VA-BENCH: FROM SPATIAL REASONING TO EMBODIED ACTION IN GENERAL-PURPOSE VLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatial intelligence in vision-language models is commonly evaluated through question answering over pre-specified visual observations. Robot manipulation offers a more demanding test, but existing evaluations typically assess learned visuomotor policies rather than the spatial intelligence of general-purpose VLMs. We introduce VA-BENCH, which uses closed-loop embodied manipulation to evaluate whether general VLMs can turn spatial understanding into physical action. Given a task goal and visual demonstrations of the task procedure, the model must infer action-relevant geometry and generate metric robot actions without relying on a task-specific action policy or privileged object or scene geometry. During execution, it can perform **active perception** by choosing additional viewpoints when needed and revise subsequent actions from interaction feedback. We refer to this overall capability as **self-directed metric spatial grounding**. VA-BENCH further tests **geometric shifts** by preserving the task procedure while changing the action-relevant geometry. Across 14 tasks, existing VLMs show near-perfect target identification and task understanding, yet remain substantially weaker at spatial grounding and online correction; the highest overall task success rate is only 53.9%. Active perception improves success by 7.86–23.93% over controls given predefined views, while geometric shifts cause drops of up to 32.14%. Overall, these results show that for general VLMs, knowing what to do does not yet translate into knowing where and how to act reliably.

1 INTRODUCTION

Spatial intelligence is useful for robotics only when it can support correct physical action. Existing vision-language benchmarks mainly evaluate this ability through spatial question answering or prediction from pre-specified observations (Fu et al., 2024; Yang et al., 2025a). These settings can reveal whether a model recognizes spatial relationships, but successful manipulation imposes a stricter requirement. The model must estimate spatial information precisely enough to determine how the robot should move, and its decisions must remain valid as the interaction changes the scene.

Robot manipulation benchmarks provide physical execution, but they usually answer a different question. Many evaluate visuomotor policies learned from robot demonstrations or large-scale action data, so their performance reflects policy learning, action representation, and control in addition to spatial reasoning. Other embodied evaluation pipelines provide spatial annotations, poses, depth, or action abstractions that resolve part of the spatial problem outside the evaluated model. Consequently, neither spatial QA nor conventional robot-policy evaluation directly answers a basic question: can a general-purpose VLM use its own spatial intelligence to guide physical interaction without learning a task-specific action policy?

To investigate the above question, we present VA-BENCH to evaluate whether general VLMs can construct and continually update the metric spatial information needed for physical action through closed-loop manipulation. Given a task goal and visual demonstrations of the task procedure, the model must infer how that procedure applies to the current scene and generate metric robot actions without privileged object or scene geometry. When the current observation is insufficient, it can acquire additional viewpoints through active perception. After acting, it must use the resulting scene and execution feedback to revise subsequent decisions. We refer to this ability to construct and maintain actionable spatial information through interaction as self-directed metric spatial ground-

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.

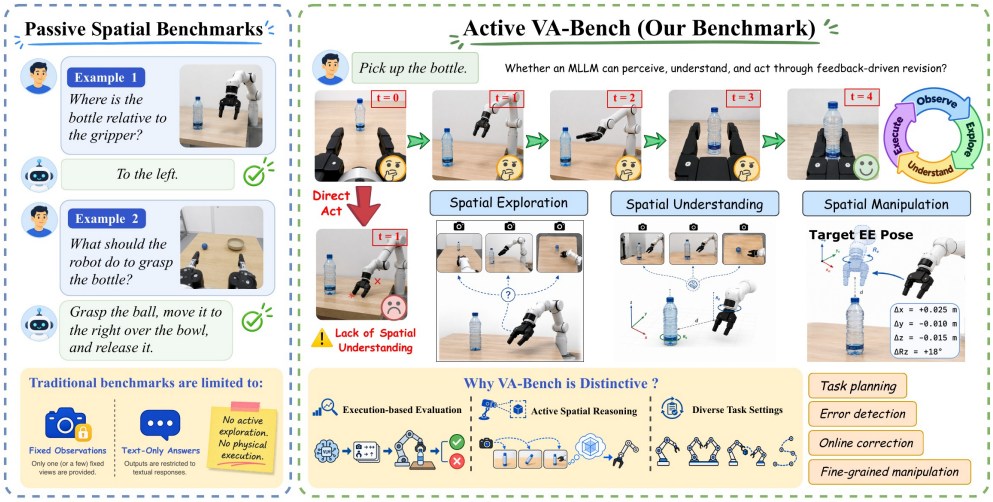


Figure 1: Overview of the evaluation target. Fixed-view spatial evaluations can stop at a verbal description, whereas VA-Bench requires an MLLM to actively acquire evidence, infer the geometry between the robot and object, issue metric manipulation commands, and revise them from execution outcomes.

ing. The fixed executor handles only low-level trajectory generation, leaving task-relevant spatial decisions to the VLM.

This design allows us to study spatial intelligence beyond overall task success. VA-BENCH contains 14 manipulation tasks and introduces controlled tests for two important failure modes. The first compares active perception with predefined visual coverage. The second introduces geometric shifts, where the task procedure remains unchanged but the geometry relevant to execution changes. These settings test whether a model can obtain useful evidence and re-ground a known procedure to a new spatial configuration.

The results expose a clear gap between task understanding and spatial execution. The strongest models reach 100% target localization and at least 99.6% accuracy in recognizing the intended manipulation, yet the highest overall task success rate is only 53.9%. Their spatial-relation reasoning and online correction remain substantially weaker. Active perception improves success by 7.86–23.93% over predefined-view controls, while geometric shifts cause drops of up to 32.14%. These results suggest that current VLMs often understand what should be done, but still struggle to determine where and how to do it reliably.

In summary, our contributions are threefold:

- We introduce an embodied evaluation paradigm that uses closed-loop manipulation to directly probe the spatial intelligence of general-purpose VLMs, rather than spatial QA or learned robot policies.
- We develop VA-BENCH, a 14-task benchmark that evaluates self-directed metric spatial grounding through metric action generation, active perception, and feedback-driven correction.
- We show that strong task understanding does not translate into reliable spatial execution, and identify active evidence acquisition and scene-specific geometric grounding as major remaining limitations.

2 RELATED WORK

2.1 SPATIAL UNDERSTANDING UNDER PARTIAL OBSERVATION

Spatial reasoning benchmarks span static images, supplied 3D observations, and interactive environments (Liu et al., 2023b; Kamath et al., 2023; Azuma et al., 2022; Ma et al., 2023; Du et al.,

2024; Feng et al., 2026; Liu et al., 2026b; Zhang et al., 2026a; Jaber et al., 2026; Zhou et al., 2025). CLEVR and GQA evaluate compositional visual reasoning, while SpatialVLM, SpatialRGPT, CV-Bench, BLINK, Thinking in Space, and 3DSRBench extend this to spatial relations, depth, distance, and 3D scene understanding (Johnson et al., 2017; Hudson & Manning, 2019; Chen et al., 2024; Cheng et al., 2024; Tong et al., 2024; Fu et al., 2024; Yang et al., 2025a; Ma et al., 2025). Quantitative Spatial Reasoning, RoboSpatial, SpatialBot, RoboRefer, and Metric-Bench further emphasize quantitative, grounded, and robot-centric spatial understanding (Liao et al., 2024; Song et al., 2025; Cai et al., 2025; Zhou et al., 2025; Xi et al., 2026), while ManipBench tests low-level spatial reasoning from supplied observations (Zhao et al., 2025).

More recent benchmarks such as Theory of Space, ESI-Bench, and SpatialWorld allow agents to acquire additional observations under partial observability (Zhang et al., 2026b; Hong et al., 2026; Gao et al., 2026). These works advance active evidence acquisition, but typically retain a benchmark-defined spatial target or evaluate spatial beliefs and high-level decisions. VA-BENCH instead couples active evidence acquisition with metric spatial grounding: the VLM must determine what task-relevant spatial information is needed, acquire evidence for it, and ground its estimate quantitatively for action.

2.2 EMBODIED MANIPULATION AND ROBOT BENCHMARKS

Meta-World, RLBench, CALVIN, LIBERO, ManiSkill, ManiSkill2, BEHAVIOR-1K, RoboCasa, and RoboTwin 2.0 provide environments and benchmarks for learning and evaluating manipulation skills (Yu et al., 2020; James et al., 2020; Mees et al., 2022; Liu et al., 2023a; Mu et al., 2021; Gu et al., 2023; Li et al., 2023; Nasiriany et al., 2024a; Chen et al., 2025), while RT-1, RT-2, OpenVLA, and Octo learn robot action generation from demonstrations or large-scale robot data (Brohan et al., 2023; Zitkovich et al., 2023; Kim et al., 2025; Ghosh et al., 2024). Related efforts further study robot imitation and large-scale cross-embodiment learning (Li et al., 2024b; O'Neill et al., 2024).

More directly related benchmarks evaluate multimodal models through executed manipulation, including VLMbench and VLABench (Zheng et al., 2022; Zhang et al., 2025). Methods such as PIVOT, RoboPoint, ReKep, ManipLLM, A3VLM, VoxPoser, and MOKA translate multimodal reasoning into actionable spatial representations or robot actions (Nasiriany et al., 2024b; Yuan et al., 2025; Huang et al., 2025b; Li et al., 2024a; Huang et al., 2025a; 2023a; Fang et al., 2024). EmbodiedBench provides object annotations and 3D coordinates, ST-BiBench uses auxiliary ground-truth poses, IMBench executes parameterized motor primitives, and ESPIRE lifts image-space predictions to robot poses using ground-truth depth (Yang et al., 2025b; Wu et al., 2026; Maurya et al., 2026; Zhao et al., 2026). In these settings, some task-relevant spatial information is supplied or resolved by the evaluation pipeline. VA-BENCH instead evaluates whether a general-purpose VLM can construct the spatial information needed for action, with physical execution serving as an action-grounded readout of its spatial estimates rather than a learned control objective.

2.3 ACTIVE PERCEPTION AND DEMONSTRATION-GROUNDED CONTROL

Active perception and next-best-view planning use sensing actions to reduce uncertainty or improve geometric coverage, while Vision in Action and SaPaVe extend this line toward task-relevant camera behavior and active perception for manipulation (Xiong et al., 2025; Liu et al., 2026a). In VA-BENCH, active evidence acquisition is part of the evaluated capability rather than a separately learned perception policy: the VLM must recognize when task-relevant spatial information is incomplete and select additional observations that help resolve it.

Related demonstration-based methods such as RoboMimic, BC-Z, MimicPlay, and MimicGen use demonstrations as training signals for robot control (Mandlekar et al., 2022; Jang et al., 2022; Wang et al., 2023; Mandlekar et al., 2023). Other imitation-learning approaches, including CLIPort, Perceiver-Actor, RVT, Diffusion Policy, and VIMA, similarly learn manipulation policies from demonstration data (Shridhar et al., 2022; 2023; Goyal et al., 2023; Chi et al., 2023; Jiang et al., 2023). By contrast, VA-BENCH uses RGB-only demonstrations as procedural context at evaluation time: they specify how the task is performed, but not the metric solution for the current scene. The model must therefore re-ground the demonstrated procedure to the task-relevant spatial configuration of each evaluation instance.

Table 1: Comparison of evaluation protocols on varied dimensions. Entries describe the cited evaluation protocols rather than all capabilities of their underlying simulators.

Setting	Model-selected views	Executed arm actions	Interleaved view–arm control	General VLM, no learned action head
ManipBench	No	No	No	Yes
Theory of Space	Yes	No	No	Yes
ESI-Bench	Yes	No	No	Yes
SpatialWorld	Yes	No	No	Yes
EmbodiedBench (manipulation)	No	Yes	No	Yes
ST-BiBench (fine control)	No	Yes	No	Yes
IMBench (action stage)	No	Yes	No	Yes
ESPIRE (execution)	No	Yes	No	Yes
VA-BENCH	Yes	Yes	Yes	Yes

Across these lines of work, the central distinction of VA-BENCH is the allocation of spatial responsibility. The VLM must determine what task-relevant spatial information is needed, acquire additional evidence when necessary, ground its estimate quantitatively, and update subsequent decisions through interaction.

Table 1 compares representative protocols along four dimensions: whether the evaluated model can select additional viewpoints, execute robot-arm actions, interleave viewpoint selection with manipulation within the same episode, and operate as a general-purpose VLM without a learned action head. Executed arm actions include parameterized motor primitives and geometrically lifted robot poses, but not spatial QA or symbolic interactions alone. The final criterion permits fixed, model-agnostic components such as inverse kinematics and trajectory execution.

3 VA-BENCH: BENCHMARK DESIGN AND EVALUATION

VA-BENCH evaluates *self-directed metric spatial grounding* through closed-loop manipulation in physics-based simulation. A general-purpose VLM is given a high-level task instruction and RGB-only visual demonstrations, but not the object geometry or metric solution for the current scene. It must determine what task-relevant spatial information is needed, acquire additional observations when necessary, generate metric robot commands, and revise subsequent decisions from execution feedback. Low-level motion execution is standardized across models so that task-relevant spatial decisions remain the responsibility of the VLM.

3.1 BENCHMARK CONSTRUCTION AND TASK SUITE

VA-BENCH contains three complementary evaluation tracks. The *base suite* measures closed-loop spatial performance across diverse manipulation settings. The *geometric-shift track* preserves the demonstrated task procedure while changing action-relevant scene geometry, testing whether models can adapt the same procedure to new spatial configurations. The *long-horizon track* composes demonstrated atomic skills into a longer multi-object task, testing whether local spatial competence can be sustained across repeated interactions. All three tracks allow the model to acquire additional viewpoints when needed.

Base task suite. VA-BENCH contains 14 tasks: 11 single-arm tasks and 3 dual-arm tasks. The tasks cover complementary spatial demands, including robot to object offsets, height and depth alignment, orientation and contact geometry, relations between objects and containers, and bimanual relative geometry. Each task contains 20 deterministic scene seeds retained after settling and physical verification, yielding 280 base instances. Every retained instance admits a complete expert execution from the robot home state to terminal success. The suite is fixed before evaluation and the same instances are reused across models and runs. Appendix C.3 summarizes the task families, and Appendix C.4 gives the full construction and verification procedure.

Geometric-shift track. Seven single-arm tasks include variants that preserve the instruction and demonstrated procedure while changing action-relevant scene geometry. Variations include object

geometry, container or box instances, spatial layouts, and additional distractors. The model must therefore adapt the same procedure to the current scene rather than reuse a scene-specific solution.

Long-horizon composition. A separate five-object basket task composes four demonstrated atomic manipulation skills into a longer placement sequence requiring sustained progress tracking and repeated adaptation as the scene changes. Because its demonstration structure and interaction horizon differ from the base suite, we report this track separately.

All three tracks allow the model to move the camera and inspect additional viewpoints when needed. Section 5.1 introduces matched passive-view controls that remove this choice and instead provide fixed multi-view coverage.

3.2 Model Interface and Evaluation Harness

The evaluation interface separates decisions made by the VLM from operations handled by the benchmark infrastructure. All evaluated models use the same observation interface, action interface, interaction harness, and low-level executor; the VLM is the only model-dependent component.

Model-visible information. The model receives the task instruction, a demonstration-derived procedural summary, the current RGB observation, filtered robot proprioception, and the retained interaction history. Proprioception includes the end-effector pose, gripper state, finger midpoint (GC), and the current gripper-local axes expressed in world coordinates for each controlled arm. The interaction history contains previous model actions and textual feedback, including action-validity and trajectory-planning outcomes.

Camera and robot actions. The model may acquire additional visual evidence through bounded camera actions. These include gripper-centered semantic viewpoints together with fixed-step local translation, zoom, yaw, and pitch adjustments. Camera actions change the available observation without manipulating the task objects. Exact camera-action definitions and bounds are given in Appendix C.6.

Robot actions are numerical commands generated directly by the VLM. The action space contains world-axis translations, gripper-local rotations, and gripper aperture commands; dual-arm tasks additionally support synchronized variants. Translations range from 1–100 mm and rotations from 1°–90°. The model therefore specifies metric actions rather than selecting from predefined motion categories. Appendix C.5 gives the complete robot-action schema and bounds.

Withheld information. The model does not receive target markers, keypoints, segmentation masks, depth, object coordinates or poses, contact state, task-checker state, task waypoints, numerical camera calibration, offline seed-verification probes, or seed provenance. These signals remain evaluator-only.

Evaluation harness and fixed executor. A shared model-agnostic harness maintains the interaction history, validates structured model decisions, and dispatches camera or robot commands. A fixed executor then checks reachability and generates trajectories for the metric robot targets specified by the VLM. The executor does not select the task object, grasp region, arm role, task waypoint, or corrective action; these spatial decisions remain with the VLM. The primary evaluation uses no learned robot policy, model-specific action head, or learned RGB-D or pre-grasp spatial module. Appendix D describes the complete harness and execution stack.

3.3 Interactive Evaluation Protocol

Demonstration conditioning. Before evaluation, model m converts RGB-only frames from a successful demonstration of task family τ into a textual procedural summary $s_{m,\tau}$. The demonstration provides procedural context—e.g., grasp region, approach strategy, or arm coordination—but no metric solution for the evaluation scene. Neither the demonstration nor its summary contains machine-readable actions, robot or object poses, contact labels, task waypoints, or numerical action targets. At test time, the model must re-ground the procedure in the current scene and generate all numerical action parameters itself. Held-out variants reuse the corresponding base-task summary, while the long-horizon task provides only four atomic-skill demonstrations and no demonstration of the complete sequence. Full details are given in Appendix C.9.

Closed-loop interaction. Each episode provides a high-level task instruction, demonstration context, current RGB observation, filtered robot proprioception, interaction history, previous execution feedback, and an action budget B_τ; the scene configuration exposes no privileged object geometry to the model. At each step, the model chooses either a camera action, which changes the viewpoint without modifying the scene, or a robot action executed by the fixed controller. After each action, it observes the resulting scene and feedback and may revise its spatial estimate before acting again. Camera and robot actions consume the same budget, making additional evidence acquisition explicitly costly. An episode terminates upon task success, model termination, or budget exhaustion. Appendix C.1 provides the complete episode specification.

3.4 Evaluation Metrics and Behavioral Diagnostics

Terminal task success. The primary benchmark outcome is task completion determined by the task-specific checker. For model m, run r, task family τ, and valid scene seed i, terminal success is

$$y_{m,r,\tau,i} = \mathbf{1}\left[\phi_\tau(x_{t_{\text{end}}}) = 1\right], \qquad t_{\text{end}} \le B_\tau. \tag{1}$$

The success predicate is checked after every dispatched environment action. Stopping or exhausting the interaction budget without satisfying the predicate counts as failure, with no post-hoc exclusions. Each task family receives equal weight in the base-suite macro-average. Terminal success therefore measures the complete VLM–interface–executor system and remains the primary evaluation criterion.

Subtask progress. We additionally measure intermediate progress using task-specific observable checkpoints. A checkpoint receives credit only when the required state transition is realized in the executed trajectory; issuing the intended command or obtaining a successful planner return is insufficient. Subtask completion is micro-averaged over scored checkpoint instances and reported separately from terminal success. The complete definition is given in Appendix C.10.

Behavioral diagnostics. Terminal success reveals whether an episode succeeds but not where the interaction breaks down. We therefore annotate nine trajectory-level behavioral criteria on one complete base-suite run per evaluated condition. The criteria cover three stages of the interaction loop. Spatial evidence acquisition and interpretation are characterized by target localization, active exploration, and robot–object spatial relations (TL/AE/SR). Manipulation is characterized by manipulation-semantics understanding, goal-directed manipulation planning, and fine-grained pre-contact analysis (MS/MP/FG). Recovery is characterized by error detection, online correction, and post-failure adjustment (ED/OC/PF).

Three annotators independently label each applicable episode–criterion pair, and the reported label is determined by majority vote. Recovery criteria apply only when an observable error occurs. These annotations are diagnostic summaries of executed behavior rather than alternative definitions of task success; terminal success remains the primary outcome. Appendix C.10 provides the full rubric, applicability rules, subtask-progress definition, and agreement analysis.

4 Main Results and Analysis

We evaluate 12 demonstration-conditioned agents using model-specific summaries. Terminal success is a task-family macro-average reported as mean $\pm$ sample standard deviation across three runs; diagnostics and subtask progress use one annotated run per model. Appendix A.1 specifies evaluation conditions and aggregation, with complete per-task and arm-specific results in Appendix A. We first characterize performance under the complete VA-BENCH protocol before isolating the roles of evidence acquisition, geometric re-grounding, and demonstration conditioning in Section 5.

4.1 Base-task performance

Table 2 compares 12 primary model conditions on the 11 single-arm and three dual-arm tasks. Qwen3.8-max, Opus-5, and GPT-5.6-sol form a clear upper group, but their three-run means lie within 2.38 points with overlapping ranges, so we do not claim a reliable ordering. The same three models also achieve the highest subtask-completion scores, with only a 2.7-point difference among them. This checkpoint-based metric captures intermediate progress separately from terminal task success. Performance then falls sharply, while coordinated dual-arm execution remains difficult, with the best mean at 22.22%. Appendix Tables 5–7 give complete task-level results.

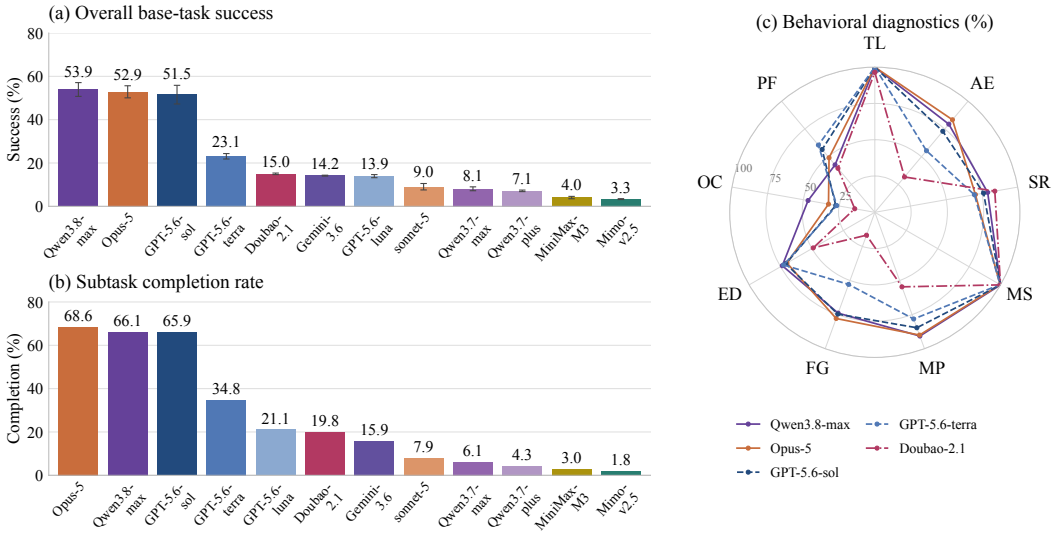


Figure 2: Execution success, subtask progress, and behavioral diagnostics. (a) Three-run mean terminal success with sample standard deviation over the 14 base tasks. (b) Micro-averaged subtask completion from one annotated run. (c) Pooled majority-vote diagnostics for the five strongest conditions. Panels (a) and (b) include all 12 models; Appendix Table 10 gives complete diagnostics.

4.2 Interaction and Failure Analysis

Figure 2 shows that failures emerge after basic task recognition. The top three conditions reach 100.0% target localization and at least 99.6% manipulation semantics, yet spatial-relation scores remain only 70.4–78.9%. Active exploration is also below ceiling, while fine-grained pre-contact analysis remains 73.9–77.9% across the top three. Thus, identifying the target and understanding the intended operation do not ensure sufficiently accurate robot–object geometry for contact-level action. These diagnostics complement rather than replace terminal success.

Table 2: VA-Bench base-task success (%) for 12 primary model conditions. Entries are mean ± sample standard deviation across three runs; each run macro-averages the 11 single-arm, three dual-arm, or all 14 tasks.

Subset	Qwen3.8 max	Opus 5	GPT-5.6 sol	GPT-5.6 terra	Doubao 2.1	Gemini 3.6
Single-arm	**65.61±3.81**	64.09±3.18	59.55±5.06	29.24±1.39	19.09±0.45	18.03±0.26
Dual-arm	11.11±2.55	11.67±1.67	**22.22±1.92**	0.56±0.96	0.00±0.00	0.00±0.00
All 14	**53.93±3.17**	52.86±2.79	51.55±4.31	23.10±1.25	15.00±0.36	14.17±0.21

Subset	GPT-5.6 luna	Sonnet 5	Qwen3.7 max	Qwen3.7 plus	MiniMax M3	Mimo v2.5
Single-arm	17.42±0.69	11.52±1.89	10.30±1.14	9.09±0.45	5.15±0.69	4.24±0.26
Dual-arm	1.11±0.96	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
All 14	13.93±0.71	9.05±1.49	8.10±0.90	7.14±0.36	4.05±0.55	3.33±0.21

Recovery reveals a second break in the closed loop. Several strong conditions detect visible errors at roughly 70%, but online correction remains only 26.6–46.7%; Gemini-3.6 instead shows the highest post-failure-adjustment rate despite near-zero online-correction behavior. Recognizing execution deviation is therefore substantially easier than using the resulting feedback to revise the next metric action. The three annotators agree unanimously on 95.7% of applicable labels (Fleiss $\kappa = 0.942$). Appendix Table 10 and Appendix C.10.1 provide complete scores and agreement analyses.

Together, these results show that strong task recognition and partial trajectory progress coexist with weaker robot–object grounding and outcome-conditioned correction. Section 5 next uses controlled interventions to separate the contributions of model-directed evidence acquisition, instance-specific geometric re-grounding, and demonstration conditioning.

5 CONTROLLED ANALYSES AND ABLATIONS

Beyond the primary benchmark, controlled interventions probe specific capabilities required by VA-BENCH. We first test active perception by comparing model-selected observations with fixed multi-view coverage. We then preserve the task procedure while changing action-relevant scene geometry to measure robustness under geometric shift, extending this analysis to long-horizon composition. Finally, we vary the demonstration-summary source to separate procedural information from downstream spatial reasoning and execution. Unless noted otherwise, these analyses use matched single runs; the three-run aggregation in Section 4 applies only to the primary base-task comparison.

5.1 TASK-DIRECTED ACTIVE PERCEPTION VS. PASSIVE MULTI-VIEW INPUT

We compare the canonical active protocol with two passive multi-view controls for Qwen3.8-max, GPT-5.6-sol, Opus-5, and Doubao-2.1. Conditions match the model checkpoint, model-specific summary, first-run setting, 14 task families with 20 seeds each, robot interface, executor, checker, scoring, context limit, and compaction policy. Active runs receive one current RGB image and may spend the shared episode budget on semantic or local camera actions. One passive control provides the same five predefined views jointly at each observation point; the other provides them sequentially in fixed order, with image-only inputs consuming no interaction steps. Both disable camera actions, while prior robot actions, planner outcomes, and textual feedback remain under the same context policy. The comparison therefore contrasts model-directed observation with fixed multi-view coverage. Appendices C.7 and C.8 specify the two controls; Table 3 summarizes overall success; Appendix Table 8 gives the arm-specific breakdown.

Table 3: Active and passive observation success (%). Each condition uses a matched single run over 280 episodes (14 tasks $\times$ 20 seeds). Active scores are from the first base-evaluation run, rather than the three-run means in Table 2. Both passive protocols use five predefined views: bundled presents them jointly, and sequential presents them in fixed order.

Model	Active	Passive (bundled)	Passive (sequential)
Qwen3.8-max	**57.50**	27.86	33.57
GPT-5.6-sol	**56.07**	33.93	38.93
Opus-5	**54.29**	30.36	33.57
Doubao-2.1	**15.00**	5.36	7.14

Camera-action usage. Across all four models, more than 99% of camera actions select one of the five predefined semantic viewpoints; incremental translation, zoom, yaw, and pitch together account for less than 1%. The active protocol therefore primarily tests when and which semantic view to select. Since presentation format and observation count also differ, the results support model-directed observation under this protocol without isolating the causal effect of viewpoint selection alone.

Table 3 shows that active observation outperforms both passive formats for all four models. Sequential presentation improves on bundled input but remains 7.86–23.93 percentage points below active observation. Thus, presenting views one at a time recovers part of the loss, while fixed multi-view coverage still does not reproduce task-directed active perception under our protocol.

The effect is also task dependent. Under bundled passive input, Qwen3.8-max falls from 85% to 5% on `place_cube_in_bowl` and from 60% to 10% on `place_single_cube`, whereas grasping tasks degrade less when the predefined views already expose the relevant object. Broader visual coverage alone therefore does not guarantee that the model receives the observations most useful for the current task.

5.2 CONTROLLED GEOMETRIC SHIFT AND LONG-HORIZON COMPOSITION

Controlled geometric shift. Seven single-arm task families reuse the same instruction and base-task demonstration summary while varying object geometry, container identity, or spatial layout, with additional distractors in some tasks. Background, lighting, camera configuration, and table height remain fixed. This track tests adaptation to changed scene geometry, including target selection among distractors, rather than isolating geometric estimation alone.

Qwen3.8-max retains 67.86% success with a 10.00-point drop, whereas GPT-5.6-sol falls from 80.00% to 47.86%, a 32.14-point decrease. With the procedure and interface unchanged, these drops reveal sensitivity to the changed spatial configuration, including distractor-related demands where present. Smaller drops for weaker conditions occur at substantially lower performance floors. Appendix Tables 14 and 15 give complete results.

On the shifted scenes, the descriptive spatial-perception module remains at or above 99.3%, while robot manipulation ranges from 54.3% to 86.7%. Thus, the performance drop is not explained by target recognition alone: strong observable perception can coexist with difficulty converting changed scene geometry into reliable manipulation. Appendix Table 16 gives the complete diagnostic breakdown.

Long-horizon composition. We further test whether competence on demonstrated atomic skills composes over a longer horizon. A separate 20-episode track requires five-object placement from four independently demonstrated atomic skills, without a full-task demonstration. No tested model completes a strict episode. Qwen3.8-max and GPT-5.6-sol nevertheless achieve 61/100 and 53/100 successful object placements, respectively, indicating substantial local progress that does not compose into reliable inventory-conditioned completion.

5.3 Demonstration-Summary Source Control

We vary only the demonstration-summary source (Table 4). The human reference, synthesized from task-level successes and failures, raises mean success from 62.00% to 75.33%. It is a feedback-informed reference, not a human–model comparison under identical video-only inputs. Shared model summaries help some readers but hurt others. Construction details, gains, and per-task results appear in Appendix C and Tables 17 and 19.

Table 4: Success (%) by summary source. Each cell pools three tasks with 20 seeds each in a matched single run. Own uses the evaluated model's summary; Qwen, GPT, and Opus denote Qwen3.8-max, GPT-5.6-sol, and Opus-5 summaries. Mean includes all five evaluated models for every source.

Evaluated model	Own	Qwen	GPT	Opus	Human
Qwen3.8-max	86.67	86.67	85.00	88.33	**95.00**
Opus-5	85.00	56.67	78.33	85.00	**91.67**
GPT-5.6-sol	73.33	21.67	73.33	76.67	**88.33**
GPT-5.6-terra	41.67	10.00	41.67	55.00	**63.33**
Gemini-3.6	23.33	10.00	35.00	35.00	**38.33**
Mean	62.00	37.00	62.67	68.00	**75.33**

Appendix A.6 additionally evaluates four locally deployed models under a fixed Qwen3.8-max procedural interface.

The scope and limitations of the behavioral diagnostics, geometric-shift setting, control architecture, simulation protocol, and ablation coverage are discussed in Appendix B.

6 Conclusion

VA-Bench evaluates whether general-purpose MLLMs can construct the spatial knowledge needed for action by actively acquiring evidence, grounding it into instance-specific metric robot–object geometry, and revising actions from execution feedback. Across 14 task families and three runs, the best macro-average success is $53.93 \pm 3.17\%$: strong target localization and manipulation semantics coexist with weaker spatial relations and online correction. Controlled analyses reveal two further failure patterns. First, bundled and sequential five-view controls underperform model-directed observation across all four tested models; even sequential presentation remains 7.86-23.93 points below active control. Second, preserving task procedure does not ensure robust geometric re-grounding: under matched geometry changes, success drops by as much as 32.14 points. Summary-source controls further show that procedural information quality affects performance without eliminating downstream failures. Together, these results show that broad passive coverage and procedural recognition are insufficient for reliable, self-directed metric spatial grounding.

AI USE STATEMENT

Generative AI tools were used solely to polish the language and presentation of this manuscript, including grammar, phrasing, and readability improvements. They were not used to generate experimental data, design the benchmark or evaluation protocol, implement the evaluation methods, or replace the authors' scientific judgment. The authors reviewed all AI-assisted text and take full responsibility for the accuracy, claims, results, and final content of this paper. The MLLMs evaluated by VA-Bench are experimental subjects in the study and are separate from this author-side language-polishing disclosure.

ETHICS STATEMENT

All experiments in this work are conducted in physics-based simulation and do not involve deployment on physical robots. Human involvement includes annotation of simulated robot trajectories for the behavioral diagnostics and construction of the human-authored procedural constraints used in the summary-source control. The annotations concern observable model and robot behavior, and the annotation records contain no personal or sensitive information about the annotators. Model-generated actions are executed only within the simulated benchmark through a fixed interface with bounded command ranges, workspace constraints, and motion-planning checks. VA-Bench is intended to evaluate spatial reasoning and manipulation behavior under controlled simulation conditions; performance on the benchmark should not be interpreted as evidence of safety or reliability for real-world robot deployment. Such deployment would require separate validation of perception, control, physical safety, failure recovery, and environment-specific risks beyond the scope of this work.

REPRODUCIBILITY STATEMENT

We provide the benchmark and evaluation details intended to support protocol-level reproduction of our experiments. Section 3 specifies the evaluation tracks, interactive episode protocol, demonstration-conditioned re-grounding, model-visible information, action interfaces, execution boundary, and outcome measures. Appendix A describes the evaluation conditions, seed usage, aggregation procedures, and detailed results. Appendix C gives the benchmark specification, including task coverage, seed construction and verification, task randomization and terminal predicates, robot and camera action spaces, passive-view controls, demonstration conditioning, and trajectory annotation. Appendix D describes the shared model-agnostic agent harness, model interaction protocol, observation and action interfaces, context handling, and prompt conditioning. Together, these sections specify the benchmark protocol and evaluation pipeline in a form intended to support protocol-level reproduction. For models accessed through externally maintained services, exact numerical reproduction may additionally depend on provider-side model and service versions.

REFERENCES

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. ScanQA: 3D question answering for spatial scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19129–19139, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.025.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. SpatialBot: Precise spatial understanding with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9490–9498. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11128671.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian 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, 2024.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. SpatialRGPT: Grounded spatial reasoning in vision-language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 135062–135093. Curran Associates, Inc., 2024. doi: 10.52202/079017-4293.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. PaLM-E: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488. PMLR, 2023.

Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. EmbSpatial-Bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 346–355. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-short.33.

Kuan Fang, Fangchen Liu, Pieter Abbeel, and Sergey Levine. MOKA: Open-world robotic manipulation through mark-based visual prompting. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.062.

Zhiyuan Feng, Zhaolu Kang, Qijie Wang, Zhiying Du, Jiongrui Yan, Shubin Shi, Chengbo Yuan, Huizhi Liang, Yu Deng, Qixiu Li, Rushuai Yang, Arctanx An, Leqi Zheng, Weijie Wang, Shawn Chen, Sicheng Xu, Yaobo Liang, Jiaolong Yang, and Baining Guo. Seeing across views: Benchmarking spatial reasoning of vision-language models in robotic scenes. In *International Conference on Learning Representations*, 2026.

Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971. doi: 10.1037/h0031619.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 148–166. Springer, 2024. doi: 10.1007/978-3-031-73337-6_9.

Hongcheng Gao, Hailong Qu, Jingyi Tang, Jiahao Wang, Zihao Huang, Hengkang Qiao, Shihong Huang, Junming Yang, Yi Li, Hongyixuan Yuan, et al. SpatialWorld: Benchmarking interactive spatial reasoning of multimodal agents in real-world tasks. *arXiv preprint arXiv:2606.09669*, 2026.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, et al. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090.

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. RVT: Robotic view transformer for 3D object manipulation. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 694–710. PMLR, 2023.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

Kilem Li Gwet. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48, 2008. doi: 10.1348/000711006X126600.

Yining Hong, Jiageng Liu, Han Yin, Manling Li, Leonidas Guibas, Fei-Fei Li, Jiajun Wu, and Yejin Choi. ESI-Bench: Towards embodied spatial intelligence that closes the perception-action loop. *arXiv preprint arXiv:2605.18746*, 2026.

Siyuan Huang, Haonan Chang, Yuhan Liu, Yimeng Zhu, Hao Dong, Abdeslam Boularias, Peng Gao, and Hongsheng Li. A3VLM: Actionable articulation-aware vision language model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1675–1690. PMLR, 2025a.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Fei-Fei Li. VoxPoser: Composable 3D value maps for robotic manipulation with language models. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 540–562. PMLR, 2023a.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782. PMLR, 2023b.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Fei-Fei Li. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4573–4602. PMLR, 2025b.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6700–6709, 2019. doi: 10.1109/CVPR.2019.00686.

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, et al. Do as I can, not as I say: Grounding language in robotic affordances. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 2023.

Hassan Jaber, Refinath S N, Luca Cagliero, Christopher E. Mower, and Haitham Bou-Ammar. EmbodimentSemantic: A spatial scene-graph dataset and benchmark for vision-language models on embodied manipulation trajectories. *arXiv preprint arXiv:2607.00020*, 2026.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. doi: 10.1109/LRA.2020.2974707.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: Zero-shot task generalization with robotic imitation learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 991–1002. PMLR, 2022.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Fei-Fei Li, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: Robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14975–15022. PMLR, 2023.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Fei-Fei Li, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2901–2910, 2017. doi: 10.1109/CVPR.2017.215.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9161–9175. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.568.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, et al. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. BEHAVIOR-1K: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 80–93. PMLR, 2023.

Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. ManipLLM: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18061–18070, 2024a.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. In *International Conference on Learning Representations*, 2024b.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023. doi: 10.1109/ICRA48891.2023.10160591.

Yuan-Hong Liao, Rafid Mahmood, Sanja Fidler, and David Acuna. Reasoning paths with reference objects elicit quantitative spatial reasoning in large vision-language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17028–17047. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.947.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023a. doi: 10.52202/075280-1939.

Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023b. doi: 10.1162/tacl_a_00566.

Mengzhen Liu, Enshen Zhou, Cheng Chi, Yi Han, Shanyu Rong, Liming Chen, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. SaPaVe: Towards active perception and manipulation in vision-language action models for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37164–37174, 2026a.

Tianhui Liu, Jie Feng, Zhiheng Zheng, Shengyuan Wang, Yiming Guo, Yanxin Xi, Hangyu Fan, Yong Li, and Pan Hui. SpatialAct: Probing spatial reasoning-to-action capabilities of VLM agents in 3D scenes. *arXiv preprint arXiv:2605.31148*, 2026b.

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.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. SQA3D: Situated question answering in 3D scenes. In *International Conference on Learning Representations*, 2023.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Fei-Fei Li, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 2022.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1820–1864. PMLR, 2023.

Anurag Maurya, Sukhvansh Jain, Prajwal Avhad, Gautham Balachandran, Ziyi Zhou, Atharva Kshirsagar, Satyam Singh, Bowen Li, Rishabh Mukund, Ritul Singh, et al. IMBench: A benchmark for intuitive robotic manipulation. *arXiv preprint arXiv:2607.15641*, 2026.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. doi: 10.1109/LRA.2022.3180108.

Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. ManiSkill: Generalizable manipulation skill benchmark with large-scale demonstrations. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of household tasks for generalist robots. In *Proceedings of Robotics: Science and Systems*, 2024a. doi: 10.15607/RSS.2024.XX.050.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. PIVOT: Iterative visual prompting elicits actionable knowledge for VLMs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37321–37341. PMLR, 2024b.

Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Sünderhauf. SayPlan: Grounding large language models using 3D scene graphs for scalable robot task planning. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 23–72. PMLR, 2023.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 894–906. PMLR, 2022.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 785–799. PMLR, 2023.

Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. RoboSpatial: Teaching spatial understanding to 2D and 3D vision-language models for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15768–15780, 2025.

Venkatesh Sripada, Samuel Carter, Frank Guerin, and Amir Ghalamzan. AP-VLM: Active perception enabled by vision-language models. *arXiv preprint arXiv:2409.17641*, 2024.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 87310–87356. Curran Associates, Inc., 2024. doi: 10.52202/079017-2771.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Fei-Fei Li, Danfei Xu, Yuke Zhu, and Anima Anandkumar. MimicPlay: Long-horizon imitation learning by watching human play. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 201–221. PMLR, 2023.

Xin Wu, Zhixuan Liang, Yue Ma, Mengkang Hu, Zhiyuan Qin, and Xiu Li. ST-BiBench: Benchmarking multi-stream multimodal coordination in bimanual embodied tasks for MLLMs. *arXiv preprint arXiv:2602.08392*, 2026.

Yuling Xi, Haokai Zhang, Muzhi Zhu, Hao Zhong, Zongze Du, Hengyu Zhao, Chenchen Jing, Yufei Yin, Bin Qin, Yongjie Yang, Zhenbo Luo, Hao Chen, and Chunhua Shen. Metric-Bench: Exploring in-context spatial metric reasoning in VLMs for indoor scenes. In *Computer Vision – ECCV 2026*, pp. 284–301. Springer, 2026.

Haoyu Xiong, Xiaomeng Xu, Jimmy Wu, Yifan Hou, Jeannette Bohg, and Shuran Song. Vision in action: Learning active perception from human demonstrations. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5450–5463. PMLR, 2025.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Fei-Fei Li, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10632–10643, 2025a.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. EmbodiedBench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70576–70631. PMLR, 2025b.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2020.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. RoboPoint: A vision-language model for spatial affordance prediction in robotics. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4005–4020. PMLR, 2025.

Jiyao Zhang, Mingxu Zhang, Yitong Peng, Haoxuan Liu, Chenshuo Wang, Yuxing Long, Haoyang Huang, Dongjiang Li, Nan Duan, Hui Shen, and Hao Dong. Embodied3DBench: Benchmarking low-level embodied spatial intelligence of vision language models. *arXiv preprint arXiv:2605.29074*, 2026a.

Pingyue Zhang, Zihan Huang, Yue Wang, Jieyu Zhang, Letian Xue, Zihan Wang, Qineng Wang, Keshigeyan Chandrasegaran, Ruohan Zhang, Yejin Choi, et al. Theory of Space: Can foundation models construct spatial beliefs through active exploration? In *International Conference on Learning Representations*, 2026b.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, et al. VLABench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11142–11152, 2025.

Enyu Zhao, Vedant Raval, Hejia Zhang, Jiageng Mao, Zeyu Shangguan, Stefanos Nikolaidis, Yue Wang, and Daniel Seita. ManipBench: Benchmarking vision-language models for low-level robot manipulation. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3413–3462. PMLR, 2025.

Yanpeng Zhao, Wentao Ding, Hongtao Li, Baoxiong Jia, and Zilong Zheng. ESPIRE: A diagnostic benchmark for embodied spatial reasoning of vision-language models. *arXiv preprint arXiv:2603.13033*, 2026.

Kaizhi Zheng, Xiaotong Chen, Odest Chadwicke Jenkins, and Xin Wang. VLMbench: A compositional benchmark for vision-and-language manipulation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 665–678. Curran Associates, Inc., 2022. doi: 10.52202/068431-0048. Datasets and Benchmarks Track.

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and Shanghang Zhang. RoboRefer: Towards spatial referring with reasoning in vision-language models for robotics. In *Advances in Neural Information Processing Systems*, volume 38, pp. 32209–32286, 2025. doi: 10.52202/085713-0957.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023.

A Complete Evaluation Results

This appendix reports the disaggregated values underlying the compact tables in the main paper. All tables remain in the normal reading orientation. The per-task base-benchmark matrix is split by model group, and the nine behavioral diagnostics are split by module. Model columns and rows follow descending 14-task macro-average; bold blue cells mark the best score within each directly comparable task row or metric column, including ties.

A.1 Base-evaluation settings and aggregation

Conditions and validity. The primary comparison evaluates 12 models, each using a summary it produced from the task's RGB-only expert video. Shared-summary controls reuse another condition's summary and are reported separately in Section 5. The summary-source control uses summaries from Qwen3.8-max, GPT-5.6-sol, Opus-5, or human experts, while the local-model comparison uses a fixed Qwen3.8-max summary. These control outcomes are not pooled with the primary conditions.

Following the instance construction and verification protocol in Section 3.1, each independent run uses the same 20 retained seeds for each of the 11 single-arm and three dual-arm families. Three runs give 60 outcomes per task and 840 run–seed outcomes per primary condition. Offline verification signals remain evaluator-only, and terminal success is determined solely by the environment checker. All models share the observation, action, budget, and executor contract in Section 3; harness details appear in Appendix D.

Terminal success. Let $y_{m,r,\tau,i} \in \{0,1\}$ be the checker outcome for model m, run r, task τ, and seed i. We average over seeds within a task and then macro-average over task families:

$$S_{m,r,\tau} = \frac{1}{20}\sum_{i=1}^{20} y_{m,r,\tau,i}, \qquad S^{\text{all}}_{m,r} = \frac{1}{14}\sum_{\tau=1}^{14} S_{m,r,\tau}. \tag{2}$$

We report the mean and sample standard deviation of the three run-level macro-averages. Single- and dual-arm scores apply the same procedure over 11 and three families, respectively. Every family has equal weight, and the error bars measure run-to-run variation, not binomial uncertainty over seeds.

Annotated run and partial progress. Diagnostics and subtask progress use the first complete base run for every primary condition, totaling 3,360 episodes. Each diagnostic pools applicable episode-level majority-vote labels across the single- and dual-arm subsets. Figure 2 plots these percentages directly, without averaging individual annotator votes or applying min–max normalization.

Subtask completion is a micro-average over scored checkpoint instances, with 420 single-arm and 140 dual-arm checkpoints per model. Because its weighting unit is a checkpoint rather than a task family, it measures intermediate progress but is not a relaxed upper bound on terminal success. Appendix C.10 defines observable checkpoint evidence and diagnostic applicability. Appendix C.10.1 reports agreement and sensitivity analyses.

A.2 Base-task success rates

Tables 5–7 report the three-run mean and sample standard deviation for each of the 14 base tasks and 12 primary model conditions. Each run evaluates the same 20 retained seeds meeting the acceptance criteria in Appendix C.4; the final row aggregates each run over all 14 tasks before computing summary statistics. Steps is the maximum interaction horizon. Doubao-2.1 and Gemini-3.6 abbreviate `doubao-seed-2.1-turbo` and `gemini-3.6-flash`, respectively.

Table 5: Base-task success rates (%, mean ± sample SD), model group 1 of 3.

Task	Steps	Qwen3.8 max	Opus 5	GPT-5.6 sol	GPT-5.6 terra
Single-arm tasks					
beat_block_hammer_right	100	3.33±2.89	8.33±2.89	**15.00±5.00**	6.67±2.89
click_bell_right	50	**73.33±5.77**	68.33±14.43	65.00±15.00	1.67±2.89
grasp_pen_leaning_cube	50	**61.67±2.89**	35.00±5.00	45.00±18.03	25.00±5.00
grasp_single_bottle	50	**91.67±2.89**	75.00±0.00	76.67±11.55	51.67±5.77
grasp_single_bottle_upright	50	65.00±5.00	78.33±16.07	70.00±10.00	68.33±2.89
grasp_single_cube	50	**83.33±10.41**	**83.33±11.55**	76.67±17.56	31.67±5.77
grasp_single_pen	50	**91.67±10.41**	63.33±15.28	28.33±16.07	48.33±7.64
place_cube_in_bowl	50	81.67±2.89	**88.33±10.41**	85.00±10.00	18.33±2.89
place_cube_on_cube	200	86.67±7.64	78.33±2.89	**88.33±12.58**	46.67±2.89
place_single_bottle_upright	100	**40.00±5.00**	38.33±2.89	13.33±2.89	6.67±5.77
place_single_cube	100	43.33±14.43	88.33±7.64	**91.67±7.64**	16.67±2.89
Dual-arm tasks					
place_shoe	100	33.33±7.64	35.00±5.00	**66.67±5.77**	0.00±0.00
handover_horizontal_block	100	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
lift_pot	50	0.00±0.00	0.00±0.00	0.00±0.00	**1.67±2.89**
Macro-average		**53.93±3.17**	52.86±2.79	51.55±4.31	23.10±1.25

Table 6: Base-task success rates (%, mean ± sample SD), model group 2 of 3.

Task	Steps	Doubao 2.1	Gemini 3.6	GPT-5.6 luna	sonnet 5
Single-arm tasks					
beat_block_hammer_right	100	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
click_bell_right	50	6.67±5.77	13.33±2.89	0.00±0.00	11.67±2.89
grasp_pen_leaning_cube	50	6.67±2.89	10.00±5.00	18.33±2.89	5.00±8.66
grasp_single_bottle	50	86.67±2.89	38.33±2.89	3.33±5.77	33.33±5.77
grasp_single_bottle_upright	50	48.33±2.89	78.33±2.89	**86.67±5.77**	40.00±10.00
grasp_single_cube	50	10.00±5.00	8.33±2.89	15.00±5.00	16.67±2.89
grasp_single_pen	50	26.67±10.41	23.33±7.64	8.33±2.89	1.67±2.89
place_cube_in_bowl	50	6.67±7.64	5.00±5.00	6.67±7.64	11.67±2.89
place_cube_on_cube	200	8.33±2.89	10.00±5.00	13.33±2.89	0.00±0.00
place_single_bottle_upright	100	6.67±2.89	11.67±2.89	26.67±7.64	0.00±0.00
place_single_cube	100	3.33±2.89	0.00±0.00	13.33±2.89	6.67±2.89
Dual-arm tasks					
place_shoe	100	0.00±0.00	0.00±0.00	3.33±2.89	0.00±0.00
handover_horizontal_block	100	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
lift_pot	50	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
Macro-average		15.00±0.36	14.17±0.21	13.93±0.71	9.05±1.49

Table 7: Base-task success rates (%, mean ± sample SD), model group 3 of 3.

Task	Steps	Qwen3.7 max	Qwen3.7 plus	MiniMax M3	Mimo v2.5
Single-arm tasks					
beat_block_hammer_right	100	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
click_bell_right	50	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
grasp_pen_leaning_cube	50	1.67±2.89	3.33±2.89	0.00±0.00	0.00±0.00
grasp_single_bottle	50	1.67±2.89	18.33±2.89	0.00±0.00	0.00±0.00
grasp_single_bottle_upright	50	76.67±7.64	76.67±2.89	46.67±2.89	41.67±10.41
grasp_single_cube	50	20.00±10.00	0.00±0.00	10.00±5.00	5.00±8.66
grasp_single_pen	50	13.33±2.89	0.00±0.00	0.00±0.00	0.00±0.00
place_cube_in_bowl	50	0.00±0.00	1.67±2.89	0.00±0.00	0.00±0.00
place_cube_on_cube	200	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
place_single_bottle_upright	100	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
place_single_cube	100	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
Dual-arm tasks					
place_shoe	100	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
handover_horizontal_block	100	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**	**0.00±0.00**
lift_pot	50	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
Macro-average		8.10±0.90	7.14±0.36	4.05±0.55	3.33±0.21

Table 8: Controlled single-run comparison of active evidence acquisition and two passive five-view input formats. Each condition reuses its own base-evaluation summary, the same 20 seeds per family, and the same executor, budget, and checker. Entries are success count / total episodes followed by success rate. Sequential denotes one image per input in fixed view order; image inputs do not consume task steps in that condition.

Model	Protocol	Single-arm (11×20)	Dual-arm (3×20)	All 14 (280)
Qwen3.8-max	Active	154/220 (70.00%)	7/60 (11.67%)	161/280 (57.50%)
	Passive	77/220 (35.00%)	1/60 (1.67%)	78/280 (27.86%)
	Sequential	94/220 (42.73%)	0/60 (0.00%)	94/280 (33.57%)
GPT-5.6-sol	Active	143/220 (65.00%)	14/60 (23.33%)	157/280 (56.07%)
	Passive	92/220 (41.82%)	3/60 (5.00%)	95/280 (33.93%)
	Sequential	101/220 (45.91%)	8/60 (13.33%)	109/280 (38.93%)
Opus-5	Active	144/220 (65.45%)	8/60 (13.33%)	152/280 (54.29%)
	Passive	82/220 (37.27%)	3/60 (5.00%)	85/280 (30.36%)
	Sequential	90/220 (40.91%)	4/60 (6.67%)	94/280 (33.57%)
Doubao-2.1	Active	42/220 (19.09%)	0/60 (0.00%)	42/280 (15.00%)
	Passive	15/220 (6.82%)	0/60 (0.00%)	15/280 (5.36%)
	Sequential	20/220 (9.09%)	0/60 (0.00%)	20/280 (7.14%)

A.3 BASE-TASK BEHAVIORAL DIAGNOSTICS

Tables 11–13 give the single- and dual-arm components behind the pooled scores in Table 10. Each pooled value combines valid counts from the two subsets. All four tables report the same twelve primary conditions, each using its first fully annotated base run. Every component-table entry is a percentage. TL: target perception and localization; AE: informative active exploration; SR: robot–object spatial relations; MS: manipulation-semantics understanding; MP: goal-directed manipulation planning; FG: fine-grained pre-contact analysis, scored over every reported episode; ED: error detection; OC: online correction; PF: post-failure adjustment. Annotators marked an assessable ambiguity in every base-task episode, so AE is applicable to all 280 annotated episodes per condition in this track; the error-recovery criteria retain their majority-voted applicability gate. For ED, OC, and PF, an episode with no observable error is recorded as NA and excluded from the denominator. When an error occurs, all three criteria are applicable and receive an explicit positive or negative judgment.

Table 9: Per-task successes out of 20 seeds for the controlled active evidence acquisition and two passive five-view input formats, first base-evaluation run. Each condition reuses its own demonstration summary and the same seeds; robot execution and terminal success/failure accounting are identical. The active protocol permits camera actions, the passive protocol supplies one five-image bundle, and the sequential protocol supplies the same views one at a time in fixed order without charging image inputs as task steps. A/P/S denotes active/passive/sequential.

	Qwen3.8-max			GPT-5.6-sol			Opus-5			Doubao-2.1		
Task	A	P	S	A	P	S	A	P	S	A	P	S
Single-arm tasks												
`click_bell_right`	16	19	18	16	14	12	12	10	8	0	3	2
`grasp_pen_leaning_cube`	13	11	12	5	2	3	8	6	4	1	0	0
`grasp_single_bottle`	19	12	11	18	11	14	15	13	10	17	7	10
`grasp_single_bottle_upright`	14	8	10	16	13	13	12	7	6	10	2	4
`grasp_single_cube`	19	7	11	19	15	14	18	11	15	3	0	1
`grasp_single_pen`	16	11	13	7	6	7	16	13	14	7	3	3
`place_cube_in_bowl`	17	1	5	19	17	13	20	8	12	0	0	0
`place_cube_on_cube`	19	5	8	20	14	13	16	6	10	2	0	0
`place_single_bottle_upright`	9	0	2	3	0	2	8	2	2	1	0	0
`place_single_cube`	12	2	4	17	0	10	18	6	7	1	0	0
`beat_block_hammer_right`	0	1	0	3	0	0	1	0	2	0	0	0
Single-arm total (/220)	154	77	94	143	92	101	144	82	90	42	15	20
Dual-arm tasks												
`place_shoe`	7	1	0	14	3	8	8	3	4	0	0	0
`handover_horizontal_block`	0	0	0	0	0	0	0	0	0	0	0	0
`lift_pot`	0	0	0	0	0	0	0	0	0	0	0	0
Dual-arm total (/60)	7	1	0	14	3	8	8	3	4	0	0	0
All-task total (/280)	161	78	94	157	95	109	152	85	94	42	15	20

Table 10: Pooled base-task trajectory-level behavioral diagnostics (%) for all twelve primary conditions. Scores combine applicable single- and dual-arm episodes; error-free episodes are excluded from ED, OC, and PF.

Condition	TL	AE	SR	MS	MP	FG	ED	OC	PF
Qwen3.8-max	**100.0**	79.3	78.9	**100.0**	**90.7**	73.9	**73.6**	**46.7**	42.6
Opus-5	**100.0**	**83.2**	70.4	99.6	90.0	**77.9**	69.9	32.4	49.1
GPT-5.6-sol	**100.0**	72.9	76.1	**100.0**	84.6	74.6	70.8	27.3	56.5
GPT-5.6-terra	**100.0**	55.4	69.6	99.6	78.2	52.9	72.6	26.6	60.5
Doubao-2.1	96.4	31.8	83.9	**100.0**	54.6	16.8	49.0	14.1	39.5
Gemini-3.6	**100.0**	31.4	85.4	**100.0**	73.2	8.9	63.8	1.9	**61.9**
GPT-5.6-luna	**100.0**	40.0	71.8	99.3	61.1	38.9	49.2	13.7	37.4
sonnet-5	99.3	21.8	80.7	**100.0**	31.8	32.1	10.6	1.2	9.8
Qwen3.7-max	96.4	13.6	81.4	97.5	35.7	2.5	32.6	0.4	32.2
Qwen3.7-plus	97.9	10.7	82.9	94.6	36.8	2.9	17.7	1.1	17.0
MiniMax-M3	95.7	6.1	**93.6**	95.4	21.4	1.8	22.6	1.4	21.1
Mimo-v2.5	83.2	5.4	71.8	98.9	19.6	2.1	25.8	0.0	26.2

Table 11: Base-task spatial perception and understanding diagnostics (%) for all twelve primary conditions.

	Single-arm			Dual-arm		
Condition	TL	AE	SR	TL	AE	SR
Qwen3.8-max	**100.0**	85.0	89.1	**100.0**	**58.3**	41.7
Opus-5	**100.0**	**90.5**	71.4	**100.0**	56.7	66.7
GPT-5.6-sol	**100.0**	81.8	81.8	**100.0**	40.0	55.0
GPT-5.6-terra	**100.0**	64.1	70.0	**100.0**	23.3	68.3
Doubao-2.1	95.5	34.1	85.5	**100.0**	23.3	78.3
Gemini-3.6	**100.0**	36.8	89.1	**100.0**	11.7	71.7
GPT-5.6-luna	**100.0**	47.7	69.5	**100.0**	11.7	80.0
sonnet-5	99.5	25.5	76.8	98.3	8.3	**95.0**
Qwen3.7-max	99.1	16.4	83.6	86.7	3.3	73.3
Qwen3.7-plus	99.1	13.2	84.1	93.3	1.7	78.3
MiniMax-M3	96.4	7.3	**94.1**	93.3	1.7	91.7
Mimo-v2.5	82.7	5.5	69.5	85.0	5.0	80.0

Table 12: Base-task robot manipulation diagnostics (%) for all twelve primary conditions.

	Single-arm			Dual-arm		
Condition	MS	MP	FG	MS	MP	FG
Qwen3.8-max	**100.0**	94.1	75.5	**100.0**	**78.3**	**68.3**
Opus-5	99.5	**94.5**	**86.4**	**100.0**	73.3	46.7
GPT-5.6-sol	**100.0**	90.0	76.4	**100.0**	65.0	**68.3**
GPT-5.6-terra	99.5	86.4	58.2	**100.0**	48.3	33.3
Doubao-2.1	**100.0**	61.4	15.0	**100.0**	30.0	23.3
Gemini-3.6	**100.0**	78.2	8.6	**100.0**	55.0	10.0
GPT-5.6-luna	99.1	63.2	45.0	**100.0**	53.3	16.7
sonnet-5	**100.0**	33.6	40.0	**100.0**	25.0	3.3
Qwen3.7-max	96.8	43.2	3.2	**100.0**	8.3	0.0
Qwen3.7-plus	93.2	43.6	3.6	**100.0**	11.7	0.0
MiniMax-M3	94.1	21.8	2.3	**100.0**	20.0	0.0
Mimo-v2.5	98.6	25.0	2.7	**100.0**	0.0	0.0

Table 13: Base-task error recovery diagnostics (%) for all twelve primary conditions. Error-free episodes are NA and excluded before computing each percentage.

	Single-arm			Dual-arm		
Condition	ED	OC	PF	ED	OC	PF
Qwen3.8-max	**81.7**	**54.2**	43.0	52.7	**27.3**	41.8
Opus-5	79.4	36.9	55.6	42.9	19.6	30.4
GPT-5.6-sol	76.9	28.8	60.6	53.6	23.2	44.6
GPT-5.6-terra	76.7	34.4	60.8	**59.3**	1.7	**59.3**
Doubao-2.1	53.2	18.2	40.9	35.0	0.0	35.0
Gemini-3.6	65.5	2.5	**63.0**	58.3	0.0	58.3
GPT-5.6-luna	49.0	17.3	34.2	50.0	1.7	48.3
sonnet-5	9.3	1.5	8.2	15.0	0.0	15.0
Qwen3.7-max	29.1	0.5	28.6	45.0	0.0	45.0
Qwen3.7-plus	12.4	1.4	11.5	36.7	0.0	36.7
MiniMax-M3	22.8	1.8	21.0	21.7	0.0	21.7
Mimo-v2.5	32.4	0.0	32.9	1.7	0.0	1.7

A.4 Controlled transfer and long-horizon diagnostics

Table 14: Controlled geometric transfer and composition. Base and held-out values average seven matched families (20 seeds each); Δ is held-out minus base. Long reports successful object placements out of 100 possible and strict episode success (%) over 20 episodes.

	Matched transfer (%)			Long horizon	
Condition	Base	Held-out	Δ	Objects	Success
Qwen3.8-max	77.86	**67.86**	−10.00	**61/100**	**0.00**
GPT-5.6-sol	**80.00**	47.86	−32.14	53/100	**0.00**
GPT-5.6-terra	35.00	27.86	−7.14	5/100	**0.00**
GPT-5.6-luna	22.86	18.57	−4.29	2/100	**0.00**

Table 15: Single-arm controlled-transfer success rates (%). Each entry uses 20 held-out seeds and reuses the corresponding base-task demonstration summary.

Task family	Qwen3.8 max	GPT-5.6 sol	GPT-5.6 terra	GPT-5.6 luna
`grasp_single_bottle`	**85**	80	60	20
`grasp_single_bottle_upright`	40	35	50	**55**
`grasp_single_cube`	**80**	50	40	15
`place_cube_in_bowl`	**70**	40	5	20
`place_cube_on_cube`	**80**	65	5	10
`place_single_bottle_upright`	**55**	20	15	10
`place_single_cube`	**65**	45	20	0
Macro-average	**67.86**	47.86	27.86	18.57

Table 16 summarizes the same held-out trajectories at the module level. These equal-weight means are descriptive profiles rather than an overall ranking.

Table 16: Overall module-level diagnostic overview (%) under controlled held-out transfer. SPU, RM, and ER equally average TL/AE/SR, MS/MP/FG, and ED/OC/PF, respectively.

Condition	SPU	RM	ER
Qwen3.8-max	99.8	**86.7**	59.8
GPT-5.6-sol	99.3	76.4	59.9
GPT-5.6-terra	**100.0**	66.2	**60.2**
GPT-5.6-luna	**100.0**	54.3	34.4

Table 17: Shared demonstration-summary sources. Pooled success (%) covers three tasks with 20 seeds each; *Own* uses each reader's summary and Δ is shared minus own. For model-generated sources, *avg* Δ excludes the source model; the human-authored condition averages all five readers.

		On Qwen3.8-max		On GPT-5.6-sol		On Opus-5		On human-authored summary	
Evaluated model	Own	Rate	Δ	Rate	Δ	Rate	Δ	Rate	Δ
Qwen3.8-max	86.67	86.67	0.00	85.00	−1.67	88.33	+1.67	**95.00**	+8.33
Opus-5	85.00	56.67	−28.33	78.33	−6.67	85.00	0.00	**91.67**	+6.67
GPT-5.6-sol	73.33	21.67	−51.67	73.33	0.00	76.67	+3.33	**88.33**	+15.00
GPT-5.6-terra	41.67	10.00	−31.67	41.67	0.00	55.00	+13.33	**63.33**	+21.67
Gemini-3.6	23.33	10.00	−13.33	35.00	+11.67	35.00	+11.67	**38.33**	+15.00
avg	62.00	37.00	−31.25	62.67	+0.83	68.00	+7.50	**75.33**	+13.33

A.5 Demonstration-summary source ablation

This ablation changes only the source of the task-level procedural summary. The evaluated model, task instruction, scene seeds, image observations, available actions, action budget, executor, context policy, terminal checker, and scoring rule are held fixed. The primary (own-summary) condition follows the standard pipeline: the evaluated model inspects multiple distinct RGB frames from the successful expert video and writes its own summary before acting in a new scene. Shared-summary conditions replace only that text with a summary generated by a named source model. The source supplies procedural context, but no actions, object poses, target coordinates, contact labels, task waypoints, or solution for the live scene.

The human-authored condition uses a common set of expert constraints written for the same demonstration tasks. These constraints record the visible task-level requirements needed to interpret the procedure, including the grasp or contact region, approach direction and posture, arm assignment and coordination order where relevant, and completion conditions. They are provided as the same textual reference to every reader. They contain no machine-readable trajectory, metric object geometry, privileged state, or seed-specific action parameters. This is therefore a procedural-reference control that preserves the downstream grounding and execution problem.

The human reference was synthesized by reviewing the task-level successes and failures across the evaluated models and retaining constraints that consistently matter for execution. In addition to the visible grasp or contact strategy, the reference records approach direction, end-effector posture, height and depth requirements, arm assignment, coordination order, and terminal completion

conditions where applicable. Recurrent failure-sensitive details that are often omitted from model-generated summaries are stated explicitly, but only at the qualitative task level. The reference contains no seed-specific object coordinates, privileged simulator state, machine-readable action sequence, or live-scene solution, and the same text is supplied to all five readers.

The matched Qwen3.7-plus control decreases from 7.14% with its own summary to 5.00% with the Qwen3.8-max summary. In the three-task cross-reader comparison, Qwen3.8-max summaries change the other readers' success by -31.25 points on average, GPT-5.6-sol summaries by $+0.83$ points, and Opus-5 summaries by $+7.50$ points. The human-authored reference improves every reader: the five-reader mean rises from 62.00% to 75.33% ($+13.33$ points), while the reader ordering remains Qwen3.8-max, Opus-5, GPT-5.6-sol, GPT-5.6-terra, and Gemini-3.6. These results separate the benefit of stronger shared procedural information from reader-dependent downstream grounding and execution ability.

Table 18 gives the subset breakdown for the matched Qwen3.7-plus comparison reported above.

Table 18: Effect of demonstration-summary source for Qwen3.7-plus. Columns are macro-averages over 11 single-arm tasks, three dual-arm tasks, and all 14 tasks, respectively.

Summary source	Single-arm	Dual-arm	All 14
Qwen3.7-plus (model-specific)	**9.09**	**0.00**	**7.14**
Qwen3.8-max (shared)	6.36	**0.00**	5.00
Shared − model-specific	−2.73	0.00	−2.14

Table 19 gives the per-task breakdown of the cross-model shared-summary comparison summarized in Section 5.3. Each entry is the success rate over the same 20 verified seeds used in the corresponding first base-evaluation run; only the demonstration summary is replaced. The own-summary block of each summary author reproduces its base run exactly.

Shared summaries come from Qwen3.8-max, GPT-5.6-sol, and Opus-5, with an additional human-authored summary as a reference condition. These summary sources are read by five models on `grasp_single_bottle`, `grasp_single_pen`, and `place_cube_in_bowl`. Each pooled cell contains 60 episodes. Model checkpoint, 20 verified seeds per task, executor, action budget, terminal checker, and all other settings match the reader's first base-evaluation run.

Table 19: Per-task success rate (%) under shared model-generated or human-authored demonstration summaries, 20 seeds per cell. Tasks are B = `grasp_single_bottle`, P = `grasp_single_pen`, C = `place_cube_in_bowl`. *Own* is the model's own model-specific summary in its first base-evaluation run.

	Own			On Qwen3.8-max			On GPT-5.6-sol			On Opus-5			On human-authored summary		
Evaluated model	B	P	C	B	P	C	B	P	C	B	P	C	B	P	C
Qwen3.8-max	**95**	80	85	**95**	80	85	90	**80**	85	**95**	**85**	85	**95**	**90**	**100**
Opus-5	75	**80**	**100**	55	**40**	**75**	70	75	**90**	75	80	**100**	**85**	**90**	**100**
GPT-5.6-sol	90	35	95	30	15	20	**90**	35	**95**	90	45	95	**95**	**70**	**100**
GPT-5.6-terra	55	55	15	20	5	5	65	30	30	60	65	40	**70**	**75**	**45**
Gemini-3.6	40	30	0	15	15	0	60	30	15	50	35	20	**55**	**40**	**20**
Mean	71.00	56.00	59.00	43.00	31.00	37.00	75.00	50.00	63.00	74.00	62.00	68.00	**80.00**	**73.00**	**73.00**

A.6 Locally deployed small-model results

Table 21 gives the task-level results summarized in Table 20. Each task contains 20 trajectories. Gemma-4-31B-it and the two Qwen3.6 variants use the Qwen3.8-max demonstration summary, while Qwen3.8-27B uses a mixed-summary condition whose summary source is also Qwen3.8-max. Without summary guidance, Gemma-4-31B-it and the two Qwen3.6 variants obtain 0.00% on every task.

Table 20: Summary-guided success for locally deployed models. Entries are success count / total trajectories followed by the success rate. The single-arm, dual-arm, and overall denominators are 220, 60, and 280, respectively.

Condition	Summary	Single-arm	Dual-arm	All 14
Qwen3.8-27B	Qwen3.8-max	**27/220 (12.27%)**	0/60 (0.00%)	**27/280 (9.64%)**
Gemma-4-31B-it	Qwen3.8-max	24/220 (10.91%)	0/60 (0.00%)	24/280 (8.57%)
Qwen3.6-27B	Qwen3.8-max	3/220 (1.36%)	0/60 (0.00%)	3/280 (1.07%)
Qwen3.6-35B-A3B	Qwen3.8-max	2/220 (0.91%)	0/60 (0.00%)	2/280 (0.71%)

Table 21: Per-task success rates (%) for summary-guided locally deployed models. Each task contains 20 trajectories. Bold blue marks the highest nonzero result in each row.

Task	Gemma-4 31B-it	Qwen3.6 27B	Qwen3.6 35B-A3B	Qwen3.8 27B
Single-arm tasks				
`click_bell_right`	**10.00**	0.00	0.00	**10.00**
`grasp_pen_leaning_cube`	5.00	0.00	0.00	**10.00**
`grasp_single_bottle`	**30.00**	0.00	5.00	15.00
`grasp_single_bottle_upright`	55.00	15.00	5.00	**85.00**
`grasp_single_cube`	5.00	0.00	0.00	**10.00**
`grasp_single_pen`	**15.00**	0.00	0.00	5.00
`place_cube_in_bowl`	0.00	0.00	0.00	0.00
`place_cube_on_cube`	0.00	0.00	0.00	0.00
`place_single_bottle_upright`	0.00	0.00	0.00	0.00
`place_single_cube`	0.00	0.00	0.00	0.00
`beat_block_hammer_right`	0.00	0.00	0.00	0.00
Dual-arm tasks				
`place_shoe`	0.00	0.00	0.00	0.00
`handover_horizontal_block`	0.00	0.00	0.00	0.00
`lift_pot`	0.00	0.00	0.00	0.00
Single-arm macro-average	10.91	1.36	0.91	**12.27**
Dual-arm macro-average	0.00	0.00	0.00	0.00
All-14 macro-average	8.57	1.07	0.71	**9.64**

B Scope and Limitations

B.1 Interpretation of the Behavioral Diagnostics

The nine trajectory-level criteria are intended to characterize observable behavioral patterns and to help localize where difficulties arise during interaction. They complement terminal task success by distinguishing behaviors related to spatial perception, manipulation, and recovery. Because these criteria are based on executed trajectories, they should be interpreted as behavioral diagnostics rather than direct measurements of the model's internal reasoning process. More fine-grained attribution among perception, planning, numerical grounding, and execution would require additional component-specific interventions. The present diagnostics therefore provide a practical decomposition of observed behavior while leaving such causal attribution to future analysis.

B.2 Scope of Controlled Geometric Transfer

The controlled-transfer experiments evaluate adaptation to changed spatial configurations while preserving the demonstrated task procedure. Held-out variants modify object geometry, container or box instances, or spatial layout, and some tasks additionally introduce distractor objects. Background, lighting, camera configuration, and table height remain fixed, but clutter does not remain fixed in the distractor variants. Consequently, the measured changes can reflect both geometric regrounding and target-selection demands; this track does not isolate their individual contributions. It is also not a comprehensive test of open-set generalization across arbitrary objects, robots, cameras, or real-world conditions.

The long-horizon evaluation similarly focuses on one compositional task family, and the local-model study covers four deployments under summary-guided conditions. These experiments provide targeted evidence for the capabilities studied here; evaluating broader forms of robustness and generalization would require additional object families, nuisance variations, and deployment settings.

B.3 Scope of the Control Architecture

VA-BENCH evaluates a direct-control setting in which the MLLM produces metric robot and camera commands through a fixed, model-agnostic harness. This design makes it possible to study spatial grounding and action generation without introducing a separately learned low-level policy into the primary evaluation.

An alternative is a modular architecture in which an MLLM performs high-level reasoning or instruction generation (Driess et al., 2023; Ichter et al., 2023; Huang et al., 2023b; Liang et al., 2023; Rana et al., 2023) and a separately trained vision-language-action policy translates those outputs into low-level robot actions. We do not evaluate that decomposition in the present work. The reported results therefore characterize the direct-control interface used in VA-BENCH, rather than the relative advantages of alternative VLM–VLA system designs.

A controlled comparison between these settings would require a common VLA controller, a clearly specified instruction interface, and matched training, execution, and calibration conditions. We view this as a complementary extension for studying how spatial reasoning responsibilities are distributed across modular embodied-agent architectures.

B.4 Simulation Evaluation Scope

All experiments are conducted in physics-based simulation using a fixed, model-agnostic executor, bounded command ranges, workspace checks, and task-specific simulator success predicates. This controlled setting enables consistent evaluation across models while isolating the spatial grounding and interaction capabilities targeted by VA-BENCH from hardware-specific sources of variation.

The present evaluation does not model several factors that arise in physical robot systems, including sensor noise, camera-calibration error, hardware variation, actuation latency, and discrepancies between simulated and physical contact dynamics. Accordingly, the reported results characterize performance under the benchmark's simulation protocol rather than direct real-robot transfer. Extending the evaluation to physical platforms would provide a complementary test of how the measured capabilities interact with real-world perception, calibration, control, and execution effects.

B.5 Scope of the Controlled Ablations

The primary base-task comparison uses three independent runs for all 12 model conditions. The additional controlled analyses isolate specific factors, including observation protocol, geometric regrounding, demonstration conditioning, and long-horizon composition. Within the available evaluation time and budget, these analyses use matched single-run settings and, in some cases, selected subsets of models or task families.

Within each comparison, the variables not targeted by the intervention are matched as specified in the corresponding protocol. These controlled settings therefore allow the analyses to focus on the intervention of interest, while not providing the same run-to-run variability estimates as the three-run primary evaluation. We interpret them as targeted diagnostic comparisons under the reported conditions rather than as exhaustive estimates across all models, tasks, and repeated runs. Broader replication across additional runs and conditions would further characterize the stability and generality of these effects.

C Benchmark Specification Details

VA-Bench evaluates whether a general-purpose MLLM can turn partial visual evidence into spatially grounded robot behavior. In the canonical protocol, each episode couples active observation, cross-view reasoning, numerical end-effector control, and physics-simulated task completion. The benchmark does not ask the model to select an answer or a reference trajectory. Instead, the model must determine what evidence is missing, acquire an informative view, infer object–robot geometry, and instantiate that inference as a sequence of metric actions. The camera interface deliberately isolates task-conditioned viewpoint selection from physical camera-arm control, while the trajectory rubric reports behavioral evidence rather than direct measurements of latent cognitive states. The matched passive controls instead supply the same five predefined RGB views without exposing a

camera-action command. One control sends the five views as a single image bundle; the other sends them one at a time in a fixed order. Robot actions, seeds, execution, scoring, and context-budget settings remain the same. In the sequential control, image-only inputs do not consume an interaction step; only robot actions advance the task budget. Prior robot actions, planner outcomes, and textual feedback remain in the retained history, subject to the same context limit and compaction policy as the active condition. Thus the comparison isolates both active evidence acquisition and the format of fixed multi-view input, rather than robot control or history retention.

C.1 Interactive task formulation

The following formulation describes the canonical active-view protocol. An evaluation instance is

$$z = (\tau, \xi, g_{\tau,\xi}, D_\tau, B_\tau, \phi_\tau), \tag{3}$$

defined by a task family τ, a scene seed ξ, a natural-language goal $g_{\tau,\xi}$, an optional observation-only demonstration D_τ, an episode budget B_τ, and a physics-based task-success predicate ϕ_τ. Before test-time control, the designated summary model converts D_τ into a condition-specific textual summary $s_\tau^{(c)}$; the primary condition uses the evaluated model itself, whereas shared-summary controls use the named source model. The simulator maintains an unobserved physical state x_t. At step t, the model receives

$$o_t = (I_t, p_t, h_t, f_{t-1}), \tag{4}$$

where I_t is the current active-camera RGB image, p_t is filtered robot proprioception, h_t is the interaction history, and f_{t-1} contains the previous action-validity and trajectory-planning outcomes. At each decision, the MLLM may call a non-environment inspection or demonstration tool, stop, or choose one camera or robot action. Let $\mathcal{A}_{\text{robot}}$ and $\mathcal{A}_{\text{cam}}$ denote the robot and camera action sets, respectively:

$$a_t \sim M_\theta\left(g_{\tau,\xi}, s_\tau^{(c)}, o_{\le t}, a_{<t}\right), \qquad \begin{cases} x_{t+1} = \mathcal{T}_\tau(x_t, a_t), & a_t \in \mathcal{A}_{\text{robot}}, \\ x_{t+1} = x_t, & a_t \in \mathcal{A}_{\text{cam}}. \end{cases} \tag{5}$$

and observes the resulting scene before choosing again. Robot actions update the physical state, whereas camera actions update only the observation pose. Tool calls and stop decisions do not advance the environment; camera and robot actions consume the same task-dependent budget. Indexing dispatched environment actions by t, an episode ends at $t_{\text{end}} \le B_\tau$ when $\phi_\tau(x_{t_{\text{end}}})$ becomes true, the model stops, or the budget is exhausted. The primary outcome is therefore binary physics-simulated task success, rather than agreement with an annotated answer or action sequence.

For the bundle passive control, the single actively controlled image I_t is replaced by a fixed-order bundle of five predefined RGB views, and $\mathcal{A}_{\text{cam}}$ is removed from the model's available action set. For the sequential passive control, the same five views are sent as separate image-only inputs in the same fixed order. These image-only inputs do not advance the task budget; the budget advances only when the model dispatches a robot action. In both passive controls, prior robot actions, planner outcomes, and textual feedback remain in a history governed by the same context limit and compaction policy as the canonical evaluation. The other observation fields, robot action space, seeds, executor, success predicate, and scoring are identical. The model cannot select, reorder, or request an additional view in either passive control.

Every RGB frame in the observation payload is clean: model-facing images contain no drawn keypoints, coordinate axes, masks, or target markers. Proprioception contains the controlled end-effector position and orientation, gripper open/closed state, the physical finger midpoint (GC), and the current gripper-local axes expressed in the world frame. In dual-arm episodes, these quantities are provided separately for both arms. The model retains a persistent history of prior robot actions, planner outcomes, and textual feedback, bounded by the same context limit and compaction policy in both conditions. The passive API supplies only the current five-view bundle and cannot obtain additional views. The model is not given depth, point clouds, segmentation, object coordinates or poses, contact state, oracle grasp points, task waypoints, or numerical camera calibration. Consequently, the numerical robot state anchors a metric reference frame by design; the benchmark does

not test recovery of the robot coordinate system itself. The task-relevant object geometry and its relation to that frame must still be inferred from RGB and interaction.

C.2 Task-level spatial demands

At the task-design level, VA-Bench spans seven spatial–action demands that become jointly observable only when perception is coupled to action (Du et al., 2024).

Active evidence acquisition. The initial observation is not assumed to expose every task-relevant dimension. The model must decide whether uncertainty concerns world-XY position, height, depth, orientation, clearance, or occlusion, and select a viewpoint that is diagnostic for that uncertainty. Because observation draws on the shared action budget, the model must also stop looking and act once the accumulated evidence is sufficient (Sripada et al., 2024).

Cross-view consistency and online state tracking. Camera motion changes image coordinates but not the world-frame robot controls. The model must reconcile observations across viewpoints, retain object identity and spatial relations, and invalidate stale evidence after either the camera, gripper, or object moves. After execution, it must interpret the new image and planner outcome relative to its previous action rather than restarting from an independent frame.

Metric object–robot grounding. The model must translate qualitative visual relations into signed world-axis translations and numerical distances. This requires relating an object seen in pixels to GC and end-effector coordinates, distinguishing camera-frame appearance from world-frame motion, and estimating whether the remaining lateral, vertical, and depth offsets are within grasp or placement tolerance (Song et al., 2025; Cai et al., 2025; Zhou et al., 2025; Xi et al., 2026).

Orientation, affordance, and contact geometry. Successful interaction depends on more than object centering. The model must infer long axes, surfaces, support relations, graspable regions, functional parts, and approach corridors. Examples include orienting the gripper closing axis across a bottle or pen, grasping a hammer by its handle while aligning the head with a block, pressing the top center of a bell, and preserving an upright container during placement.

Goal-directed metric action sequencing and outcome-conditioned revision. The model must compose approach, grasp, lift, transport, insertion, release, and retreat phases using only incremental metric commands. Low-level inverse kinematics, collision checking, and trajectory generation remain delegated to the fixed execution layer. Re-observation after each action enables online correction when a proposed translation is rejected, an object moves unexpectedly, a grasp misses, or a placement is not yet stable. The trajectory-level diagnostics measure the resulting recovery behavior at the level of observable actions (Appendix B).

Embodiment and bimanual coordination. Tasks with two available arms require reasoning about reachability, arm–object assignment, shared-workspace clearance, and temporal dependencies between roles. Handover tasks require complementary grasp regions and ordered release, whereas pot lifting requires mirrored handle geometry and synchronized motion.

Demonstration-conditioned spatial re-grounding. The demonstration communicates visible task procedure, not the live scene's metric solution. The model must extract transferable constraints and re-ground them under a new seed, object pose, arm assignment, or held-out geometry. This separates recognizing a demonstrated procedure from executing it in the current spatial configuration.

These seven demands are design dimensions that describe what the task suite requires; they are not seven additional scores or independently identified latent variables. Section C.10 instead operationalizes executed behavior through nine scored trajectory-level diagnostics in three modules. The two levels are intentionally not one-to-one: for example, bimanual coordination is a task condition and demonstration-conditioned re-grounding is evaluated through the transfer protocol. End-to-end outcomes also depend on numerical action generation, the fixed motion planner, and simulated dynamics.

C.3 TASK-FAMILY COVERAGE

VA-Bench comprises 14 base task families: eleven expose a single controllable arm, while three expose two arms and require the model to reason explicitly about their use. In `place_shoe`, the model must select and control the arm on the object's reachable side; `handover_horizontal_block` and `lift_pot` additionally require coordinated use of both arms. We therefore classify all three as dual-arm tasks, regardless of whether both arms move simultaneously.

In terms of task demands, the single-arm suite tests target pose and long-axis inference, object–robot alignment, grasp-region and approach-direction selection, and placement constraints such as containment, stacking, pose preservation, and release height. Bell pressing and hammering further test functional-part localization, contact direction, and tool geometry. The dual-arm suite requires reachability-aware arm assignment, ordered handover, or synchronized motion. The separately evaluated five-object task tests long-horizon state maintenance, repeated spatial re-grounding, and composition of atomic skills. These demands describe task coverage rather than additional independent scores.

Instances and controlled splits. A scene seed selects task-specific object positions, orientations, and, where configured, asset model IDs. Resulting layouts or arm assignments follow from these task fields rather than from a separate global randomizer. The base benchmark uses 20 retained seeds per task under the verification protocol in Appendix C.4. The primary terminal-success evaluation repeats this fixed suite in three independent runs per model; probe-only simulator state and actions are never exposed to the evaluated model. Seven base families additionally have paired held-out variants: horizontal and upright bottle grasping, cube grasping, cube-in-bowl placement, cube stacking, upright-bottle placement, and cube placement. Each variant reuses its base-task demonstration while changing controlled object geometry, container or box-like instance, or spatial layout; some variants also introduce distractor objects. The five-object basket task is a separate compositional track conditioned on four atomic-skill videos rather than a full-task demonstration.

C.4 SEED CONSTRUCTION AND PHYSICAL VERIFICATION

Candidate task families are retained only when they expose a visually recoverable spatial bottleneck and support direct control, a deterministic success checker, and observable subtask checkpoints.

Deterministic candidate instantiation. Before environment initialization, each integer seed fixes the NumPy and Torch random states. Evaluation uses `demo_clean`: global background, lighting, camera, table-height, and clutter randomization are disabled, leaving only the task-coded object position, orientation, and model-ID fields listed below. Consequently, a fixed task implementation and seed reproduce the same scene. The task-dependent `candidate_seeds` field is an ordered pool, not a declaration that every listed candidate was instantiated. Configured pools use task-dependent integer ranges 1000–1099, 1000–1119, 1000–1199, 1000–1299, or 1000–5000, but these bounds describe the available search order rather than the number of executed probes. Candidates are probed in order, and collection stops as soon as 20 accepted seeds have been obtained for a task.

Orientation-based settling and acceptance gate. Each instantiated scene is reset, advanced for 2,000 physics steps, and then observed for 500 additional steps. During the final 200 observation steps, explicit object-orientation drift must remain within approximately 3°. This is an orientation-based settling check; the current selector does not impose a separate translational-drift threshold. Initial and post-probe object poses must also be finite. A seed receives `ready=true` only when its privileged probe, the planner result, terminal checker, gripper state, and task-specific physical outcomes all pass. A probe run that ends before producing a physical outcome is recorded separately and does not by itself classify the scene as physically infeasible.

Verification scope. All 280 retained manifest entries have a matching `ready=true` probe under a single verification mode: a complete `task.play_once()` execution with the full planner chain, `planner_success`, the terminal checker, gripper state, and the task-specific physical outcome all passing. Every retained seed therefore carries a verified expert execution from the robot home state through the terminal predicate, so each base-task instance is certified physically solvable end to end rather than merely reachable at the grasp endpoint. Across all tasks, 318 candidates were

actually instantiated and probed before the per-task stopping rule was met, and 280 were retained. The resulting 88.1% is a conditional yield among candidates that were run, not an acceptance rate over the complete task-dependent candidate pools (Table 22).

Table 22: Conditional seed yield among candidates actually instantiated and probed. Selection visits each ordered pool only until 20 accepted seeds are collected, so unvisited pool entries are not part of the denominator.

Task or group	Probed	Retained	Yield (%)
Ten tasks whose first 20 probes passed	200	200	100.0
`place_cube_on_cube`	22	20	90.9
`place_single_cube`	22	20	90.9
`beat_block_hammer_right`	24	20	83.3
`lift_pot`	50	20	40.0
All 14 tasks	318	280	88.1

Task randomization and terminal predicates. Tables 23–26 report the task-coded scene fields and terminal checker conditions used by the base suite. Position ranges are in world-frame meters; tolerances are stated explicitly in millimeters. "Lifted" is measured relative to the corresponding initial object height. TCP denotes the tool-center point.

The shared target-placement predicate is implemented in `envs/_single_right_arm_place_target_task.py`; the remaining terminal predicates reside in their corresponding task environments. Deterministic initialization and the `demo_clean` policy are implemented in `envs/_base_task.py` and `task_config/demo_clean.yml`, while ordered candidate screening and both probe modes are implemented in `codex-robotwin-agent/scripts/select_safe_eval_seeds.py`.

C.5 Metric robot action space

The environment action space is

$$\mathcal{A} = \mathcal{A}_{\text{cam}} \cup \mathcal{A}_{\text{move}} \cup \mathcal{A}_{\text{rot}} \cup \mathcal{A}_{\text{grip}} \cup \mathcal{A}_{\text{dual}}. \tag{6}$$

Table 27 gives the exact robot-command parameterization. The interface exposes numerical Cartesian target changes rather than the legacy small/medium/large motion labels or low-level motor commands.

These action families map to `gripper.move_world`, `gripper.rotate_local`, and gripper aperture commands. Dual-arm episodes require `arm=left|right` for single-gripper commands and add the synchronized `dual_gripper` actions listed above. A fixed model-agnostic IK and trajectory layer attempts each metric target and returns binary planning feedback; it does not select the object, grasp region, arm role, or task waypoint. Section 3.2 defines this execution boundary.

C.6 Bounded active camera action space

The main evaluation exposes the camera actions in Table 28. Their semantic views are defined relative to GC, not to a privileged target-object location. Let c be the selected arm's finger midpoint (or the midpoint of both arms), and let the sign of the lateral offset follow the selected arm side. Each semantic-view action moves the camera toward the listed nominal offset and orients it toward c; the final pose is clamped to the camera workspace.

In single-arm tasks, the five semantic views target the active gripper. In dual-arm tasks, generic views target the two-arm midpoint, while `view_left_*`, `view_right_*`, and `view_both_*` explicitly select the observation target. Camera poses are clamped to a bounded workspace. The action changes only the observation viewpoint and never changes the world-frame meaning of robot actions.

We keep camera control discrete because continuous parameters would add control burden without making evidence acquisition more difficult. This interface deliberately evaluates task-conditioned

Table 23: Seed randomization and terminal predicates for direct-grasp tasks.

Task	Task-coded scene fields	Terminal success predicate
`grasp_single_bottle`	$x \in [0.20, 0.28]$, $y \in [-0.16, -0.04]$; yaw $\in [-22.5^\circ, 22.5^\circ]$	Target lifted > 50 mm; right gripper closed
`grasp_single_bottle_upright`	$x \in [0.20, 0.28]$, $y \in [-0.16, -0.04]$; fixed upright orientation	Target lifted > 50 mm; right gripper closed
`grasp_single_cube`	$x \in [0.18, 0.26]$, $y \in [-0.16, -0.06]$; yaw $\in [-45^\circ, 45^\circ]$	Target lifted > 50 mm; right gripper closed
`grasp_single_pen`	$x \in [0.18, 0.26]$, $y \in [-0.20, -0.10]$; yaw $\in [-30^\circ, 30^\circ]$	Target lifted > 50 mm; right gripper closed
`grasp_pen_leaning_cube`	Group $x \in [0.16, 0.22]$, $y \in [-0.24, -0.20]$; yaw $\in [-15^\circ, 15^\circ]$; fixed 30° lean	Pen lifted > 50 mm; right gripper closed

Table 24: Seed randomization and terminal predicates for placement and stacking tasks.

Task	Task-coded scene fields	Terminal success predicate
`place_cube_in_bowl`	Bowl $x \in [0.10, 0.14]$, $y \in [0, 0.04]$; cube $x \in [0.22, 0.27]$, $y \in [-0.17, -0.10]$; cube yaw $\in [-45^\circ, 45^\circ]$	Cube previously lifted ≥ 50 mm; center distance < 55 mm; relative height 15–95 mm; bowl displacement < 5 mm; contact; gripper open
`place_cube_on_cube`	Red cube $x \in [0.20, 0.26]$, $y \in [-0.14, -0.06]$; base cube $x \in [0.10, 0.14]$, $y \in [0, 0.04]$; yaw $\in [-45^\circ, 45^\circ]$	Red cube previously lifted ≥ 50 mm; top-surface overlap ≥ 2 mm; height error ≤ 12 mm; base displacement ≤ 12 mm; contact; gripper open
`place_single_cube`	Cube as in `grasp_single_cube`; target $x \in [0.10, 0.14]$, $y \in [0, 0.04]$	Cube previously lifted ≥ 50 mm; x/y error each ≤ 35 mm; height error ≤ 35 mm; target displacement ≤ 2 mm; contact; gripper open
`place_single_bottle_upright`	Bottle as in the upright-grasp task; target $x \in [0.03, 0.09]$, $y \in [0.07, 0.13]$	Bottle previously lifted ≥ 50 mm; x/y error each ≤ 40 mm; height error ≤ 35 mm; target displacement ≤ 2 mm; contact; gripper open

Table 25: Seed randomization and terminal predicates for functional-contact and tool-use tasks.

Task	Task-coded scene fields	Terminal success predicate
`click_bell_right`	$x \in [0.12, 0.26]$, $y \in [-0.18, -0.04]$; two bell model IDs	Gripper closed; contact point within 25 mm of the bell functional point on each of x and y, and within 30 mm on z
`beat_block_hammer_right`	Hammer $x \in [0.02, 0.06]$, $y \in [-0.10, -0.04]$; block $x \in [0.14, 0.22]$, $y \in [0.06, 0.13]$; yaw $\in [-0.5, 0.5]$ rad	Hammer lifted with contact by > 40 mm; functional-point x/y error each < 20 mm; physical hammer–block contact; gripper closed

Table 26: Seed randomization and terminal predicates for dual-arm tasks.

Task	Task-coded scene fields	Terminal success predicate
`place_shoe`	Shoe $x \in [-0.25, 0.25]$, $y \in [-0.10, 0.05]$; ten model IDs; samples with horizontal radius < 0.15 m excluded	x error < 50 mm; y error < 20 mm; absolute error of each quaternion component < 0.07; both grippers open
`handover_horizontal_block`	$x \in [-0.20, -0.15] \cup [0.15, 0.20]$, $y \in [-0.05, 0]$	Required handover state chain completed after vertical presentation; giver open and receiver closed; $z > 0.92$ m; object on receiver side; long axis within 20° of vertical
`lift_pot`	$x, y \in [-0.05, 0.05]$; yaw $\in [-22.5^\circ, 22.5^\circ]$; two pot model IDs	Pot $z > 0.82$ m; each TCP within 30 mm of its corresponding handle; upright dot product > 0.8

Table 27: Robot actions exposed to the MLLM. In dual-arm mode, every single-gripper numerical command must name the controlled arm.

Action family	Parameters	Semantics
World translation	$u \in \{x, y, z\}$, sign $\sigma \in \{-, +\}$, $d \in [1, 100]$ mm	Move the selected end effector by σd along a fixed world axis; the target is bounded by the robot workspace.
Local rotation	$r \in \{r_x, r_y, r_z\}$, sign σ, $\alpha \in [1, 90]^\circ$	Rotate about the selected current gripper-local axis using the right-hand rule; compensated waypoints keep GC approximately fixed.
Gripper aperture	`open`, `close`	Command the selected physical gripper without prescribing an object or contact state.
Synchronized dual translation	Independent (u, σ, d) for left and right	Plan and execute both world-axis targets in one RoboTwin move call.
Synchronized dual aperture	`dual_gripper.open` `dual_gripper.close`	Open or close both grippers in the same environment step.

Table 28: Camera actions in the active-view protocol. Semantic viewpoints are gripper-centered; incremental translations and rotations are expressed in the current camera frame unless stated otherwise.

Action	Motion	Primary geometric evidence
`view_topdown`	$c + (0, 0, 0.68)$ m	World-XY displacement, planar yaw, target coverage
`view_side`	$c + (\pm 0.58, -0.02, 0.06)$ m	Height, table clearance, vertical overlap and insertion
`view_front_side_45`	$c + (\pm 0.44, -0.44, 0.18)$ m	Break front/side occlusion and horizontal projection ambiguity
`view_side_top_45`	$c + (\pm 0.48, -0.02, 0.48)$ m	Joint evidence for lateral alignment, height, and finger placement
`view_oblique_45`	$c + (\pm 0.42, -0.42, 0.42)$ m	Overall 3D configuration and selection of the next diagnostic view
`move_left/right`		
`move_up/down`	Translate 5 cm along camera left/right/up/down	Local de-occlusion and small framing changes
`zoom_in/out`	Translate 8 cm along camera forward/backward	Inspect or widen the local interaction region
`yaw_left/right`	Rotate 10° around world Z	Horizontal parallax and side exposure
`pitch_up/down`	Rotate 10° around camera-left axis	Vertical parallax and top/side exposure

viewpoint selection and bounded camera adjustment, not physical camera-arm motion. Semantic views and incremental camera actions directly update the simulated camera pose; camera reachability, collision, continuous trajectory cost, and calibration error are outside the current scope. Each camera action still consumes an environment step, and the agent instruction limits camera-only chains to at most five consecutive actions. Table 28 is the complete set of camera commands available in every reported active-view run; the passive control condition of Appendix C.7 exposes none of them.

Observed camera-action usage. We classify each dispatched camera action as either a semantic-view selection or an incremental adjustment (translation, zoom, yaw, or pitch). Across the four models used in the matched active-versus-passive ablation, semantic-view selections account for more than 99% of camera actions for each model, and incremental adjustments account for less than 1%. Percentages are computed over dispatched camera actions; robot actions and stop decisions are excluded.

C.7 BUNDLED PASSIVE FIVE-VIEW OBSERVATION PROTOCOL

The matched passive comparison supplies five clean, predefined RGB images at each observation point, in place of the single actively acquired image used by the canonical condition. This bundled condition is evaluated for Qwen3.8-max, GPT-5.6-sol, Opus-5, and Doubao-2.1 under the same matched single-run setting. It exposes no camera-action command. The textual observation block appears first, followed by image blocks in the same fixed order `topdown`, `side`, `front_side_45`, `side_top_45`, and `oblique_45`. The local session record stores these view labels, but the evaluated OpenAI-compatible API request includes no additional textual label beside an image; the fixed block order is therefore the model-visible cue to view identity. The observer uses a 55° vertical field of view, a 0.05-m near plane, and a 100-m far plane. The images contain no end-effector or target marker, segmentation mask, depth map, or debugging overlay. These camera parameters define the renderer but are not sent to the model as numerical calibration.

The passive five-view bundle is supplied after reset and robot actions at the same control points where the canonical protocol supplies its current image. The active model may then select further views through camera actions, whereas the passive model has no such command. Its API observation contains only the current five-image bundle; prior robot actions, planner outcomes, and textual feedback remain under the same bounded-history and compaction policy as the canonical evaluation. Robot-action semantics and environment transitions are also matched. The comparison therefore contrasts model-directed active evidence acquisition with passively supplied multi-view evidence.

C.8 SEQUENTIAL FIXED-ORDER PASSIVE-VIEW CONTROL

The sequential passive control is evaluated for the same four models and uses exactly the same five rendered views and the same fixed order as the bundle passive control: `topdown`, `side`, `front_side_45`, `side_top_45`, and `oblique_45`. The difference is only in payload presentation. Instead of placing all five images in one API request, the evaluator sends one image per input in the prescribed order. The model cannot choose the view order, skip a view, request another view, or issue a camera action. These image-only inputs do not consume an environment step or the task action budget; the next budget-consuming event is the model's robot action. The resulting condition keeps visual content and order fixed while testing whether sequential presentation changes the use of the same passive evidence.

C.9 DEMONSTRATION-CONDITIONED EVALUATION PROTOCOL

For each standard task, the main protocol supplies one RGB-only video of a successful expert execution. Before evaluation, the evaluated MLLM retrieves multiple sampled frames and writes a structured textual summary of visible constraints, including grasp region, finger placement, approach, posture, arm assignment, coordination order, and completion conditions. The video contains visible robot and object motion, and therefore conveys procedural information, but exposes no machine-readable expert action, EEPose, joint state, object coordinate, contact label, or semantic phase. The summary is task-level experience; it does not contain the metric solution for a live seed. This task-level procedure video is distinct from the per-seed acceptance probes in Appendix C.4; those probe executions and their recorded trajectories remain evaluator-only and are never shown to the evaluated model. Demonstration interpretation and procedural abstraction are therefore part of the primary end-to-end construct, rather than fixed preprocessing.

For the primary model-specific-summary condition, this sequence is

$$D_\tau \xrightarrow[\text{multiple distinct RGB frames}]{\text{model inspection}} s_{m,\tau} \xrightarrow[\text{new seed}]{\text{re-grounding}} \{a_t, o_t\}_{t=0}^{t_{\text{end}}}. \tag{7}$$

At test time, the model receives the task instruction, generic safety and manipulation constraints, any task-specific semantic completion constraints, and its demonstration-derived summary. It must nevertheless reacquire the current scene, choose views, assign arms where applicable, and generate every numerical parameter. No oracle trajectory, target waypoint, target-object state, or task phase is supplied. The model-specific-summary condition is the primary end-to-end protocol. We additionally report shared-summary controls as controlled diagnostics that fix the textual procedural context across readers and therefore partially isolate downstream re-grounding and execution. For

the cross-model summary-source ablation, the reader, task, seed, visual observations, action interface, budget, executor, context policy, and checker are held fixed while only the summary text is replaced. The shared sources are summaries generated by Qwen3.8-max, GPT-5.6-sol, and Opus-5, plus a common human-authored set of expert constraints. The human constraints are constructed by reviewing the task-level successes and failures across the evaluated models and consolidating the constraints that consistently matter for execution. They describe the same task-level procedural content as a demonstration summary, including the contact or grasp region, approach direction, end-effector posture, height and depth requirements, arm assignment, coordination order, and completion conditions where applicable, as well as recurrent failure-sensitive details that model summaries tend to omit. They contain no metric scene solution, privileged state, machine-readable action sequence, or seed-specific parameters, and are supplied identically to all readers in that condition. Held-out variants reuse the base-task summary without a new video, while the composite basket task uses four independent atomic-skill videos and no full-task demonstration.

C.10 TRAJECTORY-BASED CAPABILITY DIAGNOSTICS

Terminal task success is deliberately strict, but it does not reveal whether a model failed to find the target, misunderstood the robot–object geometry, selected an unsuitable motion, or was unable to recover. We therefore augment the environment outcome with a trajectory-level diagnostic rubric. The rubric decomposes task competence into three first-level modules: *spatial perception and understanding*, *robot manipulation*, and *error recovery*. Each module has three separately scored trajectory-level behavioral criteria.

Evidence and annotation unit. The annotation unit is one complete episode. Each complete trajectory is independently reviewed by three annotators using the same episode record and rubric. Annotation proceeds in two stages: the three annotators first judge criterion applicability, and the majority decision fixes the episode-level gate; for every activated criterion, all three then record a binary behavioral judgment whose majority vote is the reported episode label. Annotators inspect the task instruction, the sequence of model-facing RGB observations, camera and robot actions, resulting scene changes, action validity, planner outcomes, and the terminal task state. The three independent judgments are retained in the annotation record, and the aggregation equations below operate on the episode-level labels used for reporting. Detailed positive and negative evidence rules and applicability conditions are provided in this appendix. Figure 3 gives a visual reference for the decision standard, with positive and negative trajectory examples for eight criteria and a scoring guide for FG. A model's textual rationale can help identify its stated intent, but does not by itself establish a capability: the executed trajectory must contain corresponding behavioral evidence. Conversely, a single planner rejection is not automatically a planning failure, because a geometrically reasonable target may still be infeasible for the fixed execution layer. The judgment instead considers whether the model subsequently changes its estimate or repeatedly issues actions that remain inconsistent with the visible task.

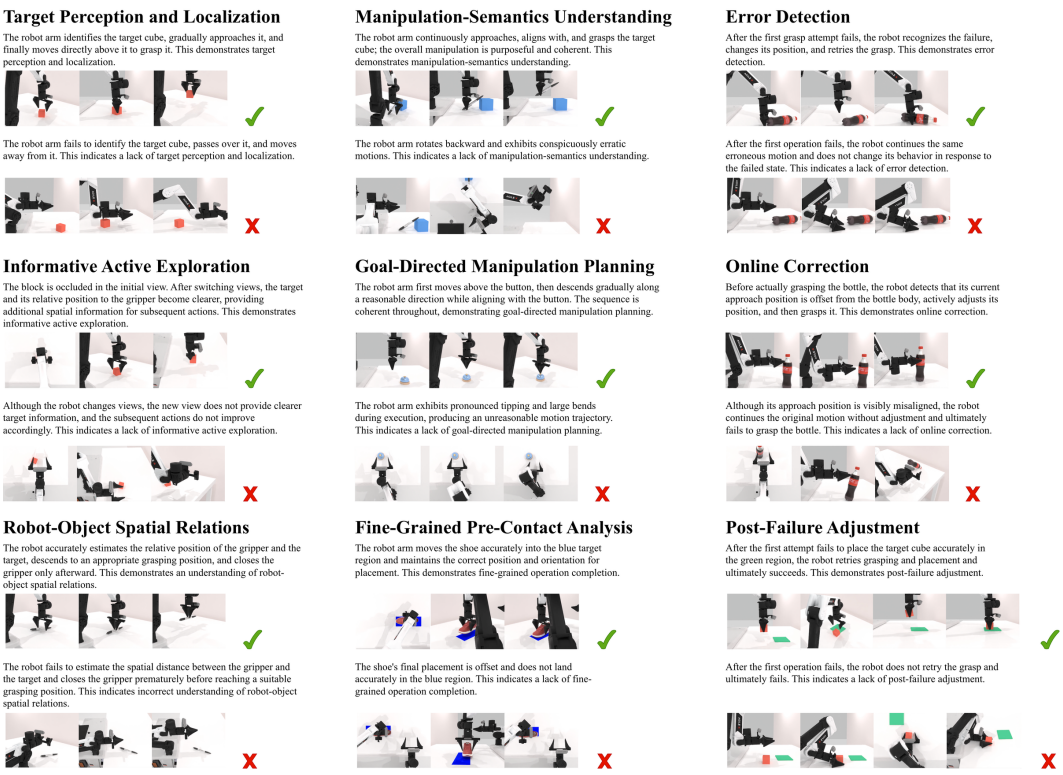


Figure 3: Annotation reference for the nine trajectory-level capability criteria. In the eight trajectory panels, the upper row is a representative positive trajectory (green check) and the lower row is a representative negative trajectory (red cross). The FG panel gives its pre-contact analysis rule; final placement alone does not establish this behavior. The examples illustrate observable evidence, not hidden model states. For error detection (ED), online correction (OC), and post-failure adjustment (PF), the examples contain an observable error; error-free episodes remain NA.

For annotator $a \in \{1, 2, 3\}$, let $g_e^{(a)} \in \{0, 1\}$ indicate whether episode e contains an observable execution error. The shared recovery gate is the majority vote

$$g_e = \mathbb{1}\left[\sum_{a=1}^{3} g_e^{(a)} \geq 2\right]. \tag{8}$$

If $g_e = 0$, ED, OC, and PF are all recorded as $\perp$ (reported as NA) and the episode is excluded from all three denominators. If $g_e = 1$, all three criteria are scored over the shared error-episode denominator. Each criterion receives positive credit only when its stage-specific behavioral evidence is observed; otherwise it is scored 0. For each activated criterion k, annotator a records $r_{e,k}^{(a)} \in \{0, 1\}$, where one denotes positive behavioral evidence and zero denotes that the behavior was not demonstrated. The episode-level label used for reporting is

$$r_{e,k} = \mathbb{1}\left[\sum_{a=1}^{3} r_{e,k}^{(a)} \geq 2\right]. \tag{9}$$

The same majority-vote rule is used for the binary labels of the other criteria. AE carries an analogous assessable-ambiguity gate, but in the annotated runs every episode was judged to contain a task-relevant ambiguity under single-view observation, so the gate never fired and AE is scored over all annotated episodes; TL, SR, MS, MP, and FG are applicable by construction. Thus, an episode is not penalized for having no error, but an applicable capability is scored zero when at least two annotators find no corresponding behavioral evidence.

Table 29: Inter-annotator agreement over the nine trajectory criteria. n is the number of applicable episode–criterion items (three judgments each); ED, OC, and PF are scored only on episodes containing an observable error, hence their smaller n. Unanimity is the fraction of items on which all three annotators agree; every remaining judgment is a 2–1 split. Positive rate is the fraction of majority-vote labels equal to one, and drives the difference between κ and AC1 for near-ceiling criteria. Statistics cover the annotated base run of the 12 primary conditions; the held-out-transfer and long-horizon annotations are excluded here and contained no split votes.

Module	Criterion	n	Unanimity	Pos. rate	Fleiss κ	Gwet AC1
SPU	TL	3,360	97.41%	0.974	0.723	0.982
SPU	AE	3,360	92.77%	0.376	0.898	0.908
SPU	SR	3,360	94.70%	0.789	0.892	0.948
RM	MS	3,360	99.88%	0.988	0.968	0.999
RM	MP	3,360	97.08%	0.565	0.960	0.962
RM	FG	3,360	87.38%	0.321	0.797	0.856
ER	ED	3,024	96.40%	0.447	0.951	0.953
ER	OC	3,024	99.40%	0.124	0.982	0.995
ER	PF	3,024	96.76%	0.369	0.953	0.960
All criteria		29,232	95.69%	0.559	0.942	0.943
All except FG		25,872	96.77%	0.589	0.956	0.958

C.10.1 Annotation reliability

Because the nine criteria are human-annotated, we report their agreement directly. The annotation record retains, for every activated criterion of every annotated episode, how many of the three annotators recorded positive evidence. Across the 3,360 annotated base episodes of the 12 primary conditions, this gives 29,232 applicable episode–criterion items and 87,696 individual binary judgments. Table 29 summarizes them. Terminal task success is determined solely by the environment checker and is never annotated, so annotation reliability bounds the interpretation of the nine diagnostics and of subtask progress, not of any success rate reported in this paper.

All 1,259 disagreements are 2–1 splits, and every reported label matches the majority vote. Overall unanimity is 95.69%, with Fleiss $\kappa = 0.942$ (Fleiss, 1971).

For criteria whose positive rate is far from balanced, κ understates agreement, because a skewed marginal inflates the chance-agreement term. TL is the clearest case: annotators agree unanimously on 97.41% of episodes, but the near-ceiling positive rate yields $\kappa = 0.723$. We also report Gwet AC1, which is less sensitive to prevalence, and interpret the two together. FG has the lowest unanimity (87.38%, $\kappa = 0.797$) and accounts for 424 of the split votes (Gwet, 2008). Its individual scores warrant particular care when comparing voting rules.

Robustness to the aggregation rule. Majority voting is one of several defensible ways to collapse three judgments. We recomputed all nine criteria for all 12 primary conditions under two alternatives: a strict rule that credits a criterion only when all three annotators are positive, and a permissive rule that credits it when any annotator is positive. Averaged over the nine criteria, the largest within-condition range across the three rules is 6.25 points. Relative to majority voting, the strict rule decreases this mean by at most 3.93 points and the permissive rule increases it by at most 2.33 points. The induced orderings are similar (Spearman $\rho \geq 0.979$ and Kendall $\tau \geq 0.909$ for every pairwise comparison). The three strongest conditions under majority voting remain the three strongest under both alternatives. Individual criteria are more sensitive: GPT-5.6-sol leads FG under unanimity (62.5%), whereas Opus-5 leads under majority voting (77.9%). Thus stability of the nine-criterion means does not establish stability of every individual score or ordering.

Annotations are not restatements of the outcome. A concern for any trajectory-level rubric is that annotators may simply infer the labels from whether the episode succeeded. The record contradicts this. On the 1,680 annotated episodes for which the per-episode file also records the checker verdict, FG agrees with terminal success on 82.08% of episodes. There are 116 episodes with FG$= 0$ and checker success, and 185 with FG$= 1$ and checker failure. FG measures pre-contact analysis, so a positive label does not require completion, and success alone does not establish such analysis. These counts do not identify the cause of individual mismatches. TL and MS behave in the opposite

way: they agree with terminal success on 31.1% and 30.7% of the same episodes, because more than $1{,}100$ episodes are labeled positive on target localization and manipulation semantics while still failing, and *no* episode is labeled negative on either criterion while succeeding. Annotators that copied the outcome could not produce this pattern. The same asymmetry is itself a finding: the visual and semantic prerequisites are usually satisfied, and the failures accumulate downstream.

What the record does not support. The retained counts are per-episode tallies rather than per-annotator labels, so statistics that require annotator identity, including pairwise Cohen κ, per-rater bias estimates, and Cochran's Q, are not recoverable and we do not report them. The concordance figures above are computed on the subset of annotation files that carry a machine-readable success column ($1{,}680$ of the $3{,}360$ annotated episodes); the agreement statistics in Table 29 use all $3{,}360$.

C.10.2 Spatial perception and understanding

Target perception and localization. This criterion records whether the executed behavior identifies the task-relevant object and localizes it well enough to guide subsequent behavior. Positive evidence includes end-effector motion directed toward the correct object, camera actions that keep the target in the region of interest, and later operations organized around that target. A negative judgment requires clear contrary evidence, such as sustained motion toward an unrelated object or workspace region, grasping a distractor, or repeatedly operating far from the target. An isolated exploratory motion is not sufficient to infer a localization failure.

Informative active exploration. This criterion records whether the executed behavior is consistent with actively acquiring and using visual evidence that resolves a task-relevant ambiguity. An exploration opportunity is annotated when the current observation is judged insufficient for a spatial or geometric fact needed by the next action, such as object orientation, depth, the contact region, the relative pose of two arms, or the visibility of a container opening. Uncertainty about whether an executed action achieved its intended effect is assessed under error detection rather than under this criterion.

An episode receives positive evidence only when all three pieces of behavioral evidence are present: (i) the model selects a semantic view or camera motion in response to the ambiguity; (ii) the resulting observation exposes information that reduces that ambiguity; and (iii) a subsequent action or state estimate is consistent with the new observation. A redundant or task-irrelevant viewpoint, or a view change whose observation does not reduce the ambiguity, is scored 0. Visiting an informative view and then immediately switching to an ineffective view without using the acquired evidence is also scored 0; a later irrelevant view does not erase positive evidence when the informative observation was already used. Episodes with no assessable task-relevant ambiguity would be marked $\perp$, but no annotated episode fell in this category: under single-view observation some task-relevant geometric fact was always judged underdetermined. The criterion measures trajectory evidence of evidence acquisition and use, not terminal task success.

Robot–object spatial relations (SR). This criterion records whether the trajectory is consistent with correct interpretation of relative height, distance, direction, and alignment between the gripper and the target. Positive evidence occurs when successive translations or rotations reduce the visible robot–object offset and it initiates contact, grasping, or release at a geometrically plausible state. Clear failures include closing the gripper while it remains visibly above or beside the object, descending at an incorrect lateral position, confusing image-plane direction with world-axis motion, or declaring alignment while a substantial separation remains.

C.10.3 Robot manipulation

Manipulation-semantics understanding. This criterion records whether the agent's behavior is consistent with correct use of the available control commands and selection of task-relevant actions. Positive evidence is a purposeful use of world-axis translation, local-axis rotation, aperture, or dual-arm commands to produce the intended physical effect. Evidently meaningless rotations, uncontrolled large motions, repeated open/close commands without a contact opportunity, or actions fundamentally unrelated to the task are inconsistent with the intended manipulation semantics. Nu-

merical imprecision alone is assessed under spatial understanding rather than automatically counted here; FG separately records pre-contact analysis.

Goal-directed manipulation planning. This criterion evaluates whether the agent constructs a coherent, goal-directed metric action sequence from the current state to the desired interaction state. Evidence includes an appropriate ordering of approach, alignment, grasp/contact, lift or actuation, transport, placement, release, and retreat, using only the phases required by the task. Intermediate actions should progressively reduce relevant spatial error or establish a necessary precondition. Large purposeless excursions, repeated movement in unrelated regions, an approach direction incompatible with the intended contact, or a sequence whose phases cannot lead to the goal constitute negative evidence.

Fine-grained pre-contact analysis (FG). This criterion records whether the model performs fine-grained spatial analysis as it approaches an intended contact. Relevant details include fingertip placement, gripper orientation and aperture, contact location, approach or placement height, and relative arm geometry. Positive evidence is a task-specific assessment of these details reflected in pre-contact decisions and actions. A generic instruction to manipulate carefully without corresponding behavior is insufficient. Every episode is scored; absence of such analysis, including failure to reach an opportunity for it, receives 0. The criterion does not require successful operation completion. A later miss, unsuitable release, or terminal failure does not by itself negate demonstrated pre-contact analysis, and terminal success alone does not establish it.

C.10.4 Error recovery

Error detection (ED). After an observable error occurs, ED is positive when subsequent behavior shows that the agent no longer treats the preceding action as successful. For example, it stops the current sequence, obtains another view, rechecks the grasp, or begins a materially different adjustment. Continuing the same plan while ignoring a visible miss, dropped object, unstable placement, or repeated planner rejection is negative evidence. Error-free episodes are NA for ED.

Online correction (OC). After an observable error occurs, OC is positive only when the agent notices and corrects a developing spatial deviation before it becomes an explicit operation failure. Examples include adjusting lateral alignment or gripper orientation before closing, re-centering an object during transport, and restoring bimanual levelness before the payload slips. If the error is allowed to progress into failure, or no timely corrective action is shown, OC is 0. Error-free episodes are NA for OC.

Post-failure adjustment (PF). Within an episode containing an observable error, PF is positive only if the error progresses to an explicit failure and the agent then re-evaluates the new state and changes its observation strategy, geometric hypothesis, approach, or numerical parameters before retrying. The recovery attempt need not ultimately complete the task: a clearly targeted, materially revised attempt is positive behavioral evidence. Repeating the failed action without a meaningful change, persisting with an invalid stale plan, or terminating despite a recoverable state is negative evidence. If no such post-failure adjustment is observed, PF is scored 0 under the shared error-episode denominator; error-free episodes are NA. OC and PF are stage-specific rather than mutually exclusive: one episode may receive positive credit for correcting a developing deviation and later making a separate post-failure adjustment.

Aggregation. Let

$$E_m = \{e : g_e = 1\}.$$

For each error-recovery criterion $k \in \{\mathrm{ED}, \mathrm{OC}, \mathrm{PF}\}$, the reported score is the positive rate over this shared error set,

$$C_{m,k} = \frac{\sum_{e \in E_m} \mathbb{1}[r_{e,k} = 1]}{|E_m|}, \tag{10}$$

where $|E_m|$ is the error-episode denominator used for the score. For the other six criteria, the denominator remains the number of episodes with a valid opportunity to assess that criterion. Each

first-level module is summarized by the equal-weight macro-average of its three constituent criterion scores; these module values are descriptive summaries, not an overall capability score. We retain all nine criterion scores in addition to the three module summaries so that a high value in one capability cannot conceal a distinct failure mode in another. We do not collapse all nine criteria into a single scalar because their applicability sets and statistical reliability differ. If a capability has no applicable episode, its score is reported as NA rather than zero and it is not silently included in a module average.

Subtask progress for complex tasks. For a multi-stage task, terminal success is supplemented with a task-specific checklist of observable subgoals, such as acquiring the correct object, completing a required orientation change or handover, reaching the destination, and establishing a stable final state. A subtask receives credit only when its required state transition is visible in the trajectory; merely issuing the intended command or obtaining a successful planner return is insufficient. Subtask completion is reported separately from terminal success and from the nine general criterion scores. Let $h_{m,\tau}$ and $n_{m,\tau}$ denote the completed and scored checkpoint instances, respectively, for model m on task τ. Across the base benchmark, we use the checkpoint-level micro-average

$$P_m^{\mathrm{sub}} = \frac{\sum_{\tau=1}^{14} h_{m,\tau}}{\sum_{\tau=1}^{14} n_{m,\tau}}. \tag{11}$$

Thus, each checkpoint instance is equally weighted, while tasks with more checklist stages contribute proportionally more instances. This preserves partial progress in long-horizon and bimanual tasks without relaxing the benchmark's environment-defined success criterion.

D Agent Harness Implementation Details

We evaluate every model through the same model-agnostic agent harness rather than attaching a learned robot policy or action head. The harness is an independent Python implementation of a persistent tool-using agent loop, specialized for interactive robot evaluation. It separates model inference, conversation state, action validation, environment execution, and run logging so that the MLLM is the only model-dependent component of the control stack.

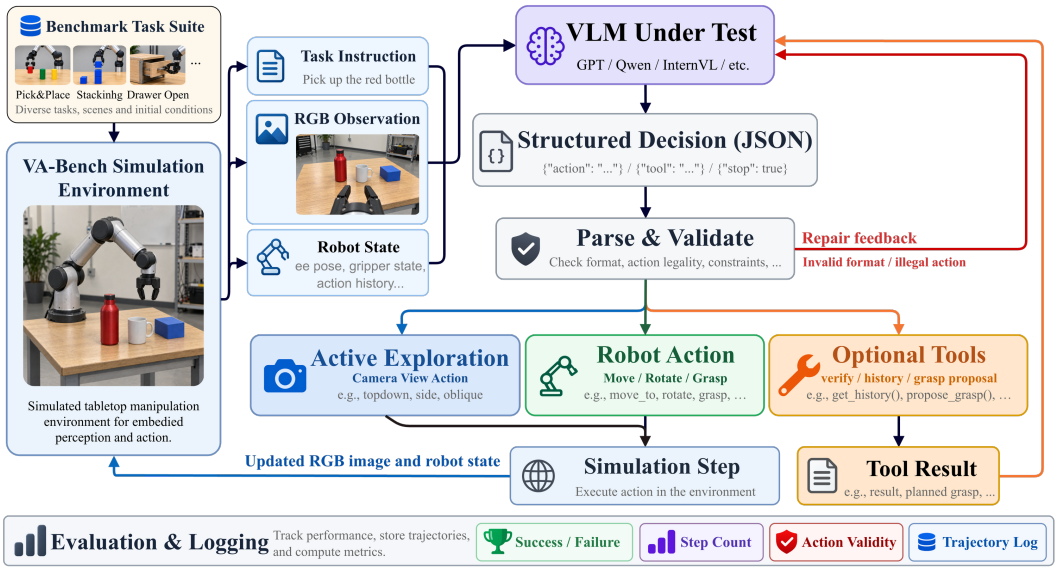


Figure 4: Schematic of the shared agent harness. Structured model decisions are validated and routed to camera control, robot execution, or non-environment tools, with observations and feedback returned to the model. Task, model, and command examples are illustrative. In the primary evaluation, robot actions are metric translations, rotations, and gripper open/close commands; the illustrated grasp-proposal option is not used. Camera actions change only the observation pose, and non-environment tools do not advance the simulator. Demonstration-derived summaries also condition the model but are omitted from the diagram.

D.1 Persistent decision loop

At each outer control decision, the harness sends the current observation payload together with the retained conversation state and asks the MLLM for one JSON object. The object denotes a non-environment tool call, a robot action, a camera action when enabled by the observation protocol, or a stop decision. Tool results are returned to the same thread and may trigger another model call; they do not advance the simulator. Robot and camera actions pass harness-level schema and numerical-range validation before final environment-side arm, validity, and workspace checks. Malformed JSON and invalid decisions are returned to the model for a bounded number of repair attempts, and tool-call iterations are likewise bounded. The harness provides adapters for OpenAI-compatible Chat Completions and Responses endpoints and Anthropic Messages endpoints, normalizing their outputs into this common JSON protocol rather than relying on a provider-specific robot tool interface. The canonical active-view condition exposes camera actions. Both passive conditions remove camera actions from the model-visible JSON vocabulary because their five predefined views are supplied directly; the robot execution stack is unchanged. In the bundled condition, the five images are sent together at each observation point. In the sequential condition, the same five images are sent as separate image-only inputs in fixed order. Sequential image-only inputs do not advance the environment or consume the task budget; only a dispatched robot action does so.

The same conversation thread is retained across the entire episode. All conditions use the standard context-length limit and compaction policy; prior robot actions, planner outcomes, action-validity feedback, and textual messages remain in the retained history. Each active API request contains only the current RGB image. A bundled-passive request contains the current fixed-order five-image bundle, while a sequential-passive request contains one image from that same bundle per image-only input. Earlier visual payloads are not re-sent as the current observation. Compaction leaves the append-only session record intact while bounding the model-visible context.

For the primary base-task terminal-success comparison, the harness repeats the fixed 20-seed suite in three independent runs for each of the 12 evaluated models. Run identity is retained in the session record, and aggregation is performed only after computing the task-macro-average within each

run. The bundled and sequential passive controls, held-out transfer, composition, and trajectory annotation analyses retain their separately identified matched single-run protocols.

D.2 Model observations and actions

After reset and after every robot action, the harness acquires the next RGB payload together with robot proprioception and retained robot/action history. The active API sends one current image; the bundled-passive API sends the current control point's fixed-order five-image bundle; and the sequential-passive API sends those same five images one at a time in fixed order. An active camera action may return a newly selected view without advancing physical state; neither passive condition has such a command. Sequential image-only inputs likewise do not advance physical state or consume the task budget. Context limit, compaction policy, robot actions, and environment transitions are otherwise shared. Proprioception includes gripper state, end-effector pose, the midpoint between the physical fingers (GC), and the current gripper-local axes expressed in the world frame. The model also receives the preceding action's validity and a binary trajectory-planning outcome. It does not receive the target object's coordinates or pose, segmentation masks, oracle grasp points, contact state, or task waypoints. The fixed motion-planning layer necessarily accesses simulator state required for inverse kinematics and trajectory generation, but this privileged state is not exposed to the MLLM.

The robot interface contains world-frame Cartesian translation increments, gripper-local rotation increments, and open/close commands for one or two arms. Numerical translations and rotations are bounded; Cartesian targets are clamped to the configured robot workspace before execution. For a local rotation, the adapter transforms the requested gripper-local axis into the world frame, compensates the end-effector position so that GC remains approximately fixed, and decomposes large rotations into planner waypoints of at most 10°. A fixed inverse-kinematics and trajectory layer then attempts the requested motion. It may reject an unreachable trajectory, but does not select the task object, grasp, or semantic waypoint for the model.

When enabled, the camera interface combines five gripper-centered semantic viewpoints with bounded translation, zoom, yaw, and pitch actions. These actions directly update the simulated camera pose: they do not model a camera arm's reachability, collision constraints, or continuous path cost. Camera and robot actions both consume the episode step budget, whereas history, self-geometry, and demonstration-retrieval tools consume inference and tool budget without advancing the physical episode. We therefore characterize this interface as bounded task-conditioned viewpoint selection, not unconstrained physical camera trajectory planning or camera-arm control.

D.3 Demonstration and prompt conditioning

In the demonstration-conditioned protocol, the evaluated model first inspects sampled frames from an RGB-only video of a successful execution and writes a structured task summary. This track exposes no machine-readable expert action, end-effector pose, joint state, object coordinate, contact label, or semantic phase. The visible robot and object motion nevertheless conveys procedural supervision, including operation order and approach strategy; the protocol removes direct numerical action supervision rather than making the demonstration action-free. A learning gate requires the model to inspect multiple distinct frames before its summary can be stored. Model-specific-summary and shared-summary tracks allow the effect of self-distilled experience to be studied separately. The model-specific-summary condition is the primary end-to-end protocol, whereas the shared-summary condition is a controlled diagnostic rather than an interchangeable input for cross-model ranking.

The model is additionally conditioned on the task instruction, generic manipulation and safety constraints, and task-specific profiles where needed for roles or completion conditions. Thus, planning remains open-ended in the chosen observation–action sequence and numerical parameters, but it is not an unscaffolded discovery process. No low-level ground-truth trajectory, oracle waypoint sequence, or target-object state is included in this scaffolding.

D.4 EXECUTION RECORDS AND OUTCOME

The adapter stores the clean RGB frames delivered to the model separately from annotated debugging and recording views. Messages and events are written to append-only JSONL files, while image artifacts are stored separately and referenced by the session record. Per-step records include the selected action, the model-provided reason, robot self-geometry, action validity, and the planner outcome. An action is dispatched through the adapter to RoboTwin, which computes the next observation and invokes the task's own success check. Consequently, reported success is the environment-defined, physics-simulated task outcome rather than a visual judgment made by either the MLLM or the harness. As emphasized in Section 1, this is an end-to-end system measure whose outcome also depends on the fixed planner, action interface, and simulator dynamics.