1 Experimental Setup

Training and evaluated systems. Both PLAN and ACT start from Qwen3-VL-8B-Instruct (Bai et al., 2025). PLAN is fine-tuned on 261 seed-task instances from 87 condition groups for 3 epochs and 198 optimizer updates. ACT uses an independent LoRA adapter trained for 1 epoch and 1,200 optimizer updates, with seed 42, data seed 42, and data shuffling disabled. R and L add test-time candidate evaluation and search without further training.

We compare against seven open-weight VLMs, including Qwen3.5-9B (Qwen Team, 2026), Gemma-3-12B (Gemma Team et al., 2025), Ministral-3-8B (Liu et al., 2026), InternVL2.5-8B (Chen et al., 2024), Kimi-VL-A3B (Kimi Team et al., 2025), and MiniCPM-V-2.6 (Yao et al., 2024); three API-accessed VLMs, including DeepSeek-V4.1-Flash (DeepSeek-AI, 2026), GPT-5.5 (OpenAI, 2026), and Claude Sonnet 5 (Anthropic, 2026); and three GUI-specialized agents, including GUI-Owl (Ye et al., 2025), OpenCUA (Wang et al., 2026b), and UI-TARS (Qin et al., 2025) (Table 2). Experiments access DeepSeek-V4.1-Flash through its official API and GPT-5.5 and Claude Sonnet 5 through a third-party relay. The table retains model names recorded in those experiments.

Evaluation protocol. PLAN uses a fixed set of 24 questions, with malformed or non-scorable predictions contributing zero. ACT evaluates next-action predictions on frozen GUI states, real preceding histories, and shared targets produced by A+R+L. This given-target protocol isolates action

Table 2: Main results on RIGOR. Models are grouped into open-weight VLMs, API-accessed VLMs, specialized GUI agents, and our method. The best results are highlighted in blue, and the best baseline results are highlighted in pink.

Model	$S_{\text{total},24}$	Relation	Action	Step	Navigation	Input	Interaction	LLM-Judge	OCS
Open-weight VLMs									
Qwen3.5-9B (Qwen Team, 2026)	54.14	21.88	51.58	38.25	0.31	91.67	5.26	3.42	40.27
Gemma-3-12B (Gemma Team et al., 2025)	36.07	17.19	65.79	50.00	35.39	83.74	9.21	3.10	38.83
Ministral-3-8B (Liu et al., 2026)	0.00	0.00	28.54	10.62	9.36	16.26	1.32	1.81	5.71
InternVL2.5-8B (Chen et al., 2024)	0.00	0.00	10.17	0.00	0.00	0.00	0.00	1.33	0.51
Kimi-VL-A3B (Kimi Team et al., 2025)	0.00	0.00	5.00	0.38	0.00	0.00	1.97	0.50	0.50
MiniCPM-V-2.6 (Yao et al., 2024)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.46	0.00
API-accessed VLMs									
DeepSeek-V4.1-Flash (DeepSeek-AI, 2026)	40.07	29.17	94.54	70.00	63.48	99.59	21.49	4.35	52.08
GPT-5.5 (OpenAI, 2026)	26.71	22.92	95.67	69.17	67.90	**100.00**	27.19	4.31	47.45
Claude Sonnet 5 (Anthropic, 2026)	27.87	27.60	95.50	73.17	68.31	98.77	28.95	3.81	49.25
Specialized GUI Agents									
GUI-Owl (Ye et al., 2025)	11.57	25.00	62.08	50.71	42.39	82.82	0.00	2.54	31.03
OpenCUA (Wang et al., 2026b)	20.24	20.31	11.25	0.00	0.21	0.00	0.00	1.60	10.72
UI-TARS (Qin et al., 2025)	6.60	9.38	11.12	8.08	5.25	15.12	0.00	1.15	7.52
RIGOR	**87.04**	**79.69**	**100.00**	**81.14**	**89.99**	97.94	**44.74**	**4.39**	**82.85**

prediction from the quality of each model's own layout. All models are evaluated on the same fixed ACT set, with identical frozen GUI states, preceding histories, and A+R+L targets. Each metric therefore uses the same predefined denominator across models.

Layout quality $S_{\text{total},24}$ aggregates process and service-layout scores, while Relation Satisfaction counts satisfied functional relations. Action Accuracy measures the selected action type; Step Success additionally requires valid parameters and effective target progress. Navigation, Input, and Interaction report operation-specific performance. OCS combines the objective PLAN and ACT metrics; it is a stage-wise composite, with a separate 0–5 LLM-Judge score (Appendix F).

5.2 Evaluation Metrics

We evaluate industrial scene construction across four aspects: (1) layout planning, (2) action execution, (3) subjective quality, and (4) overall performance. Objective metrics are computed by deterministic evaluators, while subjective evaluation uses fixed prompts and scoring rubrics.

(1) Layout planning. We report layout quality $S_{\text{total},24}$ and Relation Satisfaction, measuring process and service-layout quality and the satisfaction of predefined functional relations, respectively. Scoring details are provided in Appendix F.1.

(2) Action execution. Action Accuracy and Step Success measure action-type correctness and complete action correctness, respectively. Navigation, Input, and Interaction further assess field navigation, parameter entry, and scene interaction. The decision criteria are detailed in Appendix F.2.

(3) Subjective quality. LLM-Judge assesses overall PLAN and ACT quality on a 0–5 scale and equally averages the two stage-level scores. The evaluation prompts are provided in Appendix E, and aggregation and rater-agreement measures are defined in Appendix F.3.

(4) Overall performance. The Overall Composite Score (OCS) aggregates the objective PLAN and ACT metrics separately and then equally averages the two stage-level scores. OCS is reported on a 0–100 scale and excludes LLM-Judge. The formulas and weights are provided in Appendix F.4.

5.3 Main Results

Table 2 reports the complete comparison. RIGOR achieves the highest reported OCS, 82.85, exceeding DeepSeek-V4.1-Flash by 30.77 points, or 59.08% relative to its 52.08 composite score. Its layout quality is 87.04 and Relation Satisfaction is 79.69. On the given-target action evaluation, it reaches 100.00% Action Accuracy and 81.14% Step Success.

Action recognition leaves substantial execution error. The API-accessed baselines reach 94.54–95.67% Action Accuracy, yet their actual Step Success ranges from only 69.17% to 73.17%. This separation clearly identifies the critical importance of action parameters, field grounding, and protocol validity beyond choosing an operation category. RIGOR improves Navigation Success to 89.99% and Interaction Success to 44.74%, although scene interaction remains its lowest main ACT subscore. GPT-5.5 leads Input Success at 100.00%, compared with 97.94% for RIGOR, showing that the gains are not uniform across action groups.

Planning and execution expose different limitations. Qwen3.5-9B has the strongest baseline layout score, 54.14, but its Navigation Success is 0.31%. DeepSeek-V4.1-Flash is the strongest baseline by OCS, with 40.07 in layout quality and 29.17 in Relation Satisfaction despite substantially stronger action recognition. GUI-Owl reaches 50.71% Step Success but scores 11.57 in layout quality, while OpenCUA and UI-TARS remain lower in the composite evaluation. These patterns motivate a shared benchmark with separate planning and execution measurements.

5.4 Planning Ablations

Table 3 evaluates different PLAN configurations. The Base model achieves only 17.25 in $S_{\text{total},24}$ and 7.81 in Relation Satisfaction, while SFT improves them to 25.85 and 26.56, respectively, showing that coordinate supervision provides a stronger initialization. Adding R or L further increases $S_{\text{total},24}$ to 67.24 and 65.98, respectively. Combining both refinements, RIGOR achieves the best performance with 87.04 in $S_{\text{total},24}$ and 79.69 in Relation Satisfaction, demonstrating their complementary benefits for constraint-aware planning.

Table 3: Ablation study of different PLAN configurations. The best results are highlighted in bold.

Method	$S_{\text{total},24}$ ↑	Relation ↑	PLAN Score ↑	LLM-Judge ↑
Base	17.25	7.81	14.42	1.26
SFT	25.85	26.56	26.07	3.21
A+R	67.24	40.63	59.26	2.81
A+L	65.98	39.06	57.91	3.10
RIGOR	**87.04**	**79.69**	**84.83**	**4.39**

5.5 Field-Level Error Analysis

Figure 5 illustrates a concrete execution failure for the Position X field of `metal_office_desk_01`. The valid field occupies $x \in [1692, 1749]$, $y \in [249, 262]$. RIGOR predicts $[1721, 256]$, inside this region; GPT-5.5 predicts $[1721, 277]$, inside Rotation X; Claude Sonnet 5 predicts $[1660, 256]$, outside the input fields. The example shows how identifying the relevant panel can still leave the intended parameter unselected. Appendices A and B provide additional capability diagnostics.

5.6 Consistency with Human Judgments

Figure 6 shows a clear overall alignment between LLMJudge and Human Ratings evaluations. Lower-performing methods cluster toward the lower-left region, while stronger models move consistently toward the upper-right. RIGOR occupies the top-right region, achieving high scores under both evaluation protocols. The near-perfect correlation ($r = 0.9773$) further demonstrates strong agreement between LLM-Judge and Human Rating judgments. This consistency indicates that our visual evaluation closely reflects independently assessed scene quality, supporting the reliability of the evaluation protocol.

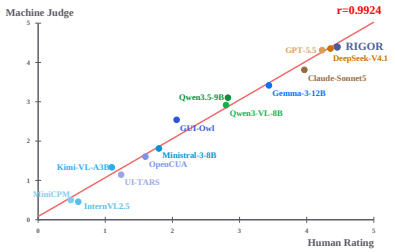

Figure 6: Automatic vs. human ratings.

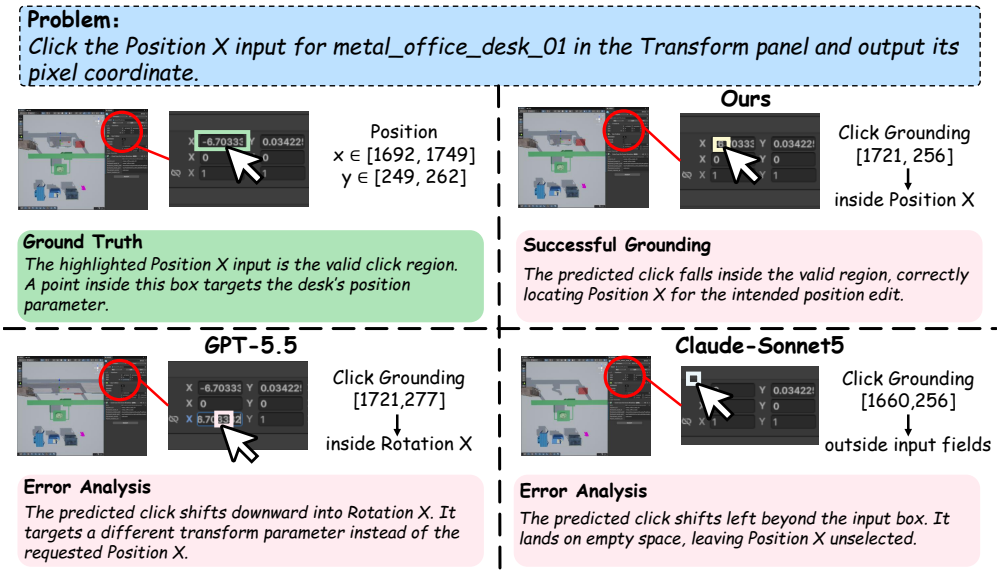

Figure 5: Qualitative comparison. RIGOR correctly grounds the Position X field, while the baselines select an incorrect field or miss the input region.

6 Conclusion and Scope

RIGOR studies industrial GUI scene construction as two connected problems: selecting a high-quality layout under spatial constraints and faithfully grounding its parameters into the intended interface fields. The dataset connects rule-validated layouts with verified expert interactions, while the framework makes the planning and execution interface explicit through device, field, and value targets. Stage-specific experiments show that strong action type recognition can coexist with poor layouts or incorrect parameter realization, and that relation-informed refinement substantially improves supervised coordinate proposals. The evidence supports improved layout planning and given-target next-action prediction, but does not establish autonomous scene completion or industrial safety. Future work should evaluate hard structural compliance and closed-loop construction in which execution outcomes are fed back into subsequent planning and interaction. Reliable industrial scene construction ultimately requires both an appropriate target and faithful realization of that target.

AI Use Statement

Generative AI tools were used solely during dataset construction to assist with the generation and organization of structured task instances. The authors designed the benchmark, dataset construction procedures, evaluation criteria, and experimental protocols, and independently verified all generated data and reported results.

Reproducibility Statement

We provide detailed descriptions of the benchmark construction, dataset validation procedures, PLAN and ACT prediction protocols, and evaluation metrics. The appendices include the industrial task template, prompt specifications, validation criteria, and metric definitions. The experimental section reports the training schedule and evaluation setup, while the ablation study provides detailed comparisons of different planning configurations.

References

Jiaxin Ai, Yukang Feng, Fanrui Zhang, Jianwen Sun, Zizhen Li, Chuanhao Li, Yifan Chang, Wenxiao Wu, Ruoxi Wang, Mingliang Zhai, et al. Prosoftarena: Benchmarking hierarchical capabilities

of multi-modal agents in professional software environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34586–34595, 2026.

Anthropic. Introducing Claude Sonnet 5. Anthropic, June 2026. URL https://www.anthropic.com/news/claude-sonnet-5.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Léo Boisvert, Megh Thakkar, Maxime Gasse, Massimo Caccia, Thibault L De Chezelles, Quentin Cappart, Nicolas Chapados, Alexandre Lacoste, and Alexandre Drouin. Workarena++: Towards compositional planning and reasoning-based common knowledge work tasks. *Advances in Neural Information Processing Systems*, 37:5996–6051, 2024.

Angel Chang, Will Monroe, Manolis Savva, Christopher Potts, and Christopher D Manning. Text to 3d scene generation with rich lexical grounding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 53–62, 2015.

Jiangong Chen, Xiaoyi Wu, Tian Lan, and Bin Li. Llmer: Crafting interactive extended reality worlds with json data generated by large language models. *IEEE transactions on visualization and computer graphics*, 31(5):2715–2724, 2025.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9313–9332, 2024.

Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. Llmr: Real-time prompting of interactive worlds using large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pp. 1–22, 2024.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression, 2026.

Zihan Deng, Changyu Du, Stavros Nousias, and André Borrmann. Bimgent: Towards autonomous building modeling via computer-use agents. *arXiv preprint arXiv:2506.07217*, 2025.

Tsz Hei Fan, Choi Wing Fung, Yuxuan Wan, Shuqing Li, and Michael R Lyu. Magic: Transition-aware generation of navigable multi-scene game worlds with large language models. *arXiv preprint arXiv:2607.11594*, 2026.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in neural information processing systems*, 36: 18225–18250, 2023.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Boyu Gou, Demi Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for gui agents. In *International Conference on Learning Representations*, volume 2025, pp. 30851–30883, 2025.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14281–14290. IEEE, 2024.

Siyuan Hu, Kevin Qinghong Lin, and Mike Zheng Shou. Showui-p: Flow-based generative models as gui dexterous hands. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8130–8140, 2026.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. Scenecraft: An llm agent for synthesizing 3d scenes as blender code. In *ICML*, pp. 19252–19282, 2024.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, et al. Kimi-VL technical report. *arXiv preprint arXiv:2504.07491*, 2025.

Jinchao Li, Yunxin Li, Chenrui Zhao, Zhenran Xu, Baotian Hu, and Min Zhang. Windowsworld: A process-centric benchmark of autonomous gui agents in professional cross-application environments. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15262–15280, 2026a.

Shuqing Li, Anson Y Lam, Yun Peng, Wenxuan Wang, and Michael R Lyu. 3d software synthesis driven by constraint-expressive intermediate representation. In *Proceedings of the 2026 IEEE/ACM 48th International Conference on Software Engineering*, pp. 1354–1366, 2026b.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19498–19508. IEEE, 2025.

Lu Ling, Chen-Hsuan Lin, Tsung-Yi Lin, Yifan Ding, Yu Zeng, Yichen Sheng, Yunhao Ge, Ming-Yu Liu, Aniket Bera, and Max Li. Scenethesis: A language and vision agentic framework for 3d scene generation. In *International Conference on Learning Representations*, volume 2026, pp. 136596–136629, 2026.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in neural information processing systems*, 37:74325–74362, 2024.

King Yiu Brandon Man, Ghadi Nehme, Md Ferdous Alam, and Faez Ahmed. Videocad: A dataset and model for learning long-horizon 3d cad ui interactions from video. *Advances in Neural Information Processing Systems*, 38, 2026.

OpenAI. Introducing GPT-5.5. OpenAI, April 2026. URL https://openai.com/index/introducing-gpt-5-5/.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Chris Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, volume 2025, pp. 406–441, 2025.

Qiushi Sun, Zhoumianze Liu, Chang Ma, Zichen Ding, Fangzhi Xu, Zhangyue Yin, Haiteng Zhao, Zhenyu Wu, Kanzhi Cheng, Zhaoyang Liu, et al. Scienceboard: Evaluating multimodal autonomous agents in realistic scientific workflows. In *International Conference on Learning Representations*, volume 2026, pp. 75694–75731, 2026.

Wenkai Wang, Tao Xiong, Jingchen Ni, Yunpeng Bao, Xiyun Li, Tianqi Liu, Hongcan Guo, Zilong Huang, and Shengyu Zhang. Deskcraft: Benchmarking desktop agents on professional workflows and human-in-the-loop collaboration. *arXiv preprint arXiv:2606.03103*, 2026a.

Xinyuan Wang, Bowen Wang, Dunjie Lu, Junlin Yang, Tianbao Xie, Junli Wang, Jiaqi Deng, Xiaole Guo, Yiheng Xu, Chen Wu, et al. Opencua: Open foundations for computer-use agents. *Advances in Neural Information Processing Systems*, 38:139756–139806, 2026b.

Yiqin Wang, Haoji Zhang, Jingqi Tian, and Yansong Tang. Ponder & press: Advancing visual gui agent towards general computer control. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 1461–1473, 2025.

Jingxuan Wei, Xi Bai, Shan Liu, Caijun Jia, Zheng Sun, Xinglong Xu, Siyuan Li, Linzhuang Sun, Bihui Yu, Conghui He, et al. Pager: Bridging the semantic-execution gap in point-precise geometric gui control. *arXiv preprint arXiv:2605.15963*, 2026.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: Foundation action model for generalist gui agents. In *International Conference on Learning Representations*, volume 2025, pp. 5090–5108, 2025.

Xiao Xia, Dan Zhang, Zibo Liao, Zhenyu Hou, Tianrui Sun, Jing Li, Ling Fu, and Yuxiao Dong. Scenegenagent: Precise industrial scene generation with coding agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17847–17875, 2025.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, et al. Scaling computer-use grounding via user interface decomposition and synthesis. *Advances in Neural Information Processing Systems*, 38, 2026.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. Aguvis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454*, 2024.

Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. Agenttrek: Agent trajectory synthesis via guiding replay with web tutorials. In *International Conference on Learning Representations*, volume 2025, pp. 79822–79843, 2025.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16277–16287. IEEE, 2024.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. MiniCPM-V: A GPT-4V level MLLM on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

Jiabo Ye, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Zhaoqing Zhu, Ziwei Zheng, Feiyu Gao, Junjie Cao, Zhengxi Lu, Jitong Liao, Qi Zheng, Fei Huang, Jingren Zhou, and Ming Yan. Mobile-agent-v3: Foundamental agents for gui automation. *arXiv preprint arXiv:2508.15144*, 2025.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

A Planning and Execution Capability Profiles

Figure 7 summarizes the reported planning and action profiles. The main table provides the direct quantitative basis for the comparison: DeepSeek-V4.1-Flash and GPT-5.5 exceed 94% in action-type accuracy while scoring 40.07 and 26.71 in layout quality, respectively. Qwen3.5-9B shows the complementary pattern of relatively strong layout quality and low navigation success. These profiles describe differences across evaluated capabilities; they do not isolate a temporal error-propagation mechanism.

Using the rounded component values in Table 2, Eqs. 20 and 21 yield approximate PLAN and ACT aggregates of 84.84 and 80.88 for RIGOR, and 36.80 and 67.37 for DeepSeek-V4.1-Flash. The reported OCS values, 82.85 and 52.08, are retained; recomputation from already rounded components can differ in the final decimal place.

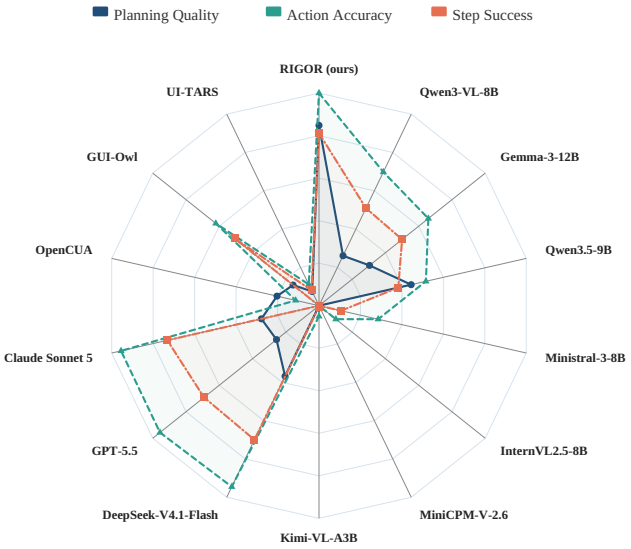

Figure 7: Planning and action capability profiles from the reported experiment. Planning Quality is defined by Eq. 20; Action Accuracy and Step Success follow their separate definitions.

B Fine-Grained Skill Decomposition

Figure 8 presents ten diagnostic dimensions: structural completeness, spatial feasibility, functional relations, action selection, field grounding, sequential navigation, position input, asset search and query, asset selection, and scene construction and persistence. Their definitions are given in Appendix F; the diagnostic action scores combine type selection and step realization and must be distinguished from the main-table success rates.

The planning diagnostics show a mixed profile. RIGOR scores 86.4 in the diagnostic labeled Spatial Feasibility and 79.7 in Functional Relations, while its Structural Completeness is 43.8, below several baselines. Spatial Feasibility is the normalized service-layout soft score specified in Eq. 25. This profile supports stronger service placement and functional relations in the reported outputs, with structural completeness remaining an unresolved weakness.

The action diagnostics likewise identify both gains and limits. RIGOR reaches 99.9 in Sequential Navigation and 90.0 in Asset Search & Query. Its Field Grounding score is 40.0, tied with several models, and its Scene Construction & Persistence score is 50.9, tied with GPT-5.5 and Claude Sonnet 5. The disaggregated results therefore locate the clearest improvements in selected relations and interaction skills, while preserving the remaining field-grounding and scene-level difficulties.

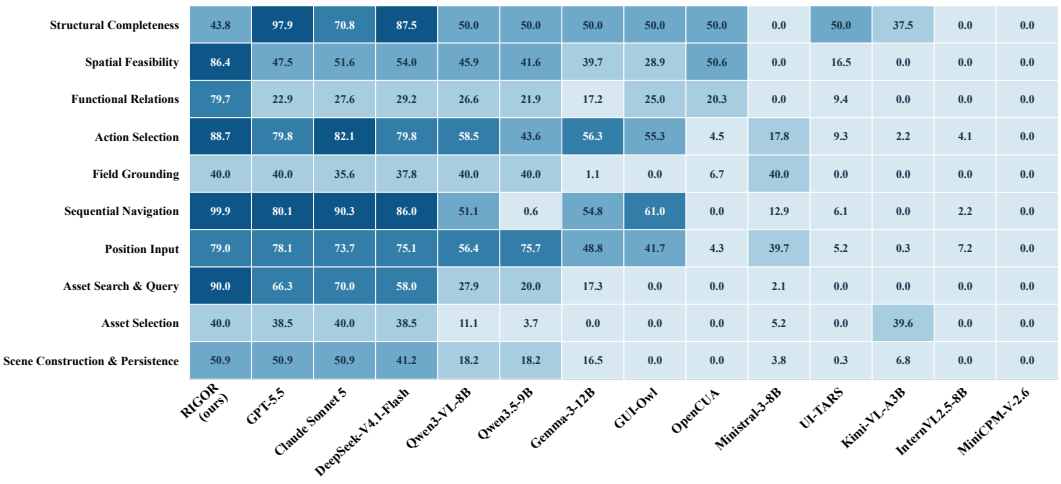

Figure 8: Original ten-dimension diagnostic values. The Spatial Feasibility label denotes the service-layout soft score, and ACT diagnostics use the type/step combination in Eq. 23.

C Industrial Task Construction and Screening Protocol

RIGOR constructs its static workcell tasks by deterministic parameter enumeration and rule-based feasibility screening. Public asset information and layout guidance are encoded as a scenario specification. A condition tuple specifies aisle width, independent-equipment clearance, boundary clearance, and aisle orientation. The construction program instantiates the template below and retains candidates that satisfy the implemented geometric and structural checks, yielding 291 task seeds from 450 candidates and 1,050 feasible layouts. The template defines the scope of the benchmark and the rules against which its layouts are evaluated.

Deterministic industrial task template

```
TASK_REQUIREMENT
Using the provided standardized factory environment and asset
    library,
design a static combined workcell for the kitting-to-assembly
    process.
Use exactly one instance of each required device:

1. kit_tray_station
2. AGV
3. kit_tray
4. assembly_table
5. assembly_insert
6. ceiling_robot_semantic_proxy
7. tool_cart
8. metal_office_desk
9. worn_metal_rack
10. plastic_container
11. wet_floor_sign_01

Preserve the following process and structural relations:
- kitting precedes assembly;
- kit_tray is placed on the AGV tray position;
- assembly_insert is placed on assembly_table;
- ceiling_robot_semantic_proxy serves the assembly area;
- tool_cart supports the kitting area;
- worn_metal_rack and plastic_container are separate assembly-
    storage
```

```
  positions;
- metal_office_desk is isolated from the main aisle;
- wet_floor_sign_01 marks a potential hazard area.

CONDITION_TUPLE
main_aisle_width_m: {{main_aisle_width_m}}
equipment_clearance_m: {{equipment_clearance_m}}
boundary_clearance_m: {{boundary_clearance_m}}
aisle_orientation: {{aisle_orientation}}

WORKSPACE
unit: meters
coordinate_convention: Unity left-handed; +Y is up; X-Z is the
ground plane
x_min_m: {{x_min_m}}
x_max_m: {{x_max_m}}
z_min_m: {{z_min_m}}
z_max_m: {{z_max_m}}

PUBLIC_ASSET_GEOMETRY
For every required device, provide its canonical local collision
    AABB:
semantic_device_id, asset_id, aabb_min_m, and aabb_max_m.
All values refer to scale [1, 1, 1]. The occupied world AABB is
    computed
by rotating the eight local corners by the predicted yaw and
    translating
them by the predicted anchor position.

HARD_CONSTRAINTS
- use exactly one instance of every required device;
- reserve a main aisle of at least {{main_aisle_width_m}} meters
    along
  {{aisle_orientation}};
- keep independent equipment at least
  {{equipment_clearance_m}} meters apart;
- keep every collision AABB at least {{boundary_clearance_m}} meters
  inside the selected work-area boundary;
- prevent overlap between independent devices and fixed environment
  geometry;
- preserve the stated parent and placement relations;
- produce a static initial scene only, without animation, kinematics
    ,
  ROS, PLC, or controller behavior.

SOFT_OBJECTIVES
- make the kitting-to-assembly flow readable and compact;
- place service and storage equipment near the work area it serves;
- avoid obstructing the reserved aisle;
- separate the office desk from traffic and assembly-risk regions;
- place the warning sign near the relevant hazard source.
```

The work area is selected by a fixed grid search over the public factory environment. For each condition tuple, the construction program chooses the first free $7.5\,\mathrm{m} \times 7.0\,\mathrm{m}$ rectangle under the fixed ordering rule and derives the reserved aisle solely from that rectangle and the public condition values. Candidate layouts are rejected if any independent device violates the boundary, clearance, aisle, collision, or parent-placement constraints. The remaining layouts are ranked by the same frozen soft-score implementation used throughout data construction. One primary layout is retained as the PLAN target for each task seed, while the other legal layouts remain candidate references. The recorded split is organized around protected task identities and held-out ACT interface profiles.

D PLAN–ACT SUPERVISION AND INFERENCE PROMPTS

RIGOR uses separate structured prediction contracts for planning and action prediction. PLAN receives the task requirement and public scene information and predicts an eleven-device layout; its evaluation input excludes reference coordinates and evaluator outputs. ACT receives the current Unity screenshot, an explicit frozen device or scene target, and the real preceding action history. In the reported given-target ACT evaluation, the target comes from the shared A+R+L layout. The scored next action and future action history are excluded from the ACT input.

D.1 PLAN PROMPT

The following system message is shared by the fixed PLAN evaluation. The user message is instantiated from the task template in Appendix C, the selected workspace, the public asset geometry, the frozen constraints, and the deterministically derived reserved aisle.

PLAN system prompt

```
MODE: PLAN_RESOLVED

You are a deterministic static layout planner. Read only the task
requirement, public workspace, asset geometry, and constraints in
    the
user message. Do not request or rely on gold coordinates, reference
layouts, target screenshots, scores, ranks, or hidden answers.

Return exactly one JSON object and nothing else. Do not use Markdown
fences, prose, comments, or extra keys. The strict output schema is:
{
  "mode": "PLAN_RESOLVED",
  "resolved_layout": {
    "devices": [
      {
        "semantic_device_id": "<one required device id>",
        "position_xyz_m": [<x>, <y>, <z>],
        "rotation_xyz_deg": [<rx>, <ry>, <rz>],
        "scale_xyz": [1, 1, 1],
        "parent_semantic_device_id": "<optional allowed parent>"
      }
    ]
  }
}

Schema requirements:
- resolved_layout.devices must contain exactly 11 objects, one for
    each
  required semantic_device_id. Do not omit, duplicate, rename, or
    add
  devices.
- Required ids are kit_tray_station, AGV, kit_tray, assembly_table,
  assembly_insert, ceiling_robot_semantic_proxy, tool_cart,
  metal_office_desk, worn_metal_rack, plastic_container, and
  wet_floor_sign_01.
- position_xyz_m contains exactly three finite numbers in meters in
    the
  Unity left-handed world frame (+Y up).
- rotation_xyz_deg contains exactly three finite numbers in Unity
    world
  Euler XYZ order.
- scale_xyz must be exactly [1, 1, 1].
- parent_semantic_device_id is optional. If present, it may only
    encode
  kit_tray -> AGV or assembly_insert -> assembly_table.
```

```
- Infer the layout only from the public input. Do not copy values
    from
  the schema illustration.

Output JSON only.
```

PLAN user-message structure

```
TASK_REQUIREMENT
{{rendered industrial task requirement}}

WORKSPACE
{{selected work-area rectangle and coordinate convention}}

ASSET_GEOMETRY
{{eleven public canonical local AABBs}}

CONSTRAINTS
{{aisle, clearance, boundary, process relations, hard constraints,
and soft objectives}}

SELECTED_WORK_AREA
{{deterministically selected rectangle}}

RESERVED_MAIN_AISLE
{{deterministically derived aisle rectangle}}

COLLISION_AABB
{{canonical local collision boxes and rotation semantics}}
```

D.2 ACT PROMPT

The ACT protocol preserves the native Unity GUI semantics. Clicks and drags use integer coordinates in the original screenshot; text entry preserves the requested string; and keyboard field transitions remain explicit `TAB_FIELD_TRANSITION` actions with their original transition fields.

ACT system prompt

```
You are a deterministic autonomous-next-action serializer. You
    receive
exactly one canonical Unity STATE screenshot, the complete target
    for the
currently selected device or scene, and the real action history up
    to that
screenshot. Choose exactly one safe action supported by the visible
    state
and history. Do not assume the recorded order, request a future
    action,
audit layout feasibility, or output an explanation.

Output exactly one JSON object and nothing else. Every output must
    contain:
  action_type: string
  device_line_id: string

device_line_id must be copied from the current target. It identifies
     the
```

```
logical device or global route, not a pixel location. Output every
    required
field for the selected action and omit all forbidden or null fields.

Allowed action types and exact grammars:

FOCUS_PROJECT_SEARCH
{"action_type":"FOCUS_PROJECT_SEARCH",
 "device_line_id":"<id>","operation":"click",
 "target_control":"project_search"}

TYPE_PREFAB_QUERY
{"action_type":"TYPE_PREFAB_QUERY",
 "device_line_id":"<id>","operation":"type_text",
 "text":"<exact query>"}

SELECT_PROJECT_RESULT
{"action_type":"SELECT_PROJECT_RESULT",
 "device_line_id":"<id>","operation":"click","click_count":1,
 "point":{"px":[<x>,<y>]}}

DRAG_PREFAB_TO_HIERARCHY
{"action_type":"DRAG_PREFAB_TO_HIERARCHY",
 "device_line_id":"<id>","operation":"drag",
 "start_point":{"px":[<x1>,<y1>]},
 "end_point":{"px":[<x2>,<y2>]},"move_count":<integer>}

CLICK_START_TRANSFORM_FIELD
{"action_type":"CLICK_START_TRANSFORM_FIELD",
 "device_line_id":"<id>","operation":"click","field":"<field>",
 "point":{"px":[<x>,<y>]}}

INPUT_TRANSFORM_VALUE
{"action_type":"INPUT_TRANSFORM_VALUE",
 "device_line_id":"<id>","operation":"type_text",
 "field":"<field>","text":"<exact text>",
 "target_value":"<requested value>"}

TAB_FIELD_TRANSITION
{"action_type":"TAB_FIELD_TRANSITION",
 "device_line_id":"<id>","operation":"key_press","key":"TAB",
 "count":<integer>,"from_field":"<field>","to_field":"<field>"}

CLICK_TRANSFORM_PX
{"action_type":"CLICK_TRANSFORM_PX",
 "device_line_id":"<id>","operation":"click","field":"PX",
 "tab_count":0,"point":{"px":[<x>,<y>]}}

SAVE_SCENE_ONCE
{"action_type":"SAVE_SCENE_ONCE",
 "device_line_id":"<id>","operation":"key_chord",
 "keys":["CTRL","S"]}

ASSERT_SCENE_CLEAN_AFTER_SAVE
{"action_type":"ASSERT_SCENE_CLEAN_AFTER_SAVE",
 "device_line_id":"<id>","operation":"state_assertion",
 "assertion":"scene_clean_after_save"}

DOUBLE_CLICK is not an allowed action type. A double TAB is
    represented as
TAB_FIELD_TRANSITION with count=2.

Coordinate contract:
```

```
- The user message gives the original screenshot width W and height
    H.
- Every point is an integer pixel pair with 0 <= x < W and 0 <= y <
    H.
- Do not output normalized, resized, cropped, translated, clamped,
    or
  processor-internal coordinates.
- The evaluation program performs any legacy normalization only
    after raw
  protocol validation.

Use the screenshot to ground visible controls and objects. Use the
    supplied
target and real history only to determine the current next action.
    Do not
output a plan, future action, score, source identity, or structured
    target.
```

ACT user-message structure

```
<CURRENT UNITY SCREENSHOT>

ORIGINAL_SCREENSHOT_SIZE: W={{width_px}}, H={{height_px}} pixels.
The screenshot is the original full canvas; every px coordinate must
     use
this exact W/H.

CURRENT_DEVICE_OR_SCENE_TARGET:
{
  "device_route_id": "{{device_route_id}}",
  "device_identity": "{{device_identity}}",
  "plan_root_id": "{{plan_root_id}}",
  "field_order": ["PX","PY","PZ","RX","RY","RZ","SX","SY","SZ"],
  "transform_fields": [
    {"field":"PX","group":"position",
     "target_text":"{{px_text}}","input_contract":"
    exact_text_for_this_field"},
    ...
  ]
}

REAL_HISTORY_TO_THIS_SCREENSHOT:
{{ordered prefix of recorded actions; the scored next action is
    excluded}}

Choose one action supported by this screenshot and real history.
    Return
exactly one action JSON using the system protocol and no explanation
    .
```

E Dataset Quality Assurance and LLM Evaluation Prompts

RIGOR separates deterministic data validation from subjective LLM evaluation. Deterministic checks establish whether a layout or trajectory is structurally traceable and whether an action can be scored. The blinded LLM raters assess conceptual PLAN quality and per-step ACT quality from the same evidence package without access to model identity, objective scores, reference answers, or rankings.

E.1 Deterministic Quality-Assurance Specification

The following specification summarizes the validation program. It is executed by code and is not presented to a language model as a scoring prompt.

Rule-based data validation specification

```
PLAN VALIDATION
1. Parse the raw response under the frozen strict JSON protocol.
2. Require exactly the eleven semantic device identities.
3. Require finite position, rotation, and unit-scale fields.
4. Allow only the two declared parent relations.
5. Reconstruct world AABBs from canonical local AABBs, anchor
    positions,
   and predicted yaw.
6. Evaluate boundary, independent-equipment clearance, reserved-
    aisle,
   collision, parent-placement, and public process-relation checks.
7. Preserve non-scorable outputs as zero-contribution cases under
    the fixed
   24-question denominator; never silently remove them.

ACT TRAJECTORY VALIDATION
1. Match every unit to one trajectory, screenshot, state id, action
    id,
   device route, and frozen target.
2. Verify screenshot bytes and original W/H before coordinate
    scoring.
3. Require the real history prefix to end before the scored next
    action.
4. Preserve native mouse, keyboard, text-entry, save, and assertion
    events.
5. Verify action-state continuity and pre/post screenshot
    correspondence.
6. For click actions, retain the authoritative target-control
    rectangle.
7. For text actions, retain the target field and exact target text.
8. For TAB actions, retain key, count, from_field, and to_field.
9. Mark insufficient evidence as UNKNOWN rather than converting it
    to PASS.
10. Retain malformed, missing, or failed predictions in the
    evaluation
    denominator according to the frozen scoring protocol.
```

E.2 Blinded PLAN Evaluation Prompt

PLAN LLM-rater prompt

```
You are an independent blinded evaluator. Score one anonymous
    industrial
layout using only the supplied task summary, device roles, anonymous
     layout
summary, and plan-view figure. Never infer the model identity,
    compare it
with another output, repair missing content, or add external
    evidence.

Do not perform exact geometry. Do not recompute AABB collisions,
    overlap
```

```
areas, boundary distances, device clearances, or aisle intrusion.
    These are
handled by the separate objective evaluator. Plainly visible
    overlaps,
extreme scatter, or incoherent organisation may still affect the
    visual
judgement.

Score four dimensions with integers from 0 to 5:

1. functional_relation_quality
5: Functional relations are complete and process-related devices
    form
   natural groups.
4: Overall reasonable, with one minor relational weakness.
3: Main relations are visible, but one or two are clearly awkward.
2: Some relations are sensible, but several key relations need
    revision.
1: Very few functional relations are recognisable.
0: No evaluable, relevant, or understandable layout.

2. workflow_coherence
5: Kitting, transport, assembly, storage, and support form a natural
     flow.
4: Coherent overall, with only a local routing weakness.
3: The main flow is understandable but contains a clear detour or
    break.
2: Multiple devices must be reorganised to form a coherent workflow.
1: A complete workflow is almost impossible to identify.
0: No layout or wholly task-irrelevant output.

3. spatial_organization_quality
5: Clear zoning, balanced density, natural support placement, and
    high
   readability.
4: Well organised, with only minor crowding or empty space.
3: Understandable overall, but visibly crowded, fragmented, or
    uneven.
2: Several regions require substantial rearrangement.
1: Nearly random scatter or severe clustering.
0: No evaluable layout.

4. overall_plan_quality
5: The conceptual industrial layout is almost directly usable.
4: High quality and requires only minor adjustment.
3: Usable foundation but requires clear revision.
2: Retains some useful structure but requires major revision.
1: Largely unusable, with very little worth retaining.
0: Unevaluable, irrelevant, or completely unusable.

overall_plan_quality is an independent holistic judgement, not the
    arithmetic
mean of the first three dimensions.

If the model was run but returned an empty, truncated, refused,
    malformed,
task-ignoring, or non-drawable output, assign 0 rather than N/A.

Return exactly one JSON object:
{
  "rater_id": "{{rater_id}}",
  "row_id": "{{row_id}}",
  "anonymous_model_id": "{{anonymous_model_id}}",
```

```
  "scores": {
    "functional_relation_quality": <integer 0-5>,
    "workflow_coherence": <integer 0-5>,
    "spatial_organization_quality": <integer 0-5>,
    "overall_plan_quality": <integer 0-5>
  },
  "reasons": {
    "functional_relation_quality": "<one short concrete Chinese
    sentence>",
    "workflow_coherence": "<one short concrete Chinese sentence>",
    "spatial_organization_quality": "<one short concrete Chinese
    sentence>",
    "overall_plan_quality": "<one short concrete Chinese sentence>"
  }
}

Output JSON only. Do not reveal chain-of-thought reasoning.
```

E.3 Blinded ACT Evaluation Prompt

ACT LLM-rater prompt

```
You are an independent blinded evaluator. Score one anonymous next-
    action
prediction using only the current Unity screenshot, the frozen task
    context,
the real history prefix, and the anonymous model output. Never infer
     the
model identity, compare it with another output, use an objective
    score or
error label, or add a reference action that is not shown.

Score four dimensions with integers from 0 to 5:

1. action_appropriateness
5: The action type is exactly appropriate for the current state,
    target,
   focus, and history.
4: The action class is correct, with only a minor equivalent-choice
    issue.
3: The general direction is useful but it is not the best immediate
    action.
2: The action type is clearly mismatched and will probably derail
    progress.
1: The action is only weakly related to the task.
0: Completely wrong, irrelevant, empty, refused, or unrecognisable.

2. parameter_grounding
5: Coordinates, field, text, TAB count, drag endpoints, query, or
    save/assert
   parameters exactly match the visible interface and target.
4: Parameters are correct with one harmless edge or formatting
    deviation.
3: The intended control is recognisable but a value, count, or
    location has
   a clear problem.
2: A key parameter selects the wrong control or value, or is missing
    .
1: Parameters barely correspond to the interface or target.
0: Parameters are unusable or unrelated.
```

```
3. task_progress_consistency
5: The action exactly advances the task without repetition, omission
    , or
   state damage.
4: It advances consistently, with only one harmless redundancy.
3: It advances but skips a necessary step or repeats completed work.
2: The ordering is clearly wrong and likely requires recovery.
1: It provides almost no progress or returns to a completed state.
0: It contradicts or damages the current state, or progress is
    unidentifiable.

4. overall_act_quality
5: The proposed step can be executed directly.
4: Executable with only a harmless minor issue.
3: Basically executable after one clear correction.
2: Requires substantial correction before execution.
1: Largely non-executable, with little usable content.
0: Completely non-executable, irrelevant, or unevaluable.

overall_act_quality is an independent holistic judgement, not the
    arithmetic
mean of the first three dimensions. There is no reference action or
    correct
answer in the rater package. Judge whether the proposed action is
    supported
by the screenshot, target, and history.

If the output is empty, truncated, refused, malformed, or task-
    ignoring,
assign 0 rather than N/A.

Return exactly one JSON object:
{
  "rater_id": "{{rater_id}}",
  "row_id": "{{row_id}}",
  "anonymous_model_id": "{{anonymous_model_id}}",
  "scores": {
    "action_appropriateness": <integer 0-5>,
    "parameter_grounding": <integer 0-5>,
    "task_progress_consistency": <integer 0-5>,
    "overall_act_quality": <integer 0-5>
  },
  "reasons": {
    "action_appropriateness": "<one short concrete Chinese sentence
    >",
    "parameter_grounding": "<one short concrete Chinese sentence>",
    "task_progress_consistency": "<one short concrete Chinese
    sentence>",
    "overall_act_quality": "<one short concrete Chinese sentence>"
  }
}

Output JSON only. Do not reveal chain-of-thought reasoning.
```

The subjective evaluation uses three independent blinded LLM raters. Each model contributes 72 PLAN ratings and 36 ACT ratings. LLM-Judge equally weights the model-level PLAN and ACT holistic scores. Strict complete-JSON parsing succeeds for 1,510 of 1,512 ratings; the other two use the versioned `SCORE_ONLY_EXTRACTION_V1` rule to read the existing complete `scores` object from the preserved raw response. This procedure retains the original scores without an additional model call. Rating counts describe subjective evaluation and are separate from the ACT-unit denominators in the objective comparison.

F EVALUATION METRICS

To comprehensively evaluate industrial scene generation, we design a three-part evaluation protocol covering layout planning, GUI action execution, and subjective output quality. We further combine the objective PLAN and ACT scores into a unified Overall Composite Score (OCS). Unless stated otherwise, all objective metrics are reported on a 0–100 scale. The subjective LLM-Judge score remains on its original 0–5 scale and is not included in OCS.

F.1 PLAN METRICS

Deterministic layout-quality score. For each PLAN question, the predicted JSON layout is parsed under the frozen schema and materialized into world-space axis-aligned bounding boxes (AABBs). Let $d(a, b)$ denote the Euclidean distance between the ground-plane projections of two AABBs. The deterministic evaluator converts the relevant device-pair distances into six normalized soft components in $[0, 1]$: S_1 measures the kitting-to-assembly process distance; s_{tool} measures tool-cart proximity to the kitting station; s_{rack} and $s_{\text{container}}$ measure storage proximity to the assembly area; s_{desk} rewards separation of the office desk from the AGV and robot; and s_{sign} rewards placement of the warning sign near the relevant hazard region. The service-layout component is

$$S_2(q) = \frac{1}{5}\left(s_{\text{tool}}(q) + s_{\text{rack}}(q) + s_{\text{container}}(q) + s_{\text{desk}}(q) + s_{\text{sign}}(q)\right). \tag{7}$$

The per-question layout-quality score is

$$S_{\text{total}}(q) = 0.60S_1(q) + 0.40S_2(q). \tag{8}$$

The paper-level PLAN score uses the fixed set of 24 questions:

$$S_{\text{total},24} = \frac{100}{24}\sum_{q=1}^{24} \widetilde{S}_{\text{total}}(q), \tag{9}$$

where $\widetilde{S}_{\text{total}}(q) = S_{\text{total}}(q)$ for an Authority-scorable layout and 0 otherwise. Invalid, incomplete, and non-scorable predictions therefore contribute zero under the fixed denominator. The nonzero score measures the specified soft layout objectives; a value on this scale is not a percentage of layouts satisfying every hard constraint.

Relation Satisfaction. Each question contains eight frozen functional relations describing the industrial workflow and service dependencies. Let $h_{q,r} = 1$ when relation r is satisfied by the materialized layout for question q, and 0 otherwise. Relation Satisfaction is

$$\text{Relation} = \frac{100}{24 \times 8}\sum_{q=1}^{24}\sum_{r=1}^{8} h_{q,r}. \tag{10}$$

The denominator is therefore fixed at 192. A non-scorable PLAN contributes no relation hits, but its eight relation checks remain in the denominator.

F.2 ACT METRICS

ACT is evaluated on a fixed set of frozen GUI states and frozen A+R+L targets shared by all models. Let $\mathcal{U}$ denote this common ACT evaluation set, with $N = |\mathcal{U}|$. Each model is evaluated on exactly the same units, and each predicted action is compared with the same set of state-valid reference actions defined by the frozen metadata.

Action Accuracy. Let $A_{m,u} = 1$ if the action type predicted by model m for unit u matches at least one state-valid reference action type, and 0 otherwise. Action Accuracy is

$$\text{Action}(m) = \frac{100}{N} \sum_{u \in \mathcal{U}} A_{m,u}. \tag{11}$$

This metric evaluates semantic action selection without requiring every parameter to be correct.

Step Success. Let $E_{m,u} = 1$ only when the complete action predicted by model m for unit u is executable and realizes effective progress toward the frozen target. This requires a valid output protocol, a state-valid action type, correct action parameters, valid click geometry when applicable, and correct target realization. Step Success is

$$\text{Step}(m) = \frac{100}{N} \sum_{u \in \mathcal{U}} E_{m,u}. \tag{12}$$

An UNKNOWN, missing, protocol-invalid, or failed prediction contributes zero to the numerator while the corresponding unit remains in the fixed evaluation set $\mathcal{U}$. This keeps the evaluation denominator identical across models.

Navigation Success. The navigation subset $\mathcal{N} \subseteq \mathcal{U}$ is selected from the frozen reference action type and is therefore identical across models. It contains sequential TAB transitions and direct transform-field clicks. A TAB action succeeds only when its key, transition count, source field, and destination field all match. A field-click action succeeds only when the declared field is correct and the predicted pixel lies inside the authoritative field rectangle. Let $N_{m,u} = 1$ when model m satisfies these conditions for unit u. Then

$$\text{Navigation}(m) = \frac{100}{|\mathcal{N}|} \sum_{u \in \mathcal{N}} N_{m,u}. \tag{13}$$

Input Success. The input subset $\mathcal{I} \subseteq \mathcal{U}$ contains units whose frozen reference action is `INPUT_TRANSFORM_VALUE` and is identical across models. Let $I_{m,u} = 1$ when decisive focus evidence is available and the predicted field, exact input text, and semantically equivalent target value all match the frozen PLAN target. Input Success is

$$\text{Input}(m) = \frac{100}{|\mathcal{I}|} \sum_{u \in \mathcal{I}} I_{m,u}. \tag{14}$$

Interaction Success. The interaction subset $\mathcal{R}_{\text{int}} \subseteq \mathcal{U}$ contains project search, prefab-query, result-selection, prefab-drag, scene-save, and clean-scene-assertion actions and is identical across models. An interaction unit succeeds when its complete step satisfies the same requirements as Eq. 12. Therefore,

$$\text{Interaction}(m) = \frac{100}{|\mathcal{R}_{\text{int}}|} \sum_{u \in \mathcal{R}_{\text{int}}} E_{m,u}. \tag{15}$$

The Navigation, Input, and Interaction subsets are assigned from frozen reference metadata rather than model predictions. Their memberships and denominators are therefore fixed and identical across all evaluated models.

F.3 LLM-Based Subjective Evaluation

The subjective evaluation scores PLAN and ACT independently on a discrete 0–5 scale. PLAN is assessed along functional relations, workflow coherence, spatial organization, and overall PLAN quality. ACT is assessed along action appropriateness, parameter grounding, task-progress consistency,

and overall ACT quality. The main LLM-Judge metric uses the two overall-quality dimensions; the remaining dimensions support fine-grained analysis and rater agreement measurement.

Let P_m and A_m denote the rated PLAN and ACT objects for model m, and let $\mathcal{R}$ denote the set of independent raters. Let $p_{i,r} \in \{0, 1, 2, 3, 4, 5\}$ be the overall PLAN-quality score assigned by rater r to PLAN object i, and let $a_{j,r}$ be the corresponding overall ACT-quality score. The two task-level subjective scores are

$$\mathrm{Judge}_{\mathrm{PLAN}}(m) = \frac{1}{|P_m||\mathcal{R}|} \sum_{i \in P_m} \sum_{r \in \mathcal{R}} p_{i,r}, \tag{16}$$

$$\mathrm{Judge}_{\mathrm{ACT}}(m) = \frac{1}{|A_m||\mathcal{R}|} \sum_{j \in A_m} \sum_{r \in \mathcal{R}} a_{j,r}. \tag{17}$$

Their equally weighted aggregate is

$$\text{LLM-Judge}(m) = \frac{1}{2} \left(\mathrm{Judge}_{\mathrm{PLAN}}(m) + \mathrm{Judge}_{\mathrm{ACT}}(m) \right). \tag{18}$$

The score remains on the original 0–5 scale and averages the two designated holistic quality scores. The other rubric dimensions support separate diagnostic analysis.

Rater agreement. Agreement is computed separately for PLAN and ACT. Exact Agreement is the proportion of available rater pairs that assign the same score, whereas Within-One Agreement is the proportion whose absolute difference is no larger than one point. Ordinal Krippendorff's α is defined as:

$$\alpha = 1 - \frac{D_o}{D_e}, \tag{19}$$

where D_o and D_e are the observed and expected disagreements under the ordinal distance function. This formulation penalizes distant disagreements more strongly than adjacent disagreements. Missing ratings, if present, are omitted pairwise.

F.4 Overall Composite Score

Following the two-stage structure of the task, OCS first aggregates the two objective PLAN metrics and the five objective ACT metrics separately. The PLAN score is

$$\mathrm{PLANScore}(m) = 0.7\, S_{\mathrm{total},24}(m) + 0.3\, \mathrm{Relation}(m). \tag{20}$$

The ACT score is

$$\begin{aligned} \mathrm{ACTScore}(m) = {} & 0.1\, \mathrm{Action}(m) + 0.3\, \mathrm{Step}(m) \\ & + 0.2\, \mathrm{Navigation}(m) + 0.2\, \mathrm{Input}(m) + 0.2\, \mathrm{Interaction}(m). \end{aligned} \tag{21}$$

The final Overall Composite Score equally weights planning and execution:

$$\mathrm{OCS}(m) = \frac{1}{2} \left(\mathrm{PLANScore}(m) + \mathrm{ACTScore}(m) \right). \tag{22}$$

All seven inputs to OCS are normalized scores or success percentages on the 0–100 scale. OCS is an arithmetic composite of separately evaluated stages; it is not an observed end-to-end construction success rate. The larger weight on $S_{\mathrm{total},24}$ reflects the central role of engineering-aware layout quality within PLAN. Within ACT, Step Success receives the largest weight because it jointly requires the intended action and its precise parameters to be correct. LLM-Judge and all diagnostic statistics are excluded from OCS.

F.5 Diagnostic Capability Metrics

The capability analyses use derived metrics to reveal where performance gaps arise. These values are not additional main-table metrics and do not enter OCS. Planning Quality is identical to the objective PLAN aggregate in Eq. 20. For an ACT capability category c, we combine action selection and exact step realization as

$$\mathrm{Capability}(m, c) = 0.4\,\mathrm{Action}(m, c) + 0.6\,\mathrm{Step}(m, c). \tag{23}$$

The category of each ACT unit is assigned using its frozen reference metadata, never the predicted action. The PLAN diagnostic analysis additionally uses Structural Completeness and Spatial Feasibility. For question q, Structural Completeness is

$$\mathrm{Structural}(q) = 100\left[0.5\frac{n_{\mathrm{device}}(q)}{11} + 0.5\mathbb{I}(\mathrm{parent}(q) = \mathrm{PASS})\right], \tag{24}$$

and Spatial Feasibility is

$$\mathrm{Spatial}(q) = 20\left(s_{\mathrm{tool}}(q) + s_{\mathrm{rack}}(q) + s_{\mathrm{container}}(q) + s_{\mathrm{desk}}(q) + s_{\mathrm{sign}}(q)\right). \tag{25}$$

The diagnostic named Spatial Feasibility equals $100S_2$ and measures the soft service-layout terms; it does not count all-hard-constraint passes. Both diagnostics use the fixed 24-question average, with non-scorable questions contributing zero. Functional Relations is identical to Eq. 10.

F.6 Aggregation and Failure Handling

All metrics retain their predefined denominators. A malformed PLAN remains in the 24-question average, and an unsuccessful ACT prediction remains in the fixed ACT evaluation set. A score is unavailable only when the metric is structurally inapplicable, such as an operation category with no frozen reference units. We apply no smoothing, extrapolation, or post-hoc imputation. All models are evaluated on the same frozen ACT manifest and therefore share identical denominators for the overall ACT evaluation and each operation-specific subset. PLAN and ACT are reported independently before their explicit combination in Eq. 22.

G Fine-Grained Recorded Action Composition

Section 4 reports the three aggregate action groups in the recorded process corpus. This appendix provides their fine-grained decomposition. Within navigation, TAB navigation accounts for 27.85% of all action units, initial focus for 3.80%, and direct clicks for 2.53%. Within parameter input, position, rotation, and scale each contribute 11.39% after rounding. Within scene interaction, search and selection account for 18.99%, object placement for 7.59%, and completion and persistence for 5.06%. These percentages describe the expert recordings used for process supervision; the frozen ACT evaluation manifest is fixed and identical across all evaluated models.