MANIPSCOPE: A DIAGNOSTIC BENCHMARK FOR INSTRUCTION FOLLOWING AND GENERALIZATION IN ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

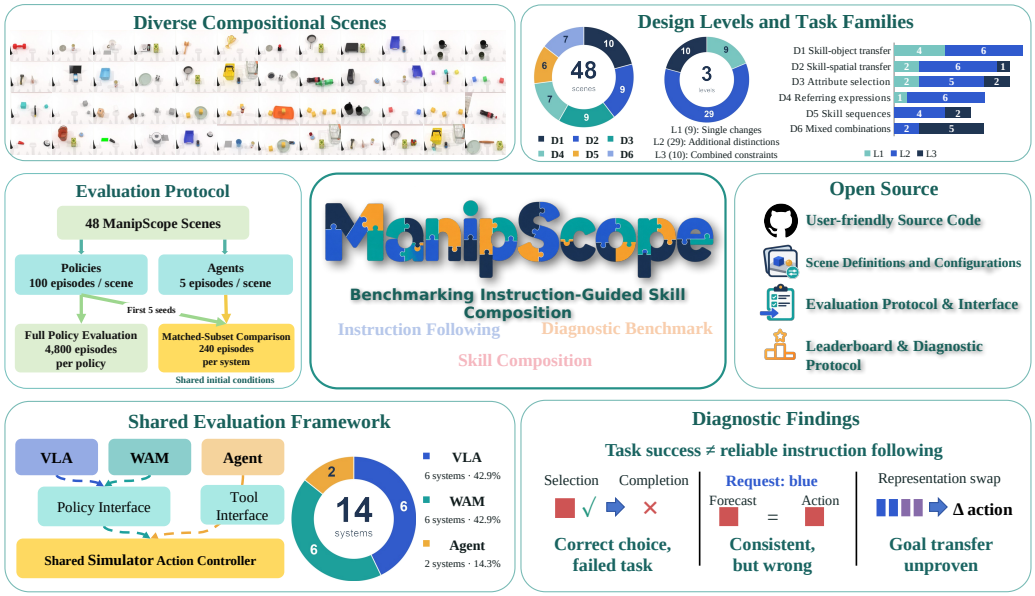

Figure 1: Overview of MANIPSCOPE. Forty-eight scenes span six task families and three design levels. Policies and tool-using agents are evaluated through their respective interfaces. Diagnostics distinguish selection from completion, prediction–action agreement from instruction correctness, and action sensitivity from goal-directed transfer. Open resources support evaluation and separate policy and agent leaderboards.

ABSTRACT

Completing a manipulation task does not establish that a robot follows the requested goal rather than a familiar routine. Our proposed MANIPSCOPE is an evaluation-only benchmark for robotic instruction following under changed task requirements. Built using RoboTwin 2.0's simulation resources, it specifies 48 scenes across six task families with explicit source-task links. We deliberately provide no expert training demonstrations for these scenes, targeting skill reuse under changed requests rather than fitting supplied examples. Task-success and selection metrics measure completion and satisfaction of the requested properties. Controlled instruction and layout changes test whether choices track the request, with future-video and internal-intervention probes where access permits. Evaluations of vision–language–action policies, world-action policies, and agents reveal uneven task performance, partial instruction satisfaction, and correct selection without completion. In controlled checkpoint case studies, individual-trial success can coexist with weak paired instruction correctness; forecasts and execution can agree on the wrong target; and internal interventions change predicted actions without establishing useful goal transfer. These findings show why success, consistency, and sensitivity must each be checked against the requested goal. Resources and separate policy and agent leaderboards: **Web** and **Code**.

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

1 INTRODUCTION

Language-conditioned robot policies aim to turn verbal goals into physical behavior. RT-2 and OpenVLA illustrate how vision–language pretraining supports robotic control (Zitkovich et al., 2023; Kim et al., 2025). When the requested object, relation, or combination changes, does the policy reuse its skills according to that request, or continue a familiar scene-dependent routine?

High success on familiar tasks cannot resolve this question alone. LIBERO-CF reports that evaluated VLAs can follow visual shortcuts and repeat learned behaviors despite alternative instructions in plausible layouts (Fang et al., 2026). InstructMove identifies a related ambiguity: visually salient or uniquely feasible targets can permit success without grounding the instruction (Zhao et al., 2026). When training on benchmark tasks is permitted, aggregate scores may not distinguish task-specific fitting from semantic generalization. Evaluation must therefore change the requirements that a familiar scene or routine would otherwise predict.

Such changes should remain interpretable. If a new task introduces an unavailable manipulation skill, failure may reflect that missing skill; if it reproduces demonstrated behavior, success need not establish generalization. This motivates a deliberate design choice: link evaluation tasks to existing skills, but provide no accompanying expert demonstrations for training on the evaluation tasks themselves. The links make intended reuse inspectable; actual novelty and prerequisite mastery still require checkpoint-specific evidence.

Task design also needs measurements that distinguish choice from completion. A robot may select the wrong object or choose correctly but fail to finish. For world-model-based policies that expose predicted video alongside actions (Bi et al., 2026), we can additionally check whether a forecast depicts the requested target and whether execution agrees with it. Agreement can nevertheless preserve the same wrong choice. A useful diagnosis must distinguish correct goal following from both successful execution in isolation and mere sensitivity of outputs or internal signals.

We propose MANIPSCOPE, an evaluation-only benchmark for robotic instruction following under changed requirements (Figure 1). Using RoboTwin 2.0's simulation resources (Chen et al., 2025), it specifies 48 scenes in six task families with explicit source-task links. Controlled tests examine responses to changed instructions and layouts, with future-video and internal-intervention evidence where access permits. Separate policy and agent leaderboards retain their respective protocols.[1]

- We introduce MANIPSCOPE, an evaluation-only benchmark with 48 scenes across six task families. We link each scene to source tasks and deliberately omit its training demonstrations to test skill reuse under changed requests.
- We pair attribute and target-selection metrics with three diagnostic layers: behavior, future video, and internal interventions. Selection and success metrics distinguish partial matches, correct choices, and completion; controlled diagnostics test whether those choices follow changed requests. Our case studies expose wrong-target prediction–action agreement and action sensitivity without established goal-directed transfer where diagnostic access permits.
- We release scene definitions, success rules, evaluation and diagnostic code, and separate policy and agent leaderboards, with reference results for twelve policy checkpoints and two agents.

2 RELATED WORK

Task generalization benchmarks. Our evaluation-only design relates first to benchmarks that define how test requirements differ from training. CompoSuite combines robots, objects, obstacles, and objectives to evaluate unseen task combinations (Mendez et al., 2022). VIMA uses multimodal prompts and separates several levels of generalization, including new combinations of familiar objects and textures (Jiang et al., 2023). LIBERO studies knowledge transfer through task suites with different demands (Liu et al., 2023). These works motivate explicit training–evaluation relationships. RoboTwin 2.0 provides the simulation and training foundation for this design: automated data generation, 50 bimanual tasks, and domain randomization (Chen et al., 2025).

[1]Project website: `https://anonymous.4open.science/w/ManipScope_webpage/index.html`. Code repository: `https://anonymous.4open.science/r/ManipScope`.

Instruction following and partial progress. Beyond task design, instruction-following evaluation must distinguish satisfying a requirement from merely completing a manipulation. ATOM-Bench evaluates atomic skills and held-out compositions in real-world manipulation, including partial progress (Wu et al., 2026). Qi et al. use instruction-factor preferences to guide data collection, with competing-factor tests that include jointly unsatisfiable instructions (Qi et al., 2026). Here the distinction between jointly unsatisfiable instructions and valid candidate targets matters: our attribute measurements concern the latter. Controlled tests make this distinction more direct. LIBERO-CF varies feasible instructions in shared layouts and separates target-contact grounding from task success (Fang et al., 2026).

Cross-system evaluation and model-access diagnostics. Zhang et al. (2026) compare VLA and world-action models under visual and language perturbations in LIBERO-Plus and RoboTwin 2.0-Plus. World-action models that jointly generate video and actions (Bi et al., 2026) permit checks of target agreement across outputs; where internal access permits, representation replacement provides a separate intervention. The latter tests a specified computational connection and requires controls on both the replacement and the measured effect (Zhang & Nanda, 2024). Model diversity broadens coverage; the substantive comparison remains the behavior of the evaluated systems under the specified tasks and access-dependent probes.

These studies motivate going beyond aggregate success: completion alone does not distinguish correct target choice from partial satisfaction, and output agreement or sensitivity does not establish correct goal following. Our proposed MANIPSCOPE connects an evaluation-only task suite to selection measurements and controlled tests under paired instructions and layouts, with future-video and internal-intervention evidence where access permits. The contribution connects source-linked task evaluation with separate measures of choice, completion, forecast correctness, and intervention direction; the model-access probes are optional case-study protocols.

3 BENCHMARK DESIGN

MANIPSCOPE tests whether existing manipulation skills can satisfy changed instruction requirements. We connect each evaluation scene to relevant RoboTwin tasks, specify what changes, and fix the resulting tests for comparison.

3.1 DESIGN RATIONALE AND SCENE CONSTRUCTION

RoboTwin 2.0 provides simulation resources, object assets, and existing robot training tasks (Chen et al., 2025). Building on these resources connects new requirements to available manipulation skills: a scene can change the object being handled, its required spatial relation, the description used to select it, or the order of familiar actions. MANIPSCOPE is evaluation-only by design: we deliberately provide no expert demonstrations for training on these evaluation scenes. The aim is to distinguish applying the requested rule from reproducing a familiar solution.

To construct a scene, we select relevant task components, specify the changed requirements, and instantiate object roles, asset variants, and initial-position regions. Instruction templates state the requested target and goal; physical success conditions specify the required object state, spatial relation, or action order. For example, placing a tea box in a basket and then pressing a bell combines two actions in a required order. Appendix A.2 gives construction examples; Figure 4 shows the complete scene overview.

The source-task mapping records reused components and changed requirements; Appendix A.2 provides the detailed links.

Remark (Scope of generalization claims). Source-task links document intended skill reuse and changed requirements; they do not by themselves establish what a checkpoint has learned or previously encountered. Claims of compositional generalization additionally require evidence of component familiarity and the absence of the tested combination from a specified training set.

3.2 TASK FAMILIES AND LEVELS

We group the scenes by the requirements changed or combined (D1–D6; Figure 1). D1, skill–object transfer, applies a manipulation skill to a changed object. D2, skill–spatial transfer, requires a

specified spatial relation in the evaluation configuration. D3, attribute selection, identifies targets by properties such as color, size, or category. D4, referring expressions, identifies targets through relations, order, or negation. D5, skill sequences, requires actions to satisfy successive goals in the requested order. D6, mixed combinations, combines multiple such requirements. These families organize overlapping task demands; selecting a spatially described object, for example, requires both resolving the reference and manipulating the object.

Levels summarize the designed combination of requirements. L1 comprises scenes with a single replacement, few distractors, and one action segment. L2 comprises scenes with an additional distinction or a two-action task. L3 comprises scenes with combined constraints or dependent actions. We note that these are manual design annotations, not a calibrated difficulty scale: changes in object geometry and execution demands can also affect performance. Appendix A.1 gives the detailed assignment rules and family–level counts.

The resulting suite specifies which requirements a robot must satisfy and how they relate to existing tasks. The next section defines what task completion and observed choices reveal, then introduces controlled changes to test whether target choices follow the instructions.

4 Evaluation Metrics and Diagnostic Protocol

The scene design specifies what is requested. We now define how recorded behavior becomes evidence. A single correct choice does not establish that language determined it.

4.1 Task Success and Selection Metrics

Task success rate (SR). Let y_i be the binary success label for episode i, determined by the requested target, physical goal, reference constraints, and any required action order. Success is recorded when these conditions are satisfied during execution. SR is the fraction of evaluation episodes with $y_i = 1$, counting failures and timeouts in the denominator. Appendices A.4 and A.5 summarize the physical success criteria.

For Overall, we average task success rates within each family, then weight the six families equally (Appendix C). This gives each family the same contribution despite unequal task counts; Pooled instead weights episodes equally. Pooled success is the fraction of all evaluated episodes that succeed. Family–level rates average tasks in the corresponding subset; global level rates average all tasks at that level.

Selection metrics. SR alone does not reveal which object a policy initially selects: an unsuccessful episode may involve a wrong target or failure after a correct choice. We therefore measure the target choice expressed through interaction and, for attribute-selection tasks, which requested properties that choice satisfies. These measurements provide behavioral evidence of target choice rather than a direct readout of latent intention.

Target selection. For tasks with a designated target and an observable selection event, let $c_i = 1$ when the object selected at that predefined event is the designated target, and zero otherwise. Target-selection correctness is the fraction of episodes with $c_i = 1$ among those with complete event records. Ambiguous choices and trajectories without a qualifying event receive zero, but are distinguished from observed wrong-object choices. Missing records are unmeasurable; Appendix D.1 specifies coverage, conditional correctness, and their denominators.

In the formal-task attribute study, selection is measured at the first loaded gripper–object or hammer–object interaction. Event thresholds, sampling, and sensitivity checks appear in Appendices D.1 and D.3.

Attribute correctness. For attribute-selection tasks, we score the requested properties of the selected object. A task is eligible only when its instruction requests the tested attribute and its candidates permit that choice to be scored. The per-scene rate averages correctness over complete event records (Appendix D.1). Eligibility is fixed before behavior is observed. Ambiguous or absent selections receive zero; unrequested attributes are not applicable. We report coverage and eligible denominators, weighting eligible tasks equally when aggregating each attribute (Appendix D.2).

This refinement retains partial matches that exact-target correctness would miss. Selecting a large red object when a small red object is requested earns color credit, but neither size nor exact-target

credit. Color-versus-size comparisons depend on the eligible task designs and are not inferred from aggregate success rates.

Selection–completion outcomes are analyzed jointly, with native success scored independently of the selection event (Appendix D.1).

4.2 Controlled Language-Following Diagnostics

Controlled contrasts test whether target choices follow changed instructions by varying instructions and layouts independently. Future-video and internal-intervention probes require access to the corresponding outputs or representations. These diagnostic layers concern access to evidence, not a shared serial architecture; unavailable layers are not scored as failures (Figure 8; Appendix E).

Paired instruction and layout responses. In the paired behavioral study, each initial scene configuration has two valid target instructions and two layouts, with the red and blue objects physically exchanged. The paired and formal-task studies use different event definitions and denominators (Appendices E.1 and D.1).

Instruction-paired correctness requires both target instructions to be followed in the same view; always selecting one object cannot pass. Layout-paired correctness requires the requested color to be selected across both layouts. These are paired correctness rates, not merely measures of action change. A correct trial has a unique first recorded loaded gripper contact with the requested object; no qualifying contact or an ambiguous contact receives zero credit. Uncertainty estimation keeps all four conditions of an initial configuration together (Appendix E.1).

Future-video outputs. For systems that expose predicted images, we compare the visible approach target with the instruction and with the later contact target, retaining unresolved predictions. The readout compares target identity across horizons, not frame-aligned contact accuracy. It separates instruction correctness from forecast–action agreement: a forecast and execution can agree on the wrong object. Agreement alone does not establish that the forecast causally directs action. Appendix E.2 specifies the labeling procedure and its limits.

Internal interventions. Where representations can be accessed, we replace selected representations with those from an alternative-instruction run while holding recipient inputs and sampling noise fixed. We measure action change and donor-goal preference change: the change from baseline in the difference between commanded distances to the recipient and donor targets. Positive values favor the donor relatively, without necessarily reducing donor-target distance. Closed-loop benefit requires executing the intervened policy and is not tested here. Appendix E.3 specifies the readouts and controls. These probes concern the tested computational connection, not a global ranking of language and vision.

Together, these diagnostics distinguish correct target choice from agreement between prediction and action, and from action sensitivity alone. Behavior provides a common starting point across systems; available forecasts and interventions help identify whether to examine target choice, future prediction, or goal-directed action.

5 Experiments and Analysis

We organize the evaluation around three questions that connect benchmark performance to evidence of instruction following:

- **RQ1.** How well do current systems complete tasks under changed requirements?
- **RQ2.** What does task success conceal about target selection and satisfaction of the requested properties?
- **RQ3.** Do choices correctly track changed requests, and what additional evidence do forecasts and internal interventions provide?

5.1 Experimental Setup and Reporting

We evaluate twelve RoboTwin-trained policy checkpoints, identified with their sources in Table 1. Each runs 100 episodes per scene without adaptation on our proposed ManipScope. Training

backgrounds and control interfaces differ, so comparisons concern the evaluated systems rather than isolated architecture effects. Appendices B.1 and B.2 give checkpoint and inference settings. The evaluated paths bypass WLA's World Expert and subtask generation, skip future-video generation in FastWAM, and disable chain-of-thought (CoT) and autoregressive action generation in G0.5.

GPT-6 and Gemini 3.8 Flash are evaluated as tool-using agents with five episodes per scene. Both agents use high reasoning settings. Their configurations, tools, and budgets are given in Appendix B.3.

We fix scene definitions, instruction-template pools, and initial-state sets independently of the tested systems' outcomes, then reuse these evaluation conditions across systems (Appendix A.3). Policies receive three RGB views with model-dependent state and history inputs; agents receive images, measured state, and tool feedback. Initial instruction wording differs across groups in 84 of 240 matched episodes. On two sequence tasks, agents receive the complete instruction while policies use stage-dependent instructions (Appendix B.5).

Table 1 reports the full 4,800-episode policy evaluation; Table 2 compares success and timing on the shared first-five subset of 240 episodes per system. The complete matched comparison remains in Table 11.

A separate instrumented study measures selection and completion on nine attribute tasks. Five policy checkpoints and both agents each run five fixed trials per task, yielding 315 episodes (Appendix D.1). Instructions, initial object states, candidate properties, and the event definition are matched across systems; agent budgets are specified in Appendix D.4. These episodes are analyzed separately from the full-suite and matched-subset results.

5.2 Task Performance and Selection Outcomes

For RQ1, the full-suite results show uneven competence across task requirements. G0.5 attains the highest observed Overall score, 40.40% (Table 1), but does not lead every family. Referring expressions (D4) and mixed combinations (D6) remain difficult across the evaluated policies, even when performance is substantially stronger in other families (Appendix C). The family breakdown therefore matters beyond ranking: it identifies requirements under which an otherwise competitive system remains unreliable. These scores describe performance on the changed requirements; without matched source-task evaluations and checkpoint-specific training coverage, they do not measure a loss from mastered components to unseen combinations.

The shared-subset comparison adds a second consideration: higher success can come at greater runtime, but more inference time does not by itself imply higher success. Under different observation, control, and instruction interfaces, GPT-6 attains 76.35% Overall compared with 42.30% for G0.5, with greater model-side and episode wall time in the observed deployments. Gemini takes longer than GPT-6 while completing fewer tasks than the three displayed policies (Table 2). Restricting the comparison to the 156 episodes with identical initial instruction strings gives 121/156 successes (77.56%) for GPT-6 and 57/156 (36.54%) for G0.5 (Appendix B.5). This descriptive check retains the other interface and deployment differences and does not isolate an architecture effect. Taken together, the results suggest judging progress by reliable coverage of task requirements alongside aggregate success and runtime.

RQ2 asks what these completion scores leave unresolved. In the separate attribute study, 50/315 episodes (15.9%) select the correct target first but do not complete the task, with such outcomes observed among both policies and agents (Figure 2; Table 16). Failure thus need not begin with a wrong-target choice. Conversely, even an incorrect target can satisfy part of the request: on the conjunctive color–category task, 14/35 episodes have a first interaction that matches only one requested property (Table 19). Completion alone does not distinguish these partial matches from other initial choices, or identify failures after a correct choice, although these distinctions pose different questions for diagnosis.

This distinction is behavioral, not a direct decomposition into language and motor errors. The first event cannot identify the cause of subsequent failure, and successful completion can follow an initial mistake or occur without a loaded selection event. The bell task illustrates the latter because its native success predicate is proximity-based. We therefore retain event coverage alongside correctness and score native success independently (Table 15; Appendices D.2 and D.3). Raising the event threshold from 0.1 to 1 N changes gripper-task correct-target rates by at most 3/35 (8.6 percentage points) per system; tool-hit counts are unchanged. These measurements answer RQ2 by retaining which requested properties the initial choice satisfies, even when the task fails.

Table 1: Task success (%) for the evaluated checkpoints and inference paths by family and level, with 100 episodes per task. Overall equally averages six family means; Pooled uses all 4,800 episodes (Equation 1). Bold marks maxima across policies within each family–level subset or aggregate, including ties, without implying significance. Dashes denote empty subsets; global summaries appear in Appendix C.

Policy	Level	Overall↑	D1↑	D2↑	D3↑	D4↑	D5↑	D6↑	Pooled↑
DM0.5* (Dexmal, 2026)	L1		57.75%	**70.00%**	73.50%	0.00%	–	–	
	L2	34.14%	43.17%	26.83%	71.60%	15.33%	49.50%	41.50%	36.21%
	L3		–	16.00%	17.00%	–	0.50%	3.60%	
$\pi_{0.5}$* (Black et al., 2025a)	L1		38.00%	43.50%	50.00%	0.00%	–	–	
	L2	31.36%	52.00%	14.17%	56.00%	21.50%	39.25%	39.00%	32.65%
	L3		–	42.00%	24.00%	–	17.50%	12.40%	
StarVLA† (StarVLA Community, 2026)	L1		54.00%	59.00%	25.50%	**9.00%**	–	–	
	L2	27.90%	44.17%	11.00%	50.60%	13.67%	54.00%	12.00%	29.38%
	L3		–	36.00%	30.00%	–	0.00%	2.80%	
FastWAM† (Yuan et al., 2026)	L1		48.25%	42.00%	70.00%	0.00%	–	–	
	L2	24.18%	43.67%	11.67%	51.40%	11.83%	24.50%	12.00%	26.56%
	L3		–	14.00%	30.50%	–	0.00%	0.20%	
π_0‡ (Black et al., 2025b)	L1		24.50%	5.00%	39.00%	0.00%	–	–	
	L2	12.75%	24.50%	4.83%	17.80%	10.50%	27.25%	0.50%	13.33%
	L3		–	0.00%	8.00%	–	0.00%	0.00%	
AHA-WAM† (Cai et al., 2026)	L1		60.25%	56.50%	72.00%	0.00%	–	–	
	L2	28.63%	43.33%	12.33%	69.60%	14.83%	28.25%	17.50%	31.27%
	L3		–	17.00%	30.00%	–	3.00%	0.20%	
X-VLA‡ (Zheng et al., 2025)	L1		42.75%	19.50%	66.50%	0.00%	–	–	
	L2	22.23%	25.33%	9.33%	25.80%	10.33%	**66.75%**	0.00%	22.42%
	L3		–	11.00%	21.50%	–	4.00%	1.00%	
WLA† (Yang et al., 2026)	L1		42.75%	46.50%	47.50%	0.00%	–	–	
	L2	25.35%	40.00%	9.33%	48.20%	16.17%	45.75%	15.00%	26.71%
	L3		–	39.00%	18.50%	–	0.00%	0.00%	
GalaxeaVLA (G0.5)* (Liu et al., 2026)	L1		67.00%	63.50%	75.50%	0.00%	–	–	
	L2	**40.40%**	51.50%	**41.83%**	**78.60%**	17.50%	48.00%	19.50%	**42.88%**
	L3		–	19.00%	31.50%	–	8.50%	**24.80%**	
X-WAM* (Guo et al., 2026)	L1		**86.00%**	55.50%	74.00%	0.00%	–	–	
	L2	37.29%	**73.17%**	11.17%	64.20%	23.67%	42.25%	4.50%	40.29%
	L3		–	34.00%	36.00%	–	18.50%	8.20%	
Motus* (Bi et al., 2026)	L1		50.75%	62.00%	**84.50%**	0.00%	–	–	
	L2	35.03%	51.33%	35.83%	53.00%	15.67%	59.00%	10.00%	37.00%
	L3		–	17.00%	39.00%	–	6.00%	7.00%	
LingBot-VA* (Li et al., 2026)	L1		60.50%	56.50%	72.00%	6.00%	–	–	
	L2	38.29%	52.67%	16.00%	59.40%	**28.50%**	45.75%	**42.00%**	39.92%
	L3		–	**53.00%**	**53.50%**	–	**27.50%**	9.80%	

Level subsets contain different tasks and do not isolate a single difficulty factor. Training provenance: *robot/latent-action pretraining followed by RoboTwin adaptation; †pretrained backbones followed by RoboTwin action training, with no additional robot-action pretraining reported; ‡checkpoint lineage unverified. Details: Appendix B.1; task counts: Appendix A.1.

5.3 Controlled Language-Following Diagnostics

A remaining ambiguity motivates RQ3: a correct choice under one instruction could still reflect a preference for a familiar object or location. We therefore test whether target choices change to match the revised request.

We instantiate the diagnostic protocol with $\pi_{0.5}$ as a VLA example and Motus as a video-generating world-action example. These checkpoint-specific case studies illustrate the protocol rather than establish class-wide behavior. The controlled study crosses 20 initial configurations of one three-color

Table 2: System-level success and timing under different observation, control, and instruction interfaces: three selected policies and both agents, 240 matched episodes each, including failures. Bold marks success maxima and time minima within displayed groups. Full results: Table 11. *Instruction differences: Appendix B.5. Timing measures the observed deployments; definitions: Appendix B.6.

System	Overall↑	Pooled↑	Calls (/ep.)	Model↓ (s/call)	Model↓ (s/ep.)	Eval wall↓ (s/ep.)	Episode wall (s/ep.)↓
Selected policies							
GalaxeaVLA (G0.5)	**42.30%**	**44.58%**	80.9	1.313	106.2	448.8	487.2
Motus	35.14%	36.67%	45.2	6.578	297.1	661.3	683.7
$\pi_{0.5}$	31.92%	33.33%	26.3	**0.226**	**6.0**	**346.2**	**359.6**
Agents							
GPT-6*	**76.35%**	**77.08%**	24.9	**13.248**	**329.4**	**710.3**	**730.4**
Gemini 3.8 Flash*	24.35%	24.17%	35.2	46.260	1630.3	1983.9	2002.2

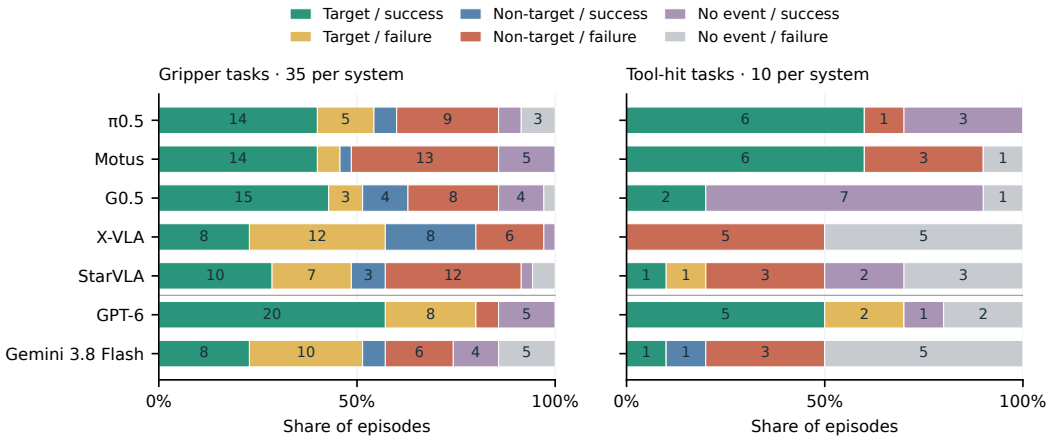

Figure 2: First loaded interaction (≥ 0.1 N) and native task success for five policies and two agents: 35 gripper and 10 tool-hit trials per system. Stacks show episode shares; labels show counts. Target/non-target refers to the first qualifying interaction; no event means none met the threshold. Success is scored independently.

block task with two instructions and two layouts for each checkpoint, yielding 160 trajectories. The 95% intervals use 20,000 bootstrap resamples of the 20 configurations (Appendix E.1).

The paired behavioral results separate isolated success from reliable responses to alternative requests. Motus succeeds in 40.0% of individual trials, yet selects the requested target under both instructions in only 1/40 same-layout pairs; $\pi_{0.5}$ does so in 29/40 pairs (Figure 3). Motus contacts the same color under both instructions in 38/40 pairs, compared with 9/40 for $\pi_{0.5}$; Table 21 separates target colors and layouts. Layout-paired results are reported in Table 20. The practical implication is to test alternative valid requests in the same scene: repeating successful behavior is insufficient if the requested target can change.

The future-video readout asks whether the same mismatch is already visible in the predicted outcome. In the Motus case, all 77 readable first-decision forecasts (out of 80) agree with the later contact target beyond the forecast horizon, including 46 agreements on the wrong object (Figure 3; Appendix E.2). Prediction and execution can therefore be mutually consistent while jointly violating the request. Labels use a task-specific image rule with visual review, without independent blinded annotation. This agreement does not establish that the forecast causes the later choice.

Internal intervention tests whether future-representation replacement changes predicted actions and donor-goal preference. Replacement produces a mean root-mean-square (RMS) joint-command difference of 0.030 rad and a mean donor-goal preference change of 0.560 mm (95% interval: $[-0.019, 1.299]$ mm). Changing the full instruction produces a 0.096 rad RMS difference and a 2.529 mm preference change. The actions are sensitive to replacement, yet useful goal transfer is not established. Only two readable contrasts predict different target objects, both from one initial

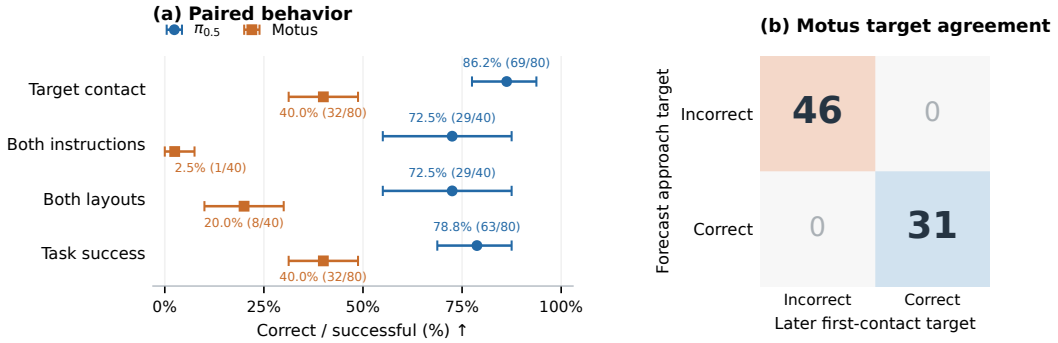

Figure 3: Checkpoint-specific color-selection diagnostics under a separate 256-native-action protocol. (a) Contact and success rates use 80 trajectories per checkpoint; paired correctness uses 40 pairs. Error bars show 95% bootstrap intervals over 20 initial configurations. (b) Motus forecast and later first-contact targets are jointly wrong in 46 cases and jointly correct in 31; three forecasts are unresolved.

configuration, and no intervened closed-loop improvement is tested (Appendix E.3). The result supports evaluating the direction of an intervention's effect, rather than treating any action change as evidence of improved goal following; it does not establish that future representations have no goal influence.

Together, these case studies identify a recurring diagnostic pitfall: successful episodes, consistent forecasts, and responsive actions each leave correct language following unresolved. Evidence must be tied to the requested goal at each observed stage. This is the contribution of the diagnostic analysis: narrowing what to investigate next, rather than inferring a complete failure mechanism from one score or signal (Appendix E.4).

6 Limitations

Our internal interventions are evaluated in open loop: we measure changes in predicted actions and relative target-distance preference under commanded kinematics without executing the intervened policy. These readouts do not establish improved target selection or task completion. Closed-loop evaluation is needed to determine whether the observed effects persist as actions change subsequent observations and whether they improve behavior.

The benchmark and diagnostic studies are conducted entirely in simulation. Their findings have not been validated on physical robots, where perception noise, contact dynamics, and hardware constraints may affect both task performance and diagnostic measurements. Extending the tasks and protocols to real robots is therefore needed to assess their applicability beyond simulation.

7 Conclusion

ManipScope tests robotic instruction following under changed requirements through tasks with explicit source-component links, deliberately providing no expert training demonstrations for its evaluation scenes. Joint selection and success measurements distinguish partial instruction satisfaction, correct choice, and completion. Controlled diagnostics then test whether choices track changed requests, with future-video and internal-intervention evidence where access permits. Our studies show why successful execution, prediction–action agreement, and action sensitivity each leave aspects of goal following unresolved. Together, the benchmark and diagnostics support more informative system comparisons and targeted investigation of language-following failures.

REPRODUCIBILITY STATEMENT

The project website (https://anonymous.4open.science/w/ManipScope_webpage/index.html) provides task documentation and separate policy and agent leaderboards. The anonymous code repository is https://anonymous.4open.science/r/ManipScope. Appendices A, B, and F specify task construction and success criteria, evaluation configurations, and per-task results. Appendices D and E describe the selection metrics and controlled diagnostic procedures.

ETHICS STATEMENT

All experiments are conducted in simulation and do not establish real-world safety or reliability. Use and redistribution of inherited simulation assets and model resources remain subject to their original licenses and terms.

AI USE STATEMENT

OpenAI Codex assisted with code review and manuscript polishing. GPT-6 and Gemini 3.8 Flash were evaluated as tool-using agents for robotic manipulation in the experiments. The authors are responsible for the experimental design, verification of results, and all claims and content in the paper.

REFERENCES

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A Unified Latent Action World Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Bi_Motus_A_Unified_Latent_Action_World_Model_CVPR_2026_paper.html.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.010.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. AHA-WAM: Asynchronous Horizon-Adaptive World-Action Modeling with Observation-Guided Context Routing. *arXiv preprint arXiv:2606.09811*, 2026. doi: 10.48550/arXiv.2606.09811. URL https://arxiv.org/abs/2606.09811.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable

data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. doi: 10.48550/arXiv.2506.18088. URL https://arxiv.org/abs/2506.18088.

Dexmal. OpenDM. GitHub software repository, 2026. URL https://github.com/dexmal/opendm. DM0.5 release; accessed September 26, 2026.

Yu Fang, Yuchun Feng, Dong Jing, Jiaqi Liu, Yue Yang, Zhenyu Wei, Daniel Szafir, and Mingyu Ding. When vision overrides language: Evaluating and mitigating counterfactual failures in VLAs. *arXiv preprint arXiv:2602.17659*, 2026. doi: 10.48550/arXiv.2602.17659. URL https://arxiv.org/abs/2602.17659.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4D World Action Modeling from Video Priors with Asynchronous Denoising. *arXiv preprint arXiv:2604.26694*, 2026. doi: 10.48550/arXiv.2604.26694. URL https://arxiv.org/abs/2604.26694.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, 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, 2023. URL https://proceedings.mlr.press/v202/jiang23b.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 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, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control. *arXiv preprint arXiv:2601.21998*, 2026. doi: 10.48550/arXiv.2601.21998. URL https://arxiv.org/abs/2601.21998.

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, 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, Xiao Liu, Dong Ke, Changxun Pan, Chenru Wu, Tailai Cheng, Xiaoshu Ren, Xinlei Zhang, Jianning Cui, Zijie Zhao, Haoyu Zhang, Kaiming Xu, Haodong Yang, Bowen Zhang, Jiahui Niu, Shaoting Zhu, Shiduo Zhang, and Hang Zhao. G0.5: One Autoregressive Stream for Robot Reasoning and Action. *arXiv preprint arXiv:2608.11739*, 2026. doi: 10.48550/arXiv.2608.11739. URL https://arxiv.org/abs/2608.11739.

Jorge A. Mendez, Marcel Hussing, Meghna Gummadi, and Eric Eaton. CompoSuite: A compositional reinforcement learning benchmark. In *Proceedings of the 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pp. 982–1003, 2022. URL https://proceedings.mlr.press/v199/mendez22a.html.

Yu Qi, Zhang Ye, Xinyi Xu, Yuxuan Lu, Amitoj Sandhu, Boce Hu, Haojie Huang, Jonathan Tremblay, and Lawson L. S. Wong. Scale up strategically: Learning compositional generalization via bias-aware evaluation and data collection for robotic manipulation. *arXiv preprint arXiv:2607.21582*, 2026. doi: 10.48550/arXiv.2607.21582. URL https://arxiv.org/abs/2607.21582.

StarVLA Community. StarVLA: A Lego-like Codebase for Vision-Language-Action Model Developing. *arXiv preprint arXiv:2604.05014*, 2026. doi: 10.48550/arXiv.2604.05014. URL https://arxiv.org/abs/2604.05014.

Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. cuRobo: Parallelized collision-free minimum-jerk robot motion generation, 2023. URL https://arxiv.org/abs/2310.17274. arXiv:2310.17274.

Zenan Wu, Bingqing Wei, Lu Liu, Zheqi He, Xi Wang, Jiakang Liu, Zehui Li, Guocai Yao, Jing-Shu Zheng, Xi Yang, and Yongtao Wang. ATOM-Bench: A real-world benchmark for atomic skills and compositional generalization in manipulation policies. *arXiv preprint arXiv:2606.16826*, 2026. doi: 10.48550/arXiv.2606.16826. URL https://arxiv.org/abs/2606.16826.

Yi Yang, Zhihong Liu, Siqi Kou, Yiyang Chen, Yanzhe Hu, Jianbo Zhou, Boyuan Zhao, Zhijie Wei, Xiao Xia, Xueqi Li, Pengfei Liu, and Zhijie Deng. World-Language-Action Model for Unified World Modeling, Language Reasoning, and Action Synthesis. *arXiv preprint arXiv:2606.05979*, 2026. doi: 10.48550/arXiv.2606.05979. URL https://arxiv.org/abs/2606.05979.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do World Action Models Need Test-time Future Imagination? *arXiv preprint arXiv:2603.16666*, 2026. doi: 10.48550/arXiv.2603.16666. URL https://arxiv.org/abs/2603.16666.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/06a52a54c8ee03cd86771136bc91eb1f-Abstract-Conference.html.

Zhanguang Zhang, Zhiyuan Li, Behnam Rahmati, Rui Heng Yang, Yintao Ma, Amir Rasouli, Sajjad Pakdamansavoji, Yangzheng Wu, Lingfeng Zhang, Tongtong Cao, Feng Wen, Xinyu Wang, Xingyue Quan, and Yingxue Zhang. Do World Action Models Generalize Better than VLAs? A Robustness Study. *arXiv preprint arXiv:2603.22078*, 2026. doi: 10.48550/arXiv.2603.22078. URL https://arxiv.org/abs/2603.22078.

Mengao Zhao, Ziang Li, Chaodong Huang, Mengchen Ma, Haoyi Jiang, Yiwei Jin, Xinjie Wang, Yun Du, Xuewu Lin, Taojun Ding, Hongyu Xie, Jackson Jiang, Chunlei Yu, Kaihua Zhang, Lichao Huang, Liu Liu, Tianwei Lin, and Zhizhong Su. InstructMove: A text-indispensable benchmark for instruction-following manipulation. *arXiv preprint arXiv:2608.22990*, 2026. doi: 10.48550/arXiv.2608.22990. URL https://arxiv.org/abs/2608.22990.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-Prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model. *arXiv preprint arXiv:2510.10274*, 2025. doi: 10.48550/arXiv.2510.10274. URL https://arxiv.org/abs/2510.10274.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. 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, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

Appendix guide. Appendix A gives task definitions and success criteria; Appendix B specifies evaluated systems and timing. Appendix C collects aggregate task scores, followed by formal-task attribute analyses in Appendix D and controlled diagnostics in Appendix E. The complete per-task result tables are collected last in Appendix F.

A Task Inventory, Construction, and Success Criteria

This appendix collects the task inventory, construction examples, source-task links, initial-state procedure, and physical success criteria used by the benchmark.

A.1 Scene Inventory and Design Levels

Table 3 lists the 48 evaluation scenes by task family and level. D1–D6 denote skill–object transfer, skill–spatial transfer, attribute selection, referring expressions, skill sequences, and mixed combinations.

Table 3: Scene counts by task family and level.

ID	Family	L1	L2	L3	Total
D1	Skill–object transfer	4	6	0	10
D2	Skill–spatial transfer	2	6	1	9
D3	Attribute selection	2	5	2	9
D4	Referring expressions	1	6	0	7
D5	Skill sequences	0	4	2	6
D6	Mixed combinations	0	2	5	7
	Total	9	29	10	48

L1 denotes single changes with few distractors and one action segment; L2 introduces additional distinctions or two-action tasks; L3 combines multiple constraints or dependent actions. The manual assignments comprise 9/29/10 scenes at L1/L2/L3. These annotations summarize designed requirements, not calibrated difficulty: object geometry and execution demands can also affect performance.

The scene atlas (Figure 4) shows the task configurations. Figure 5 shows the scene identifiers and their source-task links. Execution limits are specified in Appendix B.

A.2 Scene Construction and Source-Task Links

Each scene specifies object roles, assets or geometry, sampling regions, spatial relations, and a goal predicate. Task-dependent slot permutations and relative placements preserve the relation used to identify the requested target.

For scene c, the source-task set $P_c \subseteq \mathcal{B}$ records its links to RoboTwin base tasks $\mathcal{B}$, together with reused components and changed requirements. The 48 scenes link to 34 of the 50 RoboTwin task identifiers through reused objects, skills, and relations (Figure 5). This is design metadata, not an execution plan supplied to the system. The links alone do not establish checkpoint-specific training exposure or mastery of the reused components.

Six concrete examples illustrate the construction:

1. D1, `place_teabox_scale`: reuse the scale-placement task and substitute a tea box as the manipulated object; success uses the shared supported-on predicate.
2. D2, `card_right_of_scanner`: combine card/scanner handling with the right-of relation. Its goal requires a rightward separation between 5 and 22 cm, with lateral and height tolerances and released grippers.
3. D3, `attr_color_block_3way`: place three equal-size red, green and blue blocks in permuted slots and request the red block in the formal scene. Lifting requires 6 cm clearance

together with gripper contact and selection guards. The two-instruction diagnostic study explicitly changes the requested target; it is a separate protocol.

4. D4, `ref_block_on_plate`: use identical-color blocks and identify the target by its initial placement on a plate. The target's initial pose is specified relative to the plate; the goal is lifting it with 6 cm clearance.
5. D5, `serve_teabox_then_ring`: require placing the tea box in the basket, then pressing the bell. The native sequence state records progress and the final predicate retains the placement requirement.
6. D6, `attr_teabox_basket_then_ring`: add a distractor object to the placement-and-bell sequence. The scene combines target discrimination, placement and ordered action requirements.

Figure 4 shows the 48 evaluation scenes, grouped by task family; Figure 5 then shows their declared source-task links.



Figure 4: All 48 evaluation scenes, grouped into D1–D6, with task identifiers and assigned levels.

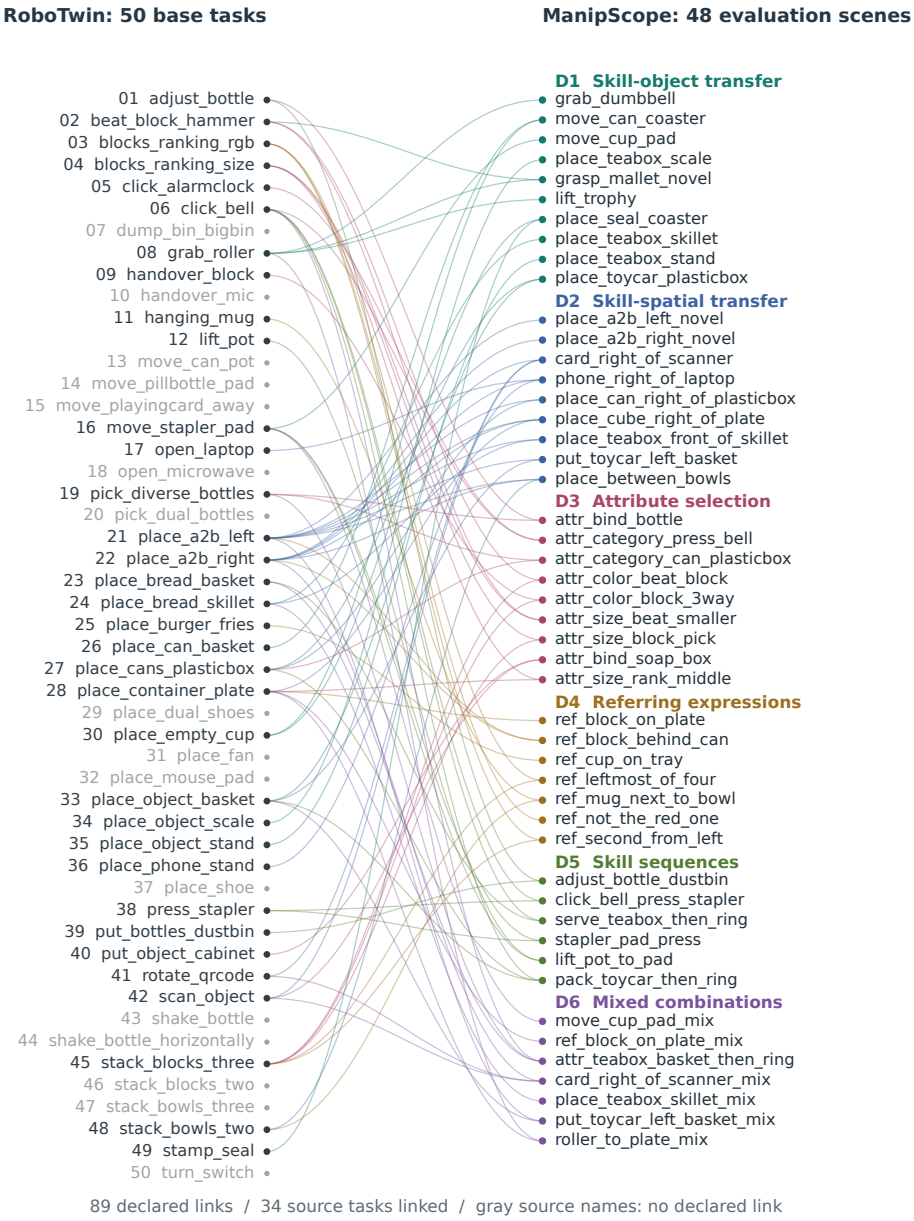

Figure 5: Declared task-component links: 48 MANIPSCOPE scenes connect to 34 RoboTwin source tasks through 89 links. Edge colors identify task families; gray source names have no declared link. Links describe declared component reuse, not verified training exposure.

Table 4: Success conditions for ordered tasks. Distances are in meters; shared sequence scenes also apply target-selection and reference-stability guards.

Task	First condition	Later/final condition
`adjust_bottle_dustbin`	Bottle functional point $x < -0.15$, $z > 0.9$; upward long-axis cosine > 0.85.	Recorded stage 1, then bottle origin inside the specified bin XY box and $0.2 < z < 0.7$; bin condition holds at success.
`click_bell_press_stapler`	Bell alignment and gripper proximity to its press point.	Recorded stage 1, then proximity to the stapler press point; stage 2 is latched.
`stapler_pad_press`	Stapler within $(0.055, 0.035, 0.02)$ of pad origin; both grippers open.	Recorded stage 1, then stapler press-point proximity while still at the pad; stage 2 is latched.
`serve_teabox_then_ring`	Supported placement of tea box in basket.	Later bell press; placement retained at success.
`pack_toycar_then_ring`	Supported placement of toy car in box.	Later bell press; placement retained at success.
`attr_teabox_basket_then_ring`	Select tea box amid a distractor; supported placement in basket.	Later bell press; placement retained at success (D6).
`lift_pot_to_pad`	Maximum pot-origin rise exceeds 0.06; stored as a historical condition.	Final XY offset from pad below $(0.09, 0.09)$, pot above pad, grippers open. Uses a stored maximum height.

A.3 Initial States and Instructions

A scripted expert checks task feasibility before initial states are fixed for evaluation. These checks support evaluation construction; expert demonstrations for training on the evaluation scenes are deliberately not part of the release.

The evaluated system starts from a reset scene with an instruction sampled from the task's configured template pool.

The organizers retain the corresponding seeds privately to protect public-evaluation fairness. Evaluation reuses the organizer-managed private seed lists. Trial counts, environment settings, and the matched subset are specified in Appendix B.

A.4 Native Success Predicates

For shared scene predicates, a movement goal requires target displacement above 5 cm and greater than distractor displacement. The first object crossing the threshold must be the target; simultaneous crossings are resolved by greatest displacement. Tools and goal references are excluded from the distractor set. Reference objects must remain within 5 cm of their initial positions.

Press and beat goals use location/contact conditions for the target and reject distractor displacement of 5 cm or more.

Lifting checks object clearance and gripper contact. Placement checks relative location, gripper release, support/container contact, and low object speed (below 0.08 m/s linear and 1 rad/s angular). Containment also checks container orientation and geometry. Relative placement uses signed offsets and scene-specific tolerances. Pressing uses gripper proximity to the designated location; hammer goals require tool–target contact.

Task-specific scenes define their own selection guards. Bottle selection uses initial displacement; soap/box selection uses lifting. The evaluator records success when the task predicate is first satisfied.

A.5 Success Conditions for Ordered Tasks

Ordered tasks advance through their predicates sequentially. Final success also requires the remaining placement obligations, target-selection guards, and reference stability. Table 4 lists task-specific conditions.

B EVALUATION CONFIGURATION

Evaluation uses the ALOHA–AgileX embodiment with head and two wrist cameras. Background, lighting, table-height, and camera-distance randomization are disabled; object placement and assets follow the scene definition. Budgets are 600 native action calls for `move_cup_pad`, `place_a2b_left_novel`, `attr_bind_bottle`, `attr_bind_soap_box`, `adjust_bottle_dustbin`, `click_bell_press_stapler`, and `stapler_pad_press`; the other 41 scenes use 1,000 native action calls. Each call submits one target to the simulator controller, which can execute multiple physical steps; these are not model-inference calls. The evaluator counter is also subject to the agent termination behavior described below.

B.1 CHECKPOINTS AND TRAINING RESOURCES

Table 5 identifies the evaluated checkpoints and their implementations.

Table 5: Weights and implementations for the twelve evaluated policy checkpoints.

Model	Weights (Hugging Face)	Code (GitHub)
Evaluated: twelve checkpoints		
DM0.5	`Dexmal/DM05-robotwin2`	`dexmal/opendm`
$\pi_{0.5}$	`motus-robotics/pi0.5_robotwin2`	`Physical-Intelligence/openpi`
StarVLA	`StarVLA/Qwen3-VL-OFT-RoboTwin2-All`	`starVLA/starVLA`
FastWAM	`yuanty/fastwam`	`yuantianyuan01/FastWAM`
π_0	`yuffish/robotwin-pi0_aloha-agilex_clean-50`	`Physical-Intelligence/openpi`
AHA-WAM	`SereneC/AHA-WAM-RoboTwin2.0`	`serene-sivy/AHA-WAM`
X-VLA	`2toINF/X-VLA-RoboTwin2`	`2toINF/X-VLA`
WLA	`SJTU-DENG-Lab/wla_robotwin_all_image_action`	`SJTU-DENG-Lab/WLA`
GalaxeaVLA (G0.5)	`OpenGalaxea/G05`	`OpenGalaxea/GalaxeaVLA`
X-WAM	`sharinka0715/X-WAM-checkpoints`	`sharinka0715/X-WAM`
Motus	`motus-robotics/Motus_robotwin2`	`thu-ml/Motus`
LingBot-VA	`robbyant/lingbot-va-posttrain-robotwin`	`Robbyant/lingbot-va`

FastWAM uses `robotwin_uncond_3cam_384.pt`; AHA-WAM uses `robotwin_ahawam.pt` with its normalization statistics; X-WAM uses `robotwin_sft`; and G0.5 uses its RoboTwin release with the corresponding action tokenizer and processor.

StarVLA denotes Qwen3-VL-OFT. In Table 1, * denotes robot/latent-action pretraining followed by RoboTwin adaptation; † denotes RoboTwin action-policy training from pretrained vision/language/video backbones, with no additional robot-action pretraining reported; ‡ denotes unverified checkpoint lineage. The $*$ group does not imply that pretraining excludes RoboTwin.

Training provenance distinguishes backbone pretraining, robot or latent-action pretraining, and RoboTwin adaptation. DM0.5, G0.5, Motus, X-WAM, and LingBot-VA report robot or latent-action pretraining; the evaluated $\pi_{0.5}$ adapts PI's base checkpoint. StarVLA, FastWAM, AHA-WAM, and WLA report RoboTwin action training from pretrained backbones. Parent robot checkpoints remain unspecified for π_0 and X-VLA.

Model releases report 50 clean and 500 randomized demonstrations per RoboTwin task for $\pi_{0.5}$, StarVLA, WLA, G0.5, FastWAM, AHA-WAM, Motus, X-WAM, and LingBot-VA. Full quotas are unspecified for DM0.5, π_0, and X-VLA. X-WAM's pretraining mixture includes RoboTwin (Guo et al., 2026).

B.2 Policy Interfaces and Inference Settings

Policies receive three RGB views with model-dependent state and history inputs. Table 6 gives state and action interfaces. Joint vectors contain six arm joints and one gripper value per side. End-effector (EEF) vectors use $[\mathrm{xyz}, \mathrm{rot6D}, \mathrm{grip}] \times 2$ or $[\mathrm{xyz}, \mathrm{quat}, \mathrm{grip}] \times 2$. Model preprocessing and simulator action conversion are specified separately.

Table 6: Control interfaces. State dimensions are per step; simulator actions are specified after adapter conversion.

Policy	Control	Proprioceptive state	Simulator action
DM0.5, $\pi_{0.5}$, π_0, FastWAM, AHA-WAM, G0.5, Motus	Joint	14-D joint positions and gripper values	Chunks of 14-D absolute joint/gripper targets
StarVLA	Joint	No proprioceptive input to the model; bridge forwards RGB and instruction	Chunks of 14-D absolute joint/gripper targets
X-VLA	EEF	20-D: xyz + rot6D + grip, two arms	16-D: xyz + quaternion + grip, two arms
WLA, X-WAM	EEF	16-D: xyz + quaternion + grip, two arms	16-D: xyz + quaternion + grip, two arms
LingBot-VA	EEF	Adapter transports 14-D joint state; model-side conditioning unverified	16-D EEF targets recovered from 16-D actions relative to the initial pose

Joint targets use trajectory time parameterization; EEF targets use cuRobo planning (Sundaralingam et al., 2023). LingBot-VA's relative EEF actions are converted to absolute targets using the initial pose.

Table 7 summarizes checkpoint and adapter configurations. Prediction and execution lengths count action targets per request. These configurations have not been independently pinned to every evaluated episode.

Table 7: Checkpoint and adapter inference settings. Prediction/execution lengths count action targets per request. FM: flow matching. Configuration scope is defined in Appendix B.2.

Policy	Action generation	Prediction / execution	Inference path and context
DM0.5	Flow matching	50 / 50	No explicit future-image sampling.
$\pi_{0.5}$	Flow matching	32 / 32	Current observation and discretized state.
π_0	Flow matching	50 / 50	Current observation; parent checkpoint and training quota unverified.
StarVLA	MLP / ℓ_1 regression	50 / 50	RGB and instruction only at the model boundary; continuous regression from token hidden states.
X-VLA	Flow matching	30 / 30	EEF state and actions; motion planning follows action decoding.
WLA	Flow matching	32 / 32	Historical head image; action sampling bypasses World Expert and subtask-text generation.
G0.5	Flow matching	32 / 8	Continuous actions enabled; autoregressive actions and CoT disabled.
FastWAM	Action denoising	32 / 24	Video co-training; evaluated uncond path skips future-video generation.

Continued on next page.

Table 7 continued: checkpoint and inference configuration.

Policy	Action generation	Prediction / execution	Inference path and context
AHA-WAM	Video/action FM	16 / 16	Video prefill every two action chunks; total configured horizon 64.
Motus	Joint video/action FM	16 / 16	Inference returns predicted frames and actions; full action chunk executed.
X-WAM	Asynchronous denoising	32 / 32	Action-completion early stop enabled; depth branch disabled.
LingBot-VA	Causal video/action FM	Variable	Cross-chunk KV updates using execution observations; lengths depend on returned frame/action dimensions.

The following details complement the interface and inference tables. AHA-WAM, FastWAM, and Motus arrange the head view above the two wrist views, yielding a composite model input of 384×320 pixels (height $\times$ width). AHA-WAM and FastWAM resize the views before concatenation; Motus concatenates them and then resizes with padding as needed. WLA resizes images to 224 and applies JPEG encoding. The evaluated π_0 uses OpenPI JAX; $\pi_{0.5}$ uses OpenPI inference with its RoboTwin-adapted checkpoint. X-VLA uses a 0.9B RoboTwin checkpoint with Florence-2-large, and StarVLA uses a Qwen3-VL-4B backbone. X-WAM converts incremental EEF predictions into absolute targets. FastWAM's serving interface accepts the instruction, but model-internal text conditioning remains unverified; LingBot-VA's model-internal state conditioning is likewise unverified.

B.3 Agent Observations, Tools, and Budgets

The website separates learned action-generating policies from agents that use an explicit language-model tool/controller interface.

Table 8 specifies API identifiers, reasoning settings, token caps, and request handling. Both agents use the provider-specific high reasoning setting: `reasoning.effort=high` for GPT-6 and `thinkingLevel=HIGH` for Gemini. Service identifiers do not pin immutable provider weight versions.

Table 8: Agent request settings.

Setting	GPT-6	Gemini 3.8 Flash
API model	`gpt-6-astra`	`gemini-3.8-flash`
API interface	Responses	`generateContent`
Reasoning setting	`reasoning.effort=high`	`thinkingLevel=HIGH`
Output-token cap	4,096	8,192
Tool constraint	Required; parallel calls disabled	Function-calling mode `ANY`
Reasoning state	Encrypted reasoning content returned; `store=false`	`thoughtSignature`; error fallback removes signature, then uses stateless context
Prompt caching	Episode-specific cache key	Provider implicit caching

Both agents receive three RGB views, measured EEF and joint/gripper state, and contact/tracking feedback. The tools are `move_eef` and `finish`. The controller retains text/tool history within an episode and uses only the current camera images. Budgets are 120 model calls, a soft 3,300 s wall limit checked between requests, and the task's action-call limit.

B.4 Agent Control and Termination

The shared prompt specifies world-frame absolute EEF targets, scalar-first quaternions, a 0.12 m EEF-to-gripper offset, and measured versus commanded gripper aperture.

The EEF controller limits transit, approach, and contact commands to 45 cm/90°, 12 cm/30°, and 3 cm/10°, respectively. Gripper changes occur at a fixed pose; recovery allows up to 3 cm translation.

Accepted commands are interpolated, with early stopping on unexpected contact or tracking error above 2.5 cm/10°. Rejected commands return feedback without advancing simulation.

Requests may be retried after API errors. The finish channel includes both model-requested stopping and controller-triggered stopping after repeated request failure. It ends an attempt and does not necessarily claim successful completion; task success is determined by the environment.

Table 9: Mutually exclusive recorded termination reasons for the 240 accepted episodes per agent in Table 2. Cells give counts and percentages. Finish-channel termination does not identify a success claim or its initiator.

Recorded termination	GPT-6	Gemini 3.8 Flash
Environment success	185 (77.08%)	58 (24.17%)
Finish channel (`model_finish`)	46 (19.17%)	120 (50.00%)
Wall budget (3,300 s)	6 (2.50%)	62 (25.83%)
Scene step limit	3 (1.25%)	0 (0.00%)
Call budget (120 calls)	0 (0.00%)	0 (0.00%)
Total	240 (100.00%)	240 (100.00%)

Table 9 reports termination frequencies over 240 episodes per agent. All 185 GPT-6 and 58 Gemini successes terminate on environment success.

B.5 Matched Evaluation Subset

Each scene has 100 distinct saved seeds for the full policy evaluation. The cross-system subset uses the first five entries of each scene's fixed seed list, yielding 240 episodes per system and 45/145/50 episodes at L1/L2/L3. Success and timing use the same episodes. Table 11 retains all eleven policies with recorded timing and both agents; the main-text table shows G0.5, $\pi_{0.5}$, and Motus as illustrative comparisons, not representatives of class-wide effects.

The main-text comparison retains three illustrative policies: G0.5 for its highest full-suite policy Overall, $\pi_{0.5}$ for its low observed runtime, and Motus to connect with the video-based diagnostics.

Initial instruction strings match within the policy group and within the agent group. Agent wording differs from policy wording in 84 of 240 episodes. On `adjust_bottle_dustbin` and `stapler_pad_press`, agents receive the full compound instruction from the start, while policies receive stage-dependent instructions updated after the first stage succeeds.

Sensitivity to instruction-string differences. We rescore the existing matched records without rerunning any system. Exact initial instruction strings agree across groups in 156 episodes spanning 40 tasks. All ten episodes from the two sequence tasks belong to the 84 mismatches, so excluding both mismatches and sequence tasks still retains 156 episodes. Excluding only the sequence tasks retains 230 episodes across 46 tasks. Table 10 reports Pooled success with explicit denominators; these restricted samples have different task mixtures and do not replace the original Overall score. Text matching does not equalize observations, feedback, controllers, budgets, or hardware.

B.6 Timing Definitions and Additional Summaries

For episode i, let C_i be model-call count, M_i total model-side time, V_i total evaluation-call time, and W_i episode wall time. Tables 2 and 11 report $\sum_i M_i / \sum_i C_i$ and episode means of C_i, M_i, V_i, and W_i, including failures and timeouts.

Policy model-side time includes service-side preprocessing; agent model-side time uses the returned API duration, with request wall time as fallback. Evaluation-call time includes inference, action execution, and rendering. Episode wall time additionally includes observation retrieval and loop overhead, excluding reset and final file writing. These boundaries define system-level timing rather than isolated model compute.

Table 10: Pooled success on descriptive subsets of the existing 240 matched episodes. Identical and different refer to initial instruction strings across the policy and agent groups. Sequence denotes the two tasks with stage-dependent policy instructions. All sequence episodes are in the different-text subset. Each entry gives successes/episodes and a percentage; no interface equivalence is implied.

System	Identical text	Different text	Non-sequence	Sequence
G0.5	57/156 (36.54%)	50/84 (59.52%)	98/230 (42.61%)	9/10 (90.00%)
Motus	46/156 (29.49%)	42/84 (50.00%)	78/230 (33.91%)	10/10 (100.00%)
$\pi_{0.5}$	49/156 (31.41%)	31/84 (36.90%)	72/230 (31.30%)	8/10 (80.00%)
StarVLA	35/156 (22.44%)	32/84 (38.10%)	58/230 (25.22%)	9/10 (90.00%)
FastWAM	38/156 (24.36%)	25/84 (29.76%)	58/230 (25.22%)	5/10 (50.00%)
AHA-WAM	47/156 (30.13%)	36/84 (42.86%)	77/230 (33.48%)	6/10 (60.00%)
WLA	35/156 (22.44%)	26/84 (30.95%)	53/230 (23.04%)	8/10 (80.00%)
X-VLA	29/156 (18.59%)	31/84 (36.90%)	52/230 (22.61%)	8/10 (80.00%)
π_0	19/156 (12.18%)	13/84 (15.48%)	27/230 (11.74%)	5/10 (50.00%)
X-WAM	61/156 (39.10%)	36/84 (42.86%)	91/230 (39.57%)	6/10 (60.00%)
LingBot-VA	62/156 (39.74%)	36/84 (42.86%)	89/230 (38.70%)	9/10 (90.00%)
GPT-6	121/156 (77.56%)	64/84 (76.19%)	180/230 (78.26%)	5/10 (50.00%)
Gemini 3.8 Flash	45/156 (28.85%)	13/84 (15.48%)	58/230 (25.22%)	0/10 (0.00%)

Table 11: Complete success and timing comparison on the first five seeds per scene (240 episodes per system, including failures). Overall weights families equally; Pooled weights episodes equally. Bold marks groupwise maxima for success and minima for time, including ties. *Instruction-delivery differences: Appendix B.5. †Agent counters are not executed-step counts; timing and counter definitions: Appendix B.6.

System	Overall↑	Pooled↑	Calls (/ep.)	Model↓ (s/call)	Model↓ (s/ep.)	Eval wall↓ (s/ep.)	Episode wall (s/ep.)↓	Step counter
Policies								
GalaxeaVLA (G0.5)	**42.30%**	**44.58%**	80.9	1.313	106.2	448.8	487.2	609
LingBot-VA	39.31%	40.83%	25.5	6.365	162.1	604.0	615.3	635
X-WAM	37.33%	40.42%	25.2	1.620	40.8	506.6	519.7	636
Motus	35.14%	36.67%	45.2	6.578	297.1	661.3	683.7	650
AHA-WAM	32.03%	34.58%	46.6	1.155	53.8	516.8	545.9	675
$\pi_{0.5}$	31.92%	33.33%	26.3	**0.226**	6.0	**346.2**	**359.6**	699
StarVLA	26.37%	27.92%	18.2	0.285	5.2	421.3	433.1	740
FastWAM	24.23%	26.25%	34.1	0.928	31.6	489.4	510.8	737
WLA	24.18%	25.42%	27.0	0.884	23.9	971.5	986.3	749
X-VLA	24.28%	25.00%	29.6	0.701	20.7	958.4	980.4	782
π_0	12.76%	13.33%	19.4	0.229	**4.4**	476.0	488.0	886
Agents								
GPT-6*	**76.35%**	**77.08%**	24.9	**13.248**	**329.4**	**710.3**	**730.4**	942†
Gemini 3.8 Flash*	24.35%	24.17%	35.2	46.260	1630.3	1983.9	2002.2	942†

With $N = 240$ episodes, the p50/p95 values are order statistics of ascending-sorted episode mean call times $M_i / \max(1, C_i)$ at zero-based indices $\lfloor N/2 \rfloor$ and $\lfloor 0.95N \rfloor$. Model-side share is $100 \sum_i M_i / \sum_i V_i$.

The step column retained in Table 11 reports the evaluator's action counter. Agent finish and budget termination set this counter to the step limit, so agent values do not count actual executed actions.

DM0.5 is omitted from the timing comparison because its timing was not recorded. Policy evaluation uses shared GPU/CPU resources, with warm-up included. Hardware and worker placement are not matched across systems; timing characterizes the deployed evaluation pipelines. Table 12 reports additional timing and level summaries on the matched subset.

Table 12: Additional summaries on the matched 240-episode subset, with L1/L2/L3 counts of 45/145/50. p50/p95 summarize episode mean call times; Model share is total model-side time divided by evaluation-call time. Bold marks the best directed value within each group. *Agent instruction delivery differs as specified in Appendix B.5; timing definitions are in Appendix B.6.

System	L1↑	L2↑	L3↑	p50↓ (s/call)	p95↓ (s/call)	Model share (%)
Policies						
GalaxeaVLA (G0.5)	**71.11%**	**43.45%**	**24.00%**	1.007	2.369	23.7%
LingBot-VA	55.56%	42.07%	**24.00%**	6.011	8.879	26.8%
X-WAM	64.44%	38.62%	**24.00%**	1.050	3.518	8.0%
Motus	44.44%	**43.45%**	10.00%	2.975	13.940	44.9%
AHA-WAM	48.89%	38.62%	10.00%	1.186	1.991	10.4%
$\pi_{0.5}$	37.78%	37.93%	16.00%	**0.115**	0.905	1.7%
StarVLA	48.89%	28.97%	6.00%	0.211	0.648	1.2%
FastWAM	37.78%	28.97%	8.00%	0.837	1.736	6.5%
WLA	35.56%	28.97%	6.00%	0.740	1.334	2.5%
X-VLA	46.67%	24.14%	8.00%	0.553	1.368	2.2%
π_0	22.22%	14.48%	2.00%	0.198	**0.461**	0.9%
Agents						
GPT-6*	**80.00%**	**78.62%**	**70.00%**	**11.590**	**15.933**	46.4%
Gemini 3.8 Flash*	31.11%	25.52%	14.00%	44.306	100.458	82.2%

C Global Level and Complete-Family Summaries

Let s_c be the success fraction for task c, and let $\mathcal{C}_d$ contain the tasks in family d. We average tasks within each family, then weight the six families equally:

$$S_d = \frac{1}{|\mathcal{C}_d|} \sum_{c \in \mathcal{C}_d} s_c, \qquad S_{\text{overall}} = \frac{1}{6} \sum_{d=1}^{6} S_d. \tag{1}$$

Global level rates average tasks within each level, and complete-family means average tasks within each family. Tables 13 and 14 report these complementary summaries; Overall and Pooled appear in Table 1.

Table 13: Global level success (%) for twelve policies, with 100 episodes per task. L1/L2/L3 average their 9/29/10 tasks equally. Bold marks column maxima, including ties. Overall and Pooled appear in Table 1; per-task intervals appear in Appendix F. Training conditions differ across checkpoints.

Policy	L1↑	L2↑	L3↑
DM0.5	57.56%	39.69%	6.90%
$\pi_{0.5}$	37.67%	35.90%	18.70%
StarVLA (Qwen3-VL-OFT)	43.78%	31.24%	11.00%
FastWAM	46.33%	26.97%	7.60%
π_0	20.67%	15.10%	1.60%
AHA-WAM	55.33%	31.69%	8.40%
X-VLA	38.11%	22.97%	6.70%
WLA	39.89%	29.21%	7.60%
GalaxeaVLA (G0.5)	60.67%	**44.45%**	22.30%
X-WAM	**67.00%**	39.55%	18.40%
Motus	55.11%	39.24%	14.20%
LingBot-VA	56.11%	39.55%	**26.40%**

Table 14: Mean task success (%) within each family. D1–D6 contain 10/9/9/7/6/7 equally weighted tasks; Overall equally weights the six unrounded family means. Bold marks column maxima, including ties. These are completion rates, not attribute-selection rates.

Policy	D1↑	D2↑	D3↑	D4↑	D5↑	D6↑
DM0.5	49.00%	35.22%	59.89%	13.14%	33.17%	14.43%
$\pi_{0.5}$	46.40%	23.78%	47.56%	18.43%	32.00%	20.00%
StarVLA (Qwen3-VL-OFT)	48.10%	24.44%	40.44%	13.00%	36.00%	5.43%
FastWAM	45.50%	18.67%	50.89%	10.14%	16.33%	3.57%
π_0	24.50%	4.33%	20.33%	9.00%	18.17%	0.14%
AHA-WAM	50.10%	22.67%	61.33%	12.71%	19.83%	5.14%
X-VLA	32.30%	11.78%	33.89%	8.86%	**45.83%**	0.71%
WLA	41.10%	20.89%	41.44%	13.86%	30.50%	4.29%
GalaxeaVLA (G0.5)	57.70%	**44.11%**	**67.44%**	15.00%	34.83%	**23.29%**
X-WAM	**78.30%**	23.56%	60.11%	20.29%	34.33%	7.14%
Motus	51.10%	39.56%	56.89%	13.43%	41.33%	7.86%
LingBot-VA	55.80%	29.11%	60.89%	**25.29%**	39.67%	19.00%

All twelve checkpoints show decreasing global success from L1 to L3. X-WAM leads L1 (67.00%), G0.5 leads L2 (44.45%), and LingBot-VA leads L3 (26.40%).

D ATTRIBUTE SELECTION AND COMPLETION ON FORMAL TASKS

D.1 EVALUATED SYSTEMS AND EVENT DEFINITIONS

The study evaluates $\pi_{0.5}$, Motus, G0.5, X-VLA, StarVLA, GPT-6, and Gemini 3.8 Flash on nine tasks with five fixed seeds each: 45 episodes per system and 315 overall. Instructions, initial poses, object properties, required attributes, and effectors are matched across systems.

Force is the sum of contact-point impulse magnitudes divided by the simulation timestep. Selection is measured at the first physical step with a loaded interaction of at least 0.1 N, using gripper–candidate interactions for seven tasks and hammer–candidate interactions for two. A unique candidate identifies the selected object; simultaneous candidates are ambiguous. These measurements sample every physical step, whereas the separate paired study uses 40 ms telemetry (Appendix E.1). Table 15 reports 252 unique events, 63 no-event outcomes, and no ambiguous or unavailable measurements.

Attribute correctness compares the selected object's annotated properties with those requested by the instruction. Eligibility is fixed before observing behavior: the instruction must request the attribute, and candidate annotations must permit it to be scored. Unrequested attributes are not applicable. A selected object can receive credit for one requested attribute and not another.

For a required attribute a, let $z_{i,a} = 1$ when the selected object's annotated value satisfies the instruction, and zero otherwise. For example, a may denote size, color, or shape; a single object can have multiple attributes, each scored when required by the instruction. With $\mathcal{E}_{c,a}$ denoting the episodes of eligible scene c with complete event records, the per-scene rate, defined when $|\mathcal{E}_{c,a}| > 0$, is

$$S_{c,a} = \frac{1}{|\mathcal{E}_{c,a}|} \sum_{i \in \mathcal{E}_{c,a}} z_{i,a}. \tag{2}$$

For selection and attribute rates, complete event records form the denominator. Ambiguous and no-event outcomes receive zero credit; missing records are reported as unmeasurable rather than treated as wrong choices. Unique-event coverage is the fraction of complete records containing one qualifying selected object. Correctness conditional on a unique event uses only that subset; it is undefined if no unique event exists. Attribute rates are averaged equally over eligible tasks, each of which contributes five trials here.

Selection and completion. An initial selection error may be corrected, while a correct choice may still end in failure. We therefore report the fraction of episodes with $c_i = 1$ and $y_i = 0$, and completion conditional on correct selection. Here c_i indicates correct first-event selection and y_i indicates native task success. Completion conditional on correct selection is $\sum_i c_i y_i / \sum_i c_i$ over complete records and is undefined when $\sum_i c_i = 0$. It describes a model-specific subset, not overall performance. The correct-selection/failure fraction uses all complete records as its denominator. Native task success y_i is evaluated independently of first-event correctness c_i; their joint outcomes distinguish selection from completion without assigning a cause to later failure. Selection and property-level measurements help distinguish cases requiring closer examination of target choice from those requiring examination of what happens after a correct choice.

Table 15: Selection-event coverage, target correctness, and task success across nine tasks, with 45 episodes per system. Correct selection denotes a correct first qualifying selection; Correct/fail counts such selections without completion.

System	Unique event	Correct selection	Native success	Correct/fail (n)	No event
$\pi_{0.5}$	37/45 (82.2%)	25/45 (55.6%)	27/45 (60.0%)	5	8
Motus	39/45 (86.7%)	22/45 (48.9%)	26/45 (57.8%)	2	6
G0.5	32/45 (71.1%)	20/45 (44.4%)	32/45 (71.1%)	3	13
X-VLA	39/45 (86.7%)	20/45 (44.4%)	17/45 (37.8%)	12	6
StarVLA	37/45 (82.2%)	19/45 (42.2%)	17/45 (37.8%)	8	8
GPT-6	37/45 (82.2%)	35/45 (77.8%)	31/45 (68.9%)	10	8
Gemini 3.8 Flash	31/45 (68.9%)	19/45 (42.2%)	16/45 (35.6%)	10	14

Table 16: First-selection and completion counts across the 315 instrumented episodes (seven systems, nine tasks, five trials). The 0.1 N event rule yields 252 unique selections and 63 no-event episodes. Native success is scored independently; correct selection followed by failure accounts for 50/315 episodes.

First selection	Success	Failure	Total
Correct target	110	50	160
Wrong unique target	21	71	92
No event	35	28	63
Total	166	149	315

Table 17: Unique selection events at 0.1 N, with five episodes per cell.

Task	$\pi_{0.5}$	Motus	G0.5	X-VLA	StarVLA	GPT-6	Gemini
Color: blocks	5/5	5/5	5/5	5/5	5/5	5/5	5/5
Color + category: soap/box	5/5	5/5	5/5	5/5	5/5	5/5	4/5
Color: bottle	5/5	5/5	5/5	5/5	5/5	5/5	5/5
Size: pick	5/5	5/5	5/5	5/5	5/5	5/5	2/5
Size: middle rank	5/5	5/5	5/5	5/5	5/5	5/5	5/5
Category: can/box	5/5	5/5	5/5	5/5	5/5	5/5	5/5
Category: bell	0/5	0/5	0/5	4/5	2/5	0/5	0/5
Color: hammer	4/5	5/5	1/5	3/5	4/5	5/5	4/5
Size: hammer	3/5	4/5	1/5	2/5	1/5	2/5	1/5

D.2 Task Coverage and Attribute Results

The nine task identifiers are `attr_color_block_3way`, `attr_bind_soap_box`, `attr_bind_bottle`, `attr_size_block_pick`, `attr_size_rank_middle`, `attr_category_can_plasticbox`, `attr_category_press_bell`, `attr_color_beat_block`, and `attr_size_beat_smaller`. The final two use the hammer effector; the remaining seven use the gripper. Figure 6 reports target-event and native-success counts for every cell, and Table 17 reports the corresponding unique-event coverage.

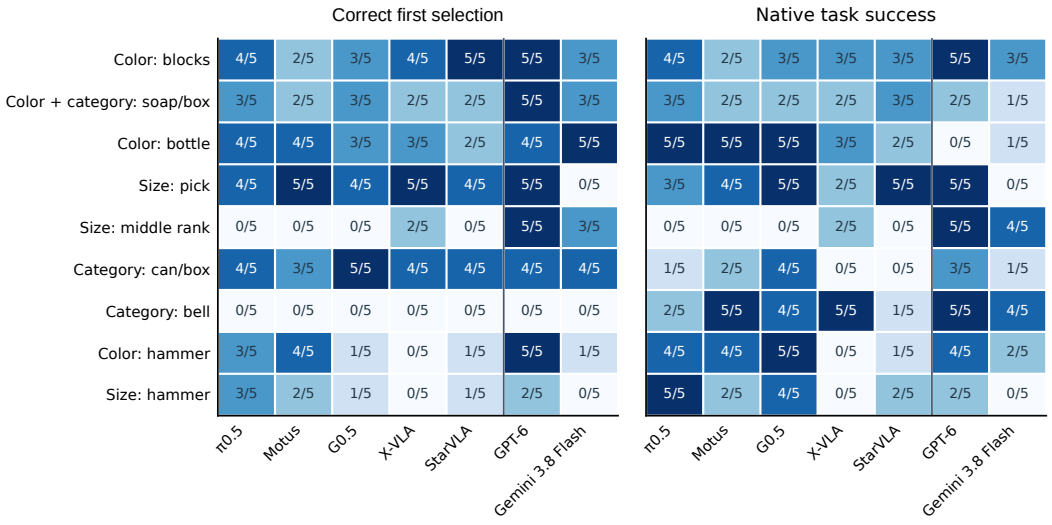

Figure 6: Correct first selection at 0.1 N (left) and native task success (right). Each cell shows the count out of five trials for one system–task pair; darker shading indicates higher rates. No-event trials remain in the denominator.

Table 18: Attribute correctness and unique-event conditional rates. Color, size, and category use four, three, and three tasks respectively, with five episodes per task.

System	Attribute	Correct / eligible	Unique / eligible	Correct / unique
$\pi_{0.5}$	Color	15/20 (75.0%)	19/20 (95.0%)	15/19 (78.9%)
	Size	7/15 (46.7%)	13/15 (86.7%)	7/13 (53.8%)
	Category	8/15 (53.3%)	10/15 (66.7%)	8/10 (80.0%)
Motus	Color	13/20 (65.0%)	20/20 (100.0%)	13/20 (65.0%)
	Size	7/15 (46.7%)	14/15 (93.3%)	7/14 (50.0%)
	Category	7/15 (46.7%)	10/15 (66.7%)	7/10 (70.0%)
G0.5	Color	10/20 (50.0%)	16/20 (80.0%)	10/16 (62.5%)
	Size	5/15 (33.3%)	11/15 (73.3%)	5/11 (45.5%)
	Category	10/15 (66.7%)	10/15 (66.7%)	10/10 (100.0%)
X-VLA	Color	11/20 (55.0%)	18/20 (90.0%)	11/18 (61.1%)
	Size	7/15 (46.7%)	12/15 (80.0%)	7/12 (58.3%)
	Category	7/15 (46.7%)	14/15 (93.3%)	7/14 (50.0%)
StarVLA	Color	11/20 (55.0%)	19/20 (95.0%)	11/19 (57.9%)
	Size	5/15 (33.3%)	11/15 (73.3%)	5/11 (45.5%)
	Category	8/15 (53.3%)	12/15 (80.0%)	8/12 (66.7%)
GPT-6	Color	19/20 (95.0%)	20/20 (100.0%)	19/20 (95.0%)
	Size	12/15 (80.0%)	12/15 (80.0%)	12/12 (100.0%)
	Category	9/15 (60.0%)	10/15 (66.7%)	9/10 (90.0%)
Gemini 3.8 Flash	Color	13/20 (65.0%)	18/20 (90.0%)	13/18 (72.2%)
	Size	3/15 (20.0%)	8/15 (53.3%)	3/8 (37.5%)
	Category	7/15 (46.7%)	9/15 (60.0%)	7/9 (77.8%)

Table 19: Joint color–category selection on the soap/box task, with five episodes per system.

System	Both / exact target	Color only	Category only	No event
$\pi_{0.5}$	3/5 (60.0%)	1/5 (20.0%)	1/5 (20.0%)	0/5 (0.0%)
Motus	2/5 (40.0%)	1/5 (20.0%)	2/5 (40.0%)	0/5 (0.0%)
G0.5	3/5 (60.0%)	0/5 (0.0%)	2/5 (40.0%)	0/5 (0.0%)
X-VLA	2/5 (40.0%)	2/5 (40.0%)	1/5 (20.0%)	0/5 (0.0%)
StarVLA	2/5 (40.0%)	1/5 (20.0%)	2/5 (40.0%)	0/5 (0.0%)
GPT-6	5/5 (100.0%)	0/5 (0.0%)	0/5 (0.0%)	0/5 (0.0%)
Gemini 3.8 Flash	3/5 (60.0%)	1/5 (20.0%)	0/5 (0.0%)	1/5 (20.0%)

Color is evaluated on the three-block, bottle, soap/box, and color-hammer tasks; size on block-pick, middle-rank, and size-hammer tasks; and category on soap/box, can/plastic-box, and bell tasks. The soap/box task contributes to both color and category rates (Table 18).

Table 19 shows that, on the soap/box task, 20/35 episodes select the exact target, six match color only, and eight match category only; one Gemini episode has no qualifying event. In the Motus example, "Raise the brown coffee box" leads to first loaded interaction with brown soap, satisfying color but not category; the task is unsuccessful.

D.3 Threshold Sensitivity and Measurement Boundaries

Figure 7 compares the prespecified 0.01, 0.1, and 1 N thresholds by rescoring the same trajectories. At 0.1 versus 1 N, gripper-task correct-target counts change from 19 to 18 for $\pi_{0.5}$, 16 to 17 for Motus, 18 to 21 for G0.5, 20 to 21 for X-VLA, 17 to 20 for StarVLA, 28 to 26 for GPT-6, and 18 to 19 for Gemini, each out of 35 episodes. Tool-hit counts remain unchanged across the three thresholds.

Threshold sensitivity can arise from incidental contact: one successful G0.5 bottle trial first contacts a distractor at approximately 0.30 N before interacting with the target above 1 N.

Press and beat predicates can register task success through geometric proximity or contact-pair conditions without reaching the loaded-event threshold. This accounts for successful bell and hammer trials with no loaded target interaction.

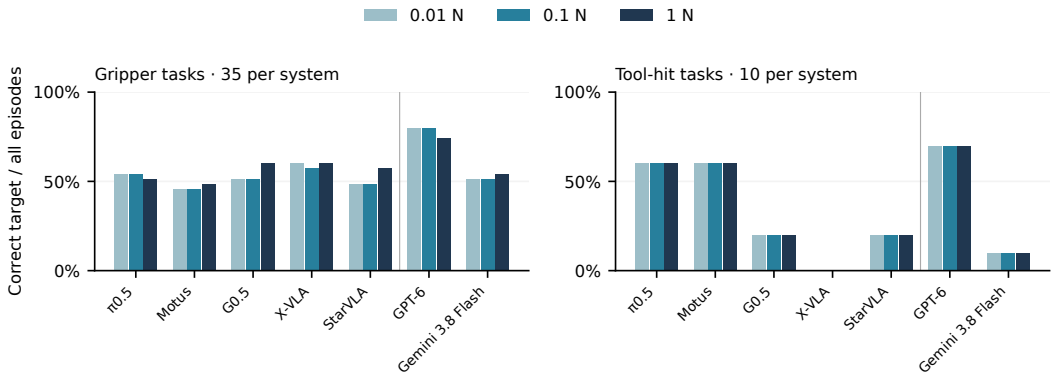

Figure 7: First-interaction target correctness at 0.01, 0.1, and 1 N. Rates use all 35 gripper or 10 tool-hit episodes per system, including no-event trials; the same episodes are rescored at each threshold.

D.4 Agent Evaluation Protocol

Agents use the shared RGB/state observations and EEF tools described in Appendix B.3, with limits of 120 model calls, 3,300 s, and the task's action budget. GPT-6 ends on success in 31 trials, finish in eight, the step limit in five, and the call limit in one. Gemini ends on success in 16, finish in 27, and the wall limit in two. Budget termination counts as task failure.

Infrastructure-interrupted attempts are restarted with the same task and seed; completed task failures are retained. Evaluation uses one completed valid episode per system–task–seed combination.

E Paired Behavioral and Model-Access Diagnostics

The controlled study crosses target instructions and object layouts on the three-color block task for $\pi_{0.5}$ and Motus, with future-video and internal-intervention analyses for Motus. For a policy that also predicts video, we then ask whether the predicted target follows the request, and whether altering future representations changes the relative target-distance preference of predicted actions.

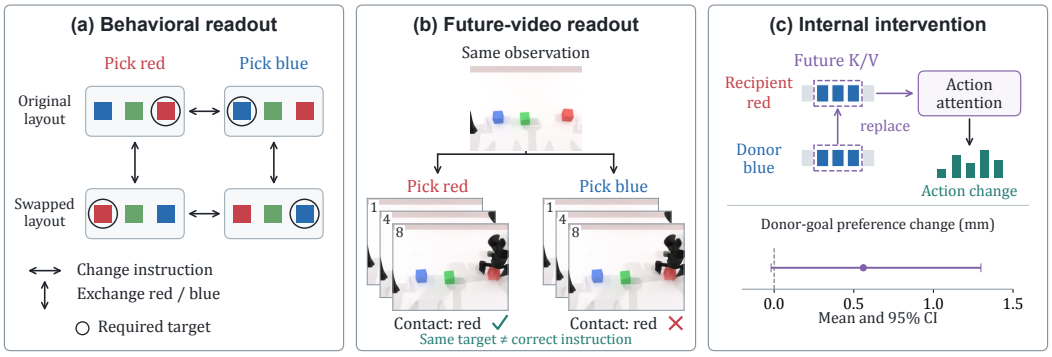

Figure 8: Three diagnostic layers. Left: paired instructions and layouts, with rings marking requested targets. Center: frames 1, 4, and 8 (back to front) from each Motus forecast; final frames and later contacts indicate red under both requests. Right: alternate-instruction future key/value (K/V) representations replace the corresponding entries at the action-attention interface, with other inputs fixed. The plot shows mean donor-goal preference change and its 95% configuration-bootstrap interval, measured from commanded kinematics; blocks and action bars are schematic. Motus results come from Section 5.3.

Table 20: Controlled color-task results: rates and 95% configuration-bootstrap intervals in percent, with counts over 80 trajectories or 40 pairs per checkpoint. Bold marks the higher rate, including ties. These trials are separate from the formal benchmark.

Measurement ↑	$\pi_{0.5}$	Motus
Correct target contact	**86.2%** [77.5, 93.8] (69/80)	40.0% [31.2, 48.8] (32/80)
Both instructions correct	**72.5%** [55.0, 87.5] (29/40)	2.5% [0.0, 7.5] (1/40)
Both layouts correct	**72.5%** [55.0, 87.5] (29/40)	20.0% [10.0, 30.0] (8/40)
Task success	**78.8%** [68.8, 87.5] (63/80)	40.0% [31.2, 48.8] (32/80)

E.1 Paired Conditions and Observable Events

Twenty initial configurations of one three-color block task each define four conditions: original or red–blue-exchanged object poses, crossed with "Pick up the red block." or "Pick up the blue block." The green block and initial robot state remain fixed. The evaluator's target matches the instruction and is hidden from the policy. Both policies receive three RGB views, the instruction, and 14-dimensional commanded joint state. Initial configurations are included independently of policy outcome. Figure 8 illustrates the paired design and an example forecast contrast.

Action chunks contain 32 commands for $\pi_{0.5}$ and 16 for Motus. Runs terminate at task success or 256 native action commands, not 256 model-inference decisions. Inference noise uses seed $17 + d$ at decision d, matching the first decision across conditions.

Loaded contact is the summed gripper–candidate contact-impulse magnitude divided by the 4 ms physical timestep, thresholded at 0.1 N. The first qualifying event in 40 ms telemetry determines the selected object. Ambiguous and no-contact cases receive zero correctness credit. Target-correct counts remain unchanged at 0.01 and 1 N.

Native success checks lift clearance and target selection. An auxiliary approach event requires tool center point (TCP) distance below 0.12 m, a 0.02 m margin over alternatives, 0.03 m approach from initialization, and 0.20 s persistence. Stable holding requires both fingers of one hand in loaded contact, 0.02 m object rise, and 0.20 s persistence.

In the paired behavioral study, each initial configuration s has two valid target instructions $t \in \{r, b\}$ and two layouts $\ell \in \{0, 1\}$, with the red and blue objects physically exchanged. Let $c_{s,\ell,t}$ indicate that the first recorded loaded gripper contact is uniquely with the requested object. No qualifying contact or an ambiguous contact receives zero. Across S initial configurations, alongside contact accuracy and task success, we report

$$P_{\text{instr}} = \frac{1}{2S} \sum_{s,\ell} c_{s,\ell,r} c_{s,\ell,b}, \qquad P_{\text{layout}} = \frac{1}{2S} \sum_{s,t} c_{s,0,t} c_{s,1,t}. \tag{3}$$

Table 20 reports the paired and per-episode results. Layout-paired correctness requires the requested color to be selected after the objects exchange positions. Paired rates average the two instruction or layout pairs within each initial configuration, then average configurations. Confidence intervals use 20,000 bootstrap samples of configurations, retaining all four conditions, and the 2.5th and 97.5th percentiles. Conditional completion is 63/69 for $\pi_{0.5}$ and 32/32 for Motus.

Table 21 shows a descriptive red preference for Motus: 48/80 contacts are red, 13 blue, and 19 green. The same contacted color persists under both instructions in 38/40 pairs, versus 9/40 for $\pi_{0.5}$. Across all four conditions within an initial configuration, Motus is correct in four/three/two/one/zero conditions in 0/1/13/3/3 configurations; the corresponding counts for $\pi_{0.5}$ are 12/5/3/0/0. These patterns do not establish a universal color preference or a causal position bias.

E.2 Output Sensitivity and Future-Video Readout

Figure 9 summarizes first-decision action sensitivity to instruction, layout, and inference-noise changes. To measure first-decision action sensitivity, we change the target instruction while holding the observation, state, and inference noise fixed, or exchange object poses while holding the instruction and inference noise fixed. The noise control changes inference seed 17 to 29. RMS differences

Table 21: Checkpoint-specific color-selection outcomes by layout and requested color, with 20 trials per row. Red/blue/green columns count first-contact colors, followed by target-correct and native-success counts. Swapped exchanges the red and blue poses while retaining the green pose.

Checkpoint	Layout	Request	Red	Blue	Green	Correct	Success
$\pi_{0.5}$	Original	red	18	0	2	18/20	17/20
$\pi_{0.5}$	Original	blue	3	17	0	17/20	15/20
$\pi_{0.5}$	Swapped	red	19	1	0	19/20	17/20
$\pi_{0.5}$	Swapped	blue	5	15	0	15/20	14/20
Motus	Original	red	13	1	6	13/20	13/20
Motus	Original	blue	13	1	6	1/20	1/20
Motus	Swapped	red	12	5	3	12/20	12/20
Motus	Swapped	blue	10	6	4	6/20	6/20

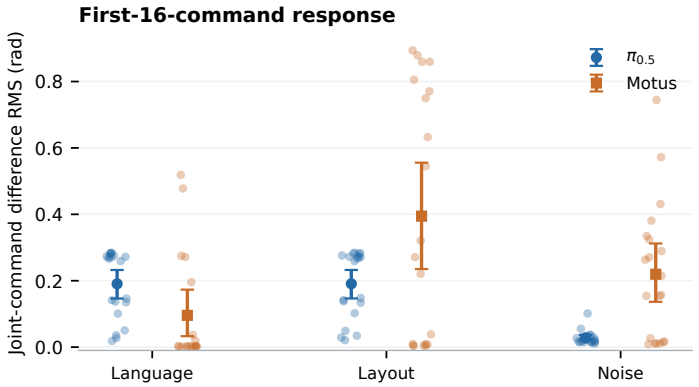

Figure 9: First-decision action sensitivity to instruction, layout, and noise changes, measured by joint-command difference RMS over the first 16 commands. Faint points show 20 configuration means; solid markers and error bars show overall means and 95% configuration-bootstrap intervals. Internal interventions are reported in Table 22.

between paired joint-command predictions use the first 16 commands and exclude grippers. Language/layout/noise configuration means are 0.191/0.191/0.028 rad for $\pi_{0.5}$ and 0.096/0.394/0.220 rad for Motus.

Motus predicts eight video frames and 16 action commands jointly. The future readout uses the last predicted head-camera frame. A task-specific RGB rule identifies the gripper and three block colors, requiring a nearest candidate within 12 pixels and a margin of at least 6 pixels. It is defined using the first initial configuration. Visual review checks all 80 conditions, with disagreements left unresolved; labels lack independent blinded human validation.

The readout identifies 77/80 targets, of which 31 match the instruction; both instructions are correct in 1/40 pairs. All 77 readable targets match later first contact, including 46 wrong selections. First contacts occur after the first action chunk, beyond the first-decision forecast horizon. Three unresolved forecasts remain in the instruction-correctness denominator. This makes forecast-target correctness a necessary companion to agreement when diagnosing these cases: agreement alone would leave the wrong goal unchallenged.

An auxiliary "Please pick up…" wording control yields action RMS of 0.005 rad for $\pi_{0.5}$ and 0.023 rad for Motus on original-layout inputs. Automated Motus readouts yield 1/40 correct instruction pairs under changed noise and 0/20 under the wording change. These auxiliary labels use the image rule alone. Language-induced final-frame RGB mean absolute error (MAE) is 0.015 on $[0, 1]$ intensities.

Table 22: Motus interventions on 80 first-decision conditions from 20 initial configurations: means and 95% configuration-bootstrap intervals. Donor-goal preference change measures relative target distances under commanded kinematics, not executed displacement or success. Matched denotes nominal key/value (K/V) perturbation-norm matching.

Intervention	Joint-command difference RMS (rad)	Donor-goal preference change (mm)
Original-value writeback	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]
Full instruction change	0.096 [0.033, 0.173]	2.529 [0.734, 5.339]
Future replacement	0.030 [0.013, 0.051]	0.560 [-0.019, 1.299]
Future, half strength	0.010 [0.005, 0.015]	0.041 [-0.190, 0.257]
Current replacement	0.007 [0.003, 0.012]	0.157 [0.017, 0.347]
Understanding replacement	0.006 [0.003, 0.011]	0.058 [-0.366, 0.430]
Future, matched	0.030 [0.013, 0.051]	0.544 [-0.043, 1.292]
Current, matched	0.015 [0.006, 0.025]	0.357 [0.090, 0.748]
Understanding, matched	0.080 [0.030, 0.142]	1.330 [-1.560, 4.050]
Future, scrambled + matched	0.003 [0.002, 0.005]	0.048 [-0.041, 0.162]

E.3 Internal Interventions: Design, Controls, and Results

For each of 80 Motus conditions, the donor uses the alternative instruction with observation, state, and noise fixed. At all 30 joint-attention layers and ten denoising steps, K/V entries for 120 tokens in the first future temporal group are replaced where action queries read them; only action-expert output rows are updated. Later computation can propagate the intervention. Token groups may contain mixed information after attention, so intervention effects are specific to the replaced representations (Zhang & Nanda, 2024).

Controls comprise original-value writeback, the full donor instruction, current-video and understanding replacements, half-strength future replacement, and norm-matched or scrambled perturbations. K and V norms are matched separately at each layer and denoising step. Sham replacements reproduce baseline predictions. The internal analysis is exploratory, including a swapped-layout extension selected after initial inspection.

Forward kinematics maps each predicted joint command q_k to the two TCP positions $p_h(q_k)$, $h \in \{L, R\}$. For target g at its initial position x_g, define $D_g(q_k) = \min_h \|p_h(q_k) - x_g\|_2$. Each target independently uses the nearer arm. For intervention m and baseline 0, donor-goal preference change is

$$\Delta^{(m)}_{\text{pref}} = \frac{1}{16} \sum_{k=1}^{16} \Big[\big(D_{\text{rec}}(q_k^{(m)}) - D_{\text{don}}(q_k^{(m)}) \big) - \big(D_{\text{rec}}(q_k^{(0)}) - D_{\text{don}}(q_k^{(0)}) \big) \Big].$$

Here rec and don denote the recipient and donor instruction targets. Distances are converted to millimeters for reporting. Positive values indicate a relative preference change toward the donor; they do not necessarily mean that donor-target distance decreases, and do not measure executed displacement or success.

Table 22 reports the intervention results. Near-future replacement produces a mean joint-command RMS difference of 0.030 rad and a donor-goal preference change of 0.560 mm (95% interval: $[-0.019, 1.299]$ mm). Donor forecasts match their instruction in 31/80 conditions. Two recipient–donor contrasts from one initial configuration predict different readable targets; their relative distance-preference change toward the donor forecast target is -4.432 mm.

After BF16 arithmetic, 100.00%, 96.32%, 100.00%, and 99.98% of applied perturbation norms fall within 10% of the reference for matched future, current, understanding, and scrambled future replacements, respectively.

E.4 Diagnostic Interpretation

The combined readouts suggest concrete next checks—whether target choices track alternative requests, whether forecasts already select the wrong target, and whether representation changes favor the intended target in predicted actions.

Joint failure to switch forecasts and actions motivates examination of conditioning and target binding. Correct forecasts with unresponsive actions motivate testing the connection to action generation, while correct selection followed by failure motivates execution analysis. These patterns guide subsequent experiments.

F Complete Task Results

Tables 23, 24, and 25 report per-task policy success and 95% Wilson intervals; Table 26 gives the two agents.

F.1 Policy Task Results

Table 23: Per-task policy success (%), part 1 of 3: 100 trials per task, with 95% Wilson intervals in parentheses. Bold marks taskwise maxima across all twelve policies in the three parts, including ties. Intervals describe per-task success, not Overall; task families and execution limits appear in Appendices A.1 and B.

Scene	DM0.5↑	$\pi_{0.5}$↑	StarVLA↑	FastWAM↑
grab_dumbbell	76% (66.8, 83.3)%	56% (46.2, 65.3)%	42% (32.8, 51.8)%	52% (42.3, 61.5)%
move_can_coaster	0% (0.0, 3.7)%	7% (3.4, 13.7)%	45% (35.6, 54.8)%	31% (22.8, 40.6)%
move_cup_pad	**99%** (94.6, 99.8)%	68% (58.3, 76.3)%	93% (86.3, 96.6)%	67% (57.3, 75.4)%
place_teabox_scale	56% (46.2, 65.3)%	21% (14.2, 30.0)%	36% (27.3, 45.8)%	43% (33.7, 52.8)%
grasp_mallet_novel	90% (82.6, 94.5)%	97% (91.5, 99.0)%	94% (87.5, 97.2)%	94% (87.5, 97.2)%
lift_trophy	54% (44.3, 63.4)%	66% (56.3, 74.5)%	74% (64.6, 81.6)%	49% (39.4, 58.7)%
place_seal_coaster	3% (1.0, 8.5)%	34% (25.5, 43.7)%	42% (32.8, 51.8)%	28% (20.1, 37.5)%
place_teabox_skillet	0% (0.0, 3.7)%	5% (2.2, 11.2)%	13% (7.8, 21.0)%	0% (0.0, 3.7)%
place_teabox_stand	57% (47.2, 66.3)%	39% (30.0, 48.8)%	1% (0.2, 5.4)%	45% (35.6, 54.8)%
place_toycar_plasticbox	55% (45.2, 64.4)%	71% (61.5, 79.0)%	41% (31.9, 50.8)%	46% (36.6, 55.7)%
place_a2b_left_novel	**75%** (65.7, 82.5)%	46% (36.6, 55.7)%	59% (49.2, 68.1)%	46% (36.6, 55.7)%
place_a2b_right_novel	**65%** (55.3, 73.6)%	41% (31.9, 50.8)%	59% (49.2, 68.1)%	38% (29.1, 47.8)%
card_right_of_scanner	23% (15.8, 32.2)%	12% (7.0, 19.8)%	14% (8.5, 22.1)%	1% (0.2, 5.4)%
phone_right_of_laptop	38% (29.1, 47.8)%	2% (0.6, 7.0)%	0% (0.0, 3.7)%	4% (1.6, 9.8)%
place_can_right_of_plasticbox	58% (48.2, 67.2)%	29% (21.0, 38.5)%	9% (4.8, 16.2)%	37% (28.2, 46.8)%
place_cube_right_of_plate	42% (32.8, 51.8)%	36% (27.3, 45.8)%	18% (11.7, 26.7)%	24% (16.7, 33.2)%
place_teabox_front_of_skillet	0% (0.0, 3.7)%	4% (1.6, 9.8)%	**23%** (15.8, 32.2)%	3% (1.0, 8.5)%
put_toycar_left_basket	0% (0.0, 3.7)%	2% (0.6, 7.0)%	2% (0.6, 7.0)%	1% (0.2, 5.4)%
place_between_bowls	16% (10.1, 24.4)%	42% (32.8, 51.8)%	36% (27.3, 45.8)%	14% (8.5, 22.1)%
attr_bind_bottle	47% (37.5, 56.7)%	46% (36.6, 55.7)%	33% (24.6, 42.7)%	40% (30.9, 49.8)%

Continued on next page.

Table 23 continued: task success (95% Wilson interval).

Scene	DM0.5↑	$\pi_{0.5}$↑	StarVLA↑	FastWAM↑
attr_category_press_bell	**100%** (96.3, 100.0)%	54% (44.3, 63.4)%	18% (11.7, 26.7)%	**100%** (96.3, 100.0)%
attr_category_can_plasticbox	19% (12.5, 27.8)%	4% (1.6, 9.8)%	2% (0.6, 7.0)%	0% (0.0, 3.7)%
attr_color_beat_block	87% (79.0, 92.2)%	69% (59.4, 77.2)%	28% (20.1, 37.5)%	71% (61.5, 79.0)%
attr_color_block_3way	69% (59.4, 77.2)%	83% (74.5, 89.1)%	85% (76.7, 90.7)%	60% (50.2, 69.1)%
attr_size_beat_smaller	**86%** (77.9, 91.5)%	58% (48.2, 67.2)%	59% (49.2, 68.1)%	40% (30.9, 49.8)%
attr_size_block_pick	97% (91.5, 99.0)%	66% (56.3, 74.5)%	79% (70.0, 85.8)%	86% (77.9, 91.5)%
attr_bind_soap_box	33% (24.6, 42.7)%	45% (35.6, 54.8)%	56% (46.2, 65.3)%	61% (51.2, 70.0)%
attr_size_rank_middle	1% (0.2, 5.4)%	3% (1.0, 8.5)%	4% (1.6, 9.8)%	0% (0.0, 3.7)%
ref_block_on_plate	0% (0.0, 3.7)%	0% (0.0, 3.7)%	**9%** (4.8, 16.2)%	0% (0.0, 3.7)%
ref_block_behind_can	2% (0.6, 7.0)%	24% (16.7, 33.2)%	5% (2.2, 11.2)%	0% (0.0, 3.7)%
ref_cup_on_tray	0% (0.0, 3.7)%	0% (0.0, 3.7)%	**22%** (15.0, 31.1)%	0% (0.0, 3.7)%
ref_leftmost_of_four	43% (33.7, 52.8)%	52% (42.3, 61.5)%	23% (15.8, 32.2)%	27% (19.3, 36.4)%
ref_mug_next_to_bowl	32% (23.7, 41.7)%	45% (35.6, 54.8)%	6% (2.8, 12.5)%	17% (10.9, 25.5)%
ref_not_the_red_one	0% (0.0, 3.7)%	1% (0.2, 5.4)%	16% (10.1, 24.4)%	16% (10.1, 24.4)%
ref_second_from_left	15% (9.3, 23.3)%	7% (3.4, 13.7)%	10% (5.5, 17.4)%	11% (6.3, 18.6)%
adjust_bottle_dustbin	94% (87.5, 97.2)%	97% (91.5, 99.0)%	**100%** (96.3, 100.0)%	1% (0.2, 5.4)%
click_bell_press_stapler	7% (3.4, 13.7)%	0% (0.0, 3.7)%	17% (10.9, 25.5)%	3% (1.0, 8.5)%
serve_teabox_then_ring	0% (0.0, 3.7)%	**1%** (0.2, 5.4)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
stapler_pad_press	97% (91.5, 99.0)%	59% (49.2, 68.1)%	99% (94.6, 99.8)%	94% (87.5, 97.2)%
lift_pot_to_pad	0% (0.0, 3.7)%	11% (6.3, 18.6)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
pack_toycar_then_ring	1% (0.2, 5.4)%	24% (16.7, 33.2)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
move_cup_pad_mix	**83%** (74.5, 89.1)%	78% (68.9, 85.0)%	8% (4.1, 15.0)%	24% (16.7, 33.2)%
ref_block_on_plate_mix	0% (0.0, 3.7)%	0% (0.0, 3.7)%	**16%** (10.1, 24.4)%	0% (0.0, 3.7)%
attr_teabox_basket_then_ring	0% (0.0, 3.7)%	0% (0.0, 3.7)%	**2%** (0.6, 7.0)%	0% (0.0, 3.7)%
card_right_of_scanner_mix	11% (6.3, 18.6)%	8% (4.1, 15.0)%	7% (3.4, 13.7)%	1% (0.2, 5.4)%
place_teabox_skillet_mix	0% (0.0, 3.7)%	0% (0.0, 3.7)%	3% (1.0, 8.5)%	0% (0.0, 3.7)%
put_toycar_left_basket_mix	2% (0.6, 7.0)%	4% (1.6, 9.8)%	2% (0.6, 7.0)%	0% (0.0, 3.7)%
roller_to_plate_mix	5% (2.2, 11.2)%	**50%** (40.4, 59.6)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%

Table 24: Per-task policy success (%), part 2 of 3, using the 100-trial protocol and 95% Wilson intervals of Table 23. Bold marks taskwise maxima across all twelve policies, including ties.

Scene	π_0↑	AHA-WAM↑	X-VLA↑	WLA↑	G0.5↑
grab_dumbbell	52% (42.3, 61.5)%	80% (71.1, 86.7)%	81% (72.2, 87.5)%	47% (37.5, 56.7)%	90% (82.6, 94.5)%
move_can_coaster	0% (0.0, 3.7)%	33% (24.6, 42.7)%	18% (11.7, 26.7)%	6% (2.8, 12.5)%	45% (35.6, 54.8)%
move_cup_pad	40% (30.9, 49.8)%	76% (66.8, 83.3)%	72% (62.5, 79.9)%	77% (67.8, 84.2)%	66% (56.3, 74.5)%
place_teabox_scale	6% (2.8, 12.5)%	52% (42.3, 61.5)%	0% (0.0, 3.7)%	41% (31.9, 50.8)%	67% (57.3, 75.4)%
grasp_mallet_novel	94% (87.5, 97.2)%	93% (86.3, 96.6)%	93% (86.3, 96.6)%	95% (88.8, 97.8)%	79% (70.0, 85.8)%
lift_trophy	44% (34.7, 53.8)%	29% (21.0, 38.5)%	25% (17.5, 34.3)%	76% (66.8, 83.3)%	30% (21.9, 39.6)%
place_seal_coaster	1% (0.2, 5.4)%	49% (39.4, 58.7)%	11% (6.3, 18.6)%	16% (10.1, 24.4)%	47% (37.5, 56.7)%
place_teabox_skillet	0% (0.0, 3.7)%	0% (0.0, 3.7)%	2% (0.6, 7.0)%	0% (0.0, 3.7)%	26% (18.4, 35.4)%
place_teabox_stand	0% (0.0, 3.7)%	47% (37.5, 56.7)%	2% (0.6, 7.0)%	22% (15.0, 31.1)%	67% (57.3, 75.4)%
place_toycar_plasticbox	8% (4.1, 15.0)%	42% (32.8, 51.8)%	19% (12.5, 27.8)%	31% (22.8, 40.6)%	60% (50.2, 69.1)%
place_a2b_left_novel	5% (2.2, 11.2)%	61% (51.2, 70.0)%	19% (12.5, 27.8)%	46% (36.6, 55.7)%	67% (57.3, 75.4)%
place_a2b_right_novel	5% (2.2, 11.2)%	52% (42.3, 61.5)%	20% (13.3, 28.9)%	47% (37.5, 56.7)%	60% (50.2, 69.1)%
card_right_of_scanner	2% (0.6, 7.0)%	1% (0.2, 5.4)%	10% (5.5, 17.4)%	5% (2.2, 11.2)%	**56%** (46.2, 65.3)%
phone_right_of_laptop	21% (14.2, 30.0)%	16% (10.1, 24.4)%	8% (4.1, 15.0)%	0% (0.0, 3.7)%	**47%** (37.5, 56.7)%
place_can_right_of_plasticbox	3% (1.0, 8.5)%	28% (20.1, 37.5)%	16% (10.1, 24.4)%	32% (23.7, 41.7)%	78% (68.9, 85.0)%
place_cube_right_of_plate	0% (0.0, 3.7)%	25% (17.5, 34.3)%	11% (6.3, 18.6)%	14% (8.5, 22.1)%	40% (30.9, 49.8)%
place_teabox_front_of_skillet	3% (1.0, 8.5)%	3% (1.0, 8.5)%	11% (6.3, 18.6)%	4% (1.6, 9.8)%	2% (0.6, 7.0)%
put_toycar_left_basket	0% (0.0, 3.7)%	1% (0.2, 5.4)%	0% (0.0, 3.7)%	1% (0.2, 5.4)%	28% (20.1, 37.5)%
place_between_bowls	0% (0.0, 3.7)%	17% (10.9, 25.5)%	11% (6.3, 18.6)%	39% (30.0, 48.8)%	19% (12.5, 27.8)%
attr_bind_bottle	26% (18.4, 35.4)%	44% (34.7, 53.8)%	39% (30.0, 48.8)%	28% (20.1, 37.5)%	**74%** (64.6, 81.6)%
attr_category_press_bell	52% (42.3, 61.5)%	**100%** (96.3, 100.0)%	94% (87.5, 97.2)%	67% (57.3, 75.4)%	77% (67.8, 84.2)%
attr_category_can_plasticbox	0% (0.0, 3.7)%	23% (15.8, 32.2)%	0% (0.0, 3.7)%	2% (0.6, 7.0)%	**44%** (34.7, 53.8)%
attr_color_beat_block	6% (2.8, 12.5)%	85% (76.7, 90.7)%	5% (2.2, 11.2)%	22% (15.0, 31.1)%	**91%** (83.8, 95.2)%
attr_color_block_3way	50% (40.4, 59.6)%	74% (64.6, 81.6)%	58% (48.2, 67.2)%	59% (49.2, 68.1)%	75% (65.7, 82.5)%
attr_size_beat_smaller	28% (20.1, 37.5)%	71% (61.5, 79.0)%	11% (6.3, 18.6)%	68% (58.3, 76.3)%	84% (75.6, 89.9)%
attr_size_block_pick	5% (2.2, 11.2)%	95% (88.8, 97.8)%	55% (45.2, 64.4)%	90% (82.6, 94.5)%	99% (94.6, 99.8)%
attr_bind_soap_box	13% (7.8, 21.0)%	54% (44.3, 63.4)%	34% (25.5, 43.7)%	37% (28.2, 46.8)%	63% (53.2, 71.8)%
attr_size_rank_middle	3% (1.0, 8.5)%	6% (2.8, 12.5)%	9% (4.8, 16.2)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
ref_block_on_plate	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%

Continued on next page.

Table 24 continued: task success (95% Wilson interval).

Scene	π_0↑	AHA-WAM↑	X-VLA↑	WLA↑	G0.5↑
ref_block_behind_can	**28%** (20.1, 37.5)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	3% (1.0, 8.5)%	21% (14.2, 30.0)%
ref_cup_on_tray	0% (0.0, 3.7)%	3% (1.0, 8.5)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
ref_leftmost_of_four	13% (7.8, 21.0)%	28% (20.1, 37.5)%	4% (1.6, 9.8)%	32% (23.7, 41.7)%	**70%** (60.4, 78.1)%
ref_mug_next_to_bowl	12% (7.0, 19.8)%	27% (19.3, 36.4)%	40% (30.9, 49.8)%	45% (35.6, 54.8)%	5% (2.2, 11.2)%
ref_not_the_red_one	7% (3.4, 13.7)%	14% (8.5, 22.1)%	2% (0.6, 7.0)%	15% (9.3, 23.3)%	7% (3.4, 13.7)%
ref_second_from_left	3% (1.0, 8.5)%	17% (10.9, 25.5)%	16% (10.1, 24.4)%	2% (0.6, 7.0)%	2% (0.6, 7.0)%
adjust_bottle_dustbin	94% (87.5, 97.2)%	6% (2.8, 12.5)%	99% (94.6, 99.8)%	**100%** (96.3, 100.0)%	84% (75.6, 89.9)%
click_bell_press_stapler	0% (0.0, 3.7)%	7% (3.4, 13.7)%	**95%** (88.8, 97.8)%	0% (0.0, 3.7)%	12% (7.0, 19.8)%
serve_teabox_then_ring	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	**1%** (0.2, 5.4)%
stapler_pad_press	15% (9.3, 23.3)%	**100%** (96.3, 100.0)%	73% (63.6, 80.7)%	83% (74.5, 89.1)%	95% (88.8, 97.8)%
lift_pot_to_pad	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
pack_toycar_then_ring	0% (0.0, 3.7)%	6% (2.8, 12.5)%	8% (4.1, 15.0)%	0% (0.0, 3.7)%	17% (10.9, 25.5)%
move_cup_pad_mix	1% (0.2, 5.4)%	35% (26.4, 44.7)%	0% (0.0, 3.7)%	30% (21.9, 39.6)%	39% (30.0, 48.8)%
ref_block_on_plate_mix	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
attr_teabox_basket_then_ring	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
card_right_of_scanner_mix	0% (0.0, 3.7)%	0% (0.0, 3.7)%	5% (2.2, 11.2)%	0% (0.0, 3.7)%	**56%** (46.2, 65.3)%
place_teabox_skillet_mix	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	**10%** (5.5, 17.4)%
put_toycar_left_basket_mix	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	22% (15.0, 31.1)%
roller_to_plate_mix	0% (0.0, 3.7)%	1% (0.2, 5.4)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%	36% (27.3, 45.8)%

Table 25: Per-task policy success (%), part 3 of 3: X-WAM, Motus, and LingBot-VA. Each cell gives success and a 95% Wilson interval over 100 trials. Bold marks taskwise maxima across all twelve policies, including ties.

Scene	X-WAM↑	Motus↑	LingBot-VA↑
grab_dumbbell	**97%** (91.5, 99.0)%	62% (52.2, 70.9)%	86% (77.9, 91.5)%
move_can_coaster	**72%** (62.5, 79.9)%	14% (8.5, 22.1)%	14% (8.5, 22.1)%
move_cup_pad	92% (85.0, 95.9)%	**99%** (94.6, 99.8)%	93% (86.3, 96.6)%
place_teabox_scale	**83%** (74.5, 89.1)%	28% (20.1, 37.5)%	49% (39.4, 58.7)%
grasp_mallet_novel	**98%** (93.0, 99.4)%	**98%** (93.0, 99.4)%	97% (91.5, 99.0)%
lift_trophy	**87%** (79.0, 92.2)%	52% (42.3, 61.5)%	69% (59.4, 77.2)%

Continued on next page.

Table 25 continued: task success (95% Wilson interval).

Scene	X-WAM↑	Motus↑	LingBot-VA↑
place_seal_coaster	**60%** (50.2, 69.1)%	59% (49.2, 68.1)%	8% (4.1, 15.0)%
place_teabox_skillet	**29%** (21.0, 38.5)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
place_teabox_stand	**79%** (70.0, 85.8)%	51% (41.3, 60.6)%	75% (65.7, 82.5)%
place_toycar_plasticbox	**86%** (77.9, 91.5)%	48% (38.5, 57.7)%	67% (57.3, 75.4)%
place_a2b_left_novel	55% (45.2, 64.4)%	60% (50.2, 69.1)%	64% (54.2, 72.7)%
place_a2b_right_novel	56% (46.2, 65.3)%	64% (54.2, 72.7)%	49% (39.4, 58.7)%
card_right_of_scanner	5% (2.2, 11.2)%	1% (0.2, 5.4)%	7% (3.4, 13.7)%
phone_right_of_laptop	14% (8.5, 22.1)%	33% (24.6, 42.7)%	2% (0.6, 7.0)%
place_can_right_of_plasticbox	3% (1.0, 8.5)%	**87%** (79.0, 92.2)%	29% (21.0, 38.5)%
place_cube_right_of_plate	20% (13.3, 28.9)%	**62%** (52.2, 70.9)%	48% (38.5, 57.7)%
place_teabox_front_of_skillet	4% (1.6, 9.8)%	1% (0.2, 5.4)%	4% (1.6, 9.8)%
put_toycar_left_basket	21% (14.2, 30.0)%	**31%** (22.8, 40.6)%	6% (2.8, 12.5)%
place_between_bowls	34% (25.5, 43.7)%	17% (10.9, 25.5)%	**53%** (43.3, 62.5)%
attr_bind_bottle	62% (52.2, 70.9)%	69% (59.4, 77.2)%	66% (56.3, 74.5)%
attr_category_press_bell	86% (77.9, 91.5)%	**100%** (96.3, 100.0)%	78% (68.9, 85.0)%
attr_category_can_plasticbox	41% (31.9, 50.8)%	32% (23.7, 41.7)%	3% (1.0, 8.5)%
attr_color_beat_block	69% (59.4, 77.2)%	55% (45.2, 64.4)%	**91%** (83.8, 95.2)%
attr_color_block_3way	62% (52.2, 70.9)%	68% (58.3, 76.3)%	**99%** (94.6, 99.8)%
attr_size_beat_smaller	49% (39.4, 58.7)%	27% (19.3, 36.4)%	35% (26.4, 44.7)%
attr_size_block_pick	**100%** (96.3, 100.0)%	83% (74.5, 89.1)%	69% (59.4, 77.2)%
attr_bind_soap_box	71% (61.5, 79.0)%	72% (62.5, 79.9)%	**79%** (70.0, 85.8)%
attr_size_rank_middle	1% (0.2, 5.4)%	6% (2.8, 12.5)%	**28%** (20.1, 37.5)%
ref_block_on_plate	0% (0.0, 3.7)%	0% (0.0, 3.7)%	6% (2.8, 12.5)%
ref_block_behind_can	18% (11.7, 26.7)%	11% (6.3, 18.6)%	7% (3.4, 13.7)%
ref_cup_on_tray	0% (0.0, 3.7)%	1% (0.2, 5.4)%	7% (3.4, 13.7)%
ref_leftmost_of_four	50% (40.4, 59.6)%	37% (28.2, 46.8)%	25% (17.5, 34.3)%
ref_mug_next_to_bowl	46% (36.6, 55.7)%	18% (11.7, 26.7)%	**70%** (60.4, 78.1)%
ref_not_the_red_one	18% (11.7, 26.7)%	**25%** (17.5, 34.3)%	0% (0.0, 3.7)%
ref_second_from_left	10% (5.5, 17.4)%	2% (0.6, 7.0)%	**62%** (52.2, 70.9)%
adjust_bottle_dustbin	72% (62.5, 79.9)%	85% (76.7, 90.7)%	79% (70.0, 85.8)%
click_bell_press_stapler	1% (0.2, 5.4)%	59% (49.2, 68.1)%	4% (1.6, 9.8)%
serve_teabox_then_ring	**1%** (0.2, 5.4)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
stapler_pad_press	95% (88.8, 97.8)%	92% (85.0, 95.9)%	**100%** (96.3, 100.0)%

Continued on next page.

Table 25 continued: task success (95% Wilson interval).

Scene	X-WAM↑	Motus↑	LingBot-VA↑
lift_pot_to_pad	2% (0.6, 7.0)%	9% (4.8, 16.2)%	**34%** (25.5, 43.7)%
pack_toycar_then_ring	**35%** (26.4, 44.7)%	3% (1.0, 8.5)%	21% (14.2, 30.0)%
move_cup_pad_mix	9% (4.8, 16.2)%	20% (13.3, 28.9)%	73% (63.6, 80.7)%
ref_block_on_plate_mix	0% (0.0, 3.7)%	0% (0.0, 3.7)%	11% (6.3, 18.6)%
attr_teabox_basket_then_ring	1% (0.2, 5.4)%	1% (0.2, 5.4)%	0% (0.0, 3.7)%
card_right_of_scanner_mix	2% (0.6, 7.0)%	0% (0.0, 3.7)%	6% (2.8, 12.5)%
place_teabox_skillet_mix	0% (0.0, 3.7)%	0% (0.0, 3.7)%	0% (0.0, 3.7)%
put_toycar_left_basket_mix	18% (11.7, 26.7)%	**28%** (20.1, 37.5)%	5% (2.2, 11.2)%
roller_to_plate_mix	20% (13.3, 28.9)%	6% (2.8, 12.5)%	38% (29.1, 47.8)%

F.2 Agent Task Results

GPT-6 succeeds in 185/240 episodes, with family means of 84.00, 77.78, 71.11, 97.14, 56.67, and 71.43%, yielding Overall 76.35%.

Gemini succeeds in 58/240 episodes, with family means of 18.00, 24.44, 22.22, 45.71, 10.00, and 25.71%, yielding Overall 24.35%. Table 26 gives both agents' per-task results.

Table 26: Agent task success (%) on five frozen seeds per scene, with 95% Wilson intervals. Bold marks the higher agent score per task, including ties. On two tasks, agents receive full compound instructions while policies use staged delivery (Appendix B.5).

Scene	GPT-6↑	Gemini 3.8 Flash↑
grab_dumbbell	**100%** (56.6, 100.0)%	20% (3.6, 62.4)%
move_can_coaster	**60%** (23.1, 88.2)%	20% (3.6, 62.4)%
move_cup_pad	**80%** (37.6, 96.4)%	0% (0.0, 43.4)%
place_teabox_scale	**80%** (37.6, 96.4)%	0% (0.0, 43.4)%
grasp_mallet_novel	**100%** (56.6, 100.0)%	20% (3.6, 62.4)%
lift_trophy	**100%** (56.6, 100.0)%	0% (0.0, 43.4)%
place_seal_coaster	**80%** (37.6, 96.4)%	20% (3.6, 62.4)%
place_teabox_skillet	**80%** (37.6, 96.4)%	60% (23.1, 88.2)%
place_teabox_stand	**80%** (37.6, 96.4)%	0% (0.0, 43.4)%
place_toycar_plasticbox	**80%** (37.6, 96.4)%	40% (11.8, 76.9)%
place_a2b_left_novel	**80%** (37.6, 96.4)%	60% (23.1, 88.2)%
place_a2b_right_novel	**100%** (56.6, 100.0)%	40% (11.8, 76.9)%
card_right_of_scanner	**40%** (11.8, 76.9)%	20% (3.6, 62.4)%
phone_right_of_laptop	**80%** (37.6, 96.4)%	0% (0.0, 43.4)%
place_can_right_of_plasticbox	**40%** (11.8, 76.9)%	0% (0.0, 43.4)%

Continued on next page.

Table 26 continued: task success (95% Wilson interval).

Scene	GPT-6↑	Gemini 3.8 Flash↑
place_cube_right_of_plate	**100%** (56.6, 100.0)%	80% (37.6, 96.4)%
place_teabox_front_of_skillet	**100%** (56.6, 100.0)%	20% (3.6, 62.4)%
put_toycar_left_basket	**60%** (23.1, 88.2)%	0% (0.0, 43.4)%
place_between_bowls	**100%** (56.6, 100.0)%	0% (0.0, 43.4)%
attr_bind_bottle	**20%** (3.6, 62.4)%	0% (0.0, 43.4)%
attr_category_press_bell	**100%** (56.6, 100.0)%	**100%** (56.6, 100.0)%
attr_category_can_plasticbox	**80%** (37.6, 96.4)%	0% (0.0, 43.4)%
attr_color_beat_block	**20%** (3.6, 62.4)%	0% (0.0, 43.4)%
attr_color_block_3way	**100%** (56.6, 100.0)%	20% (3.6, 62.4)%
attr_size_beat_smaller	**40%** (11.8, 76.9)%	0% (0.0, 43.4)%
attr_size_block_pick	**100%** (56.6, 100.0)%	40% (11.8, 76.9)%
attr_bind_soap_box	**80%** (37.6, 96.4)%	0% (0.0, 43.4)%
attr_size_rank_middle	**100%** (56.6, 100.0)%	40% (11.8, 76.9)%
ref_block_on_plate	**100%** (56.6, 100.0)%	40% (11.8, 76.9)%
ref_block_behind_can	**100%** (56.6, 100.0)%	0% (0.0, 43.4)%
ref_cup_on_tray	**100%** (56.6, 100.0)%	20% (3.6, 62.4)%
ref_leftmost_of_four	**80%** (37.6, 96.4)%	**80%** (37.6, 96.4)%
ref_mug_next_to_bowl	**100%** (56.6, 100.0)%	20% (3.6, 62.4)%
ref_not_the_red_one	**100%** (56.6, 100.0)%	60% (23.1, 88.2)%
ref_second_from_left	**100%** (56.6, 100.0)%	**100%** (56.6, 100.0)%
adjust_bottle_dustbin	**20%** (3.6, 62.4)%	0% (0.0, 43.4)%
click_bell_press_stapler	**80%** (37.6, 96.4)%	40% (11.8, 76.9)%
serve_teabox_then_ring	**60%** (23.1, 88.2)%	20% (3.6, 62.4)%
stapler_pad_press	**80%** (37.6, 96.4)%	0% (0.0, 43.4)%
lift_pot_to_pad	**0%** (0.0, 43.4)%	**0%** (0.0, 43.4)%
pack_toycar_then_ring	**100%** (56.6, 100.0)%	0% (0.0, 43.4)%
move_cup_pad_mix	**80%** (37.6, 96.4)%	0% (0.0, 43.4)%
ref_block_on_plate_mix	**100%** (56.6, 100.0)%	80% (37.6, 96.4)%
attr_teabox_basket_then_ring	**20%** (3.6, 62.4)%	0% (0.0, 43.4)%
card_right_of_scanner_mix	**60%** (23.1, 88.2)%	40% (11.8, 76.9)%
place_teabox_skillet_mix	**100%** (56.6, 100.0)%	60% (23.1, 88.2)%
put_toycar_left_basket_mix	**40%** (11.8, 76.9)%	0% (0.0, 43.4)%
roller_to_plate_mix	**100%** (56.6, 100.0)%	0% (0.0, 43.4)%