TASK-DEFINITION CONFLICTS IN ROBOT BENCHMARKS: MEASUREMENT AND LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot benchmarks distribute task definitions across instructions, demonstrations, rewards, and success checks. These components can agree at runtime yet implement a goal different from the declaration. We introduce *executable task contracts* to trace this correspondence and separate measurement changes from learning interventions. The same reference guides *Contract-Guided Supervision Repair* (CGSR), which searches late-to-early rollback boundaries, preserves action prefixes, and verifies complete repaired executions. In LIBERO, native and calibrated completion checks differ by 38.0 percentage points on identical executions. In a frozen extension, CGSR repairs nine of nine fresh-execution failures versus six for full recollection, while halving candidate-execution cost per verified repair. A subsequent exploratory student comparison improves calibrated completion by 13.3 points under equal training updates and unequal acquired-data yields. Blinded human review and RoboCasa monitor diagnostics clarify the interpretation of these outcomes. Together, explicit task-reference tracing and verified repair connect benchmark claims to the behaviors actually measured and taught.

1 INTRODUCTION

Robot benchmarks turn verbal goals into executable tests of manipulation capability. In LIBERO and SIMPLER, that translation spans descriptions, policy-facing instructions, demonstrations, rewards, and success predicates (Liu et al., 2023; Li et al., 2024b). Yet agreement between training and evaluation does not ensure agreement with the declaration. A policy may reliably place a mug *beside* a caddy while the declaration requests placement *inside* its compartment. The score is internally consistent, but its apparent interpretation refers to a different capability.

Recent work examines benchmark validity, instruction sensitivity, and specification-based monitoring (Jiang et al., 2026; Fang et al., 2026; Huang et al., 2026; Peng et al., 2026). These perspectives leave an important correspondence question: how does a stated task propagate through the components that teach, execute, and score behavior? Moreover, correcting a discrepancy can change either the labels assigned to an existing execution or the behavior acquired by a learner. These interventions answer different questions. A new success criterion changes measurement; a new demonstration changes supervision and must be followed by a separate policy evaluation.

An executable reference for diagnosis and repair. We introduce *executable task contracts* that bind a chosen task reference to its runtime implementation. The audit follows instructions into actual policy calls and constructs source-grounded and same-execution witnesses of disagreement. It also makes monitor-update rules explicit, since repeated reads can alter a stateful success history. The framework provides a common task reference for the measurement and learning branches in Figure 1.

A failed demonstration can contain useful actions. Rather than discard them or repeat collection, *Contract-Guided Supervision Repair* (CGSR) searches backward through admissible phases, regenerates a suffix, and verifies the resulting execution. Late rollback favors locality; earlier rollback makes more failures recoverable. We evaluate yield, retention, and execution cost together, then separately test the acquired data through student learning.

Our experiments trace caddy conflicts through released behavior, a 38.0-point difference between completion criteria on identical executions, and repaired supervision. A frozen extension recovers nine of nine failures at half the per-repair candidate-execution cost of full recollection; a later three-

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

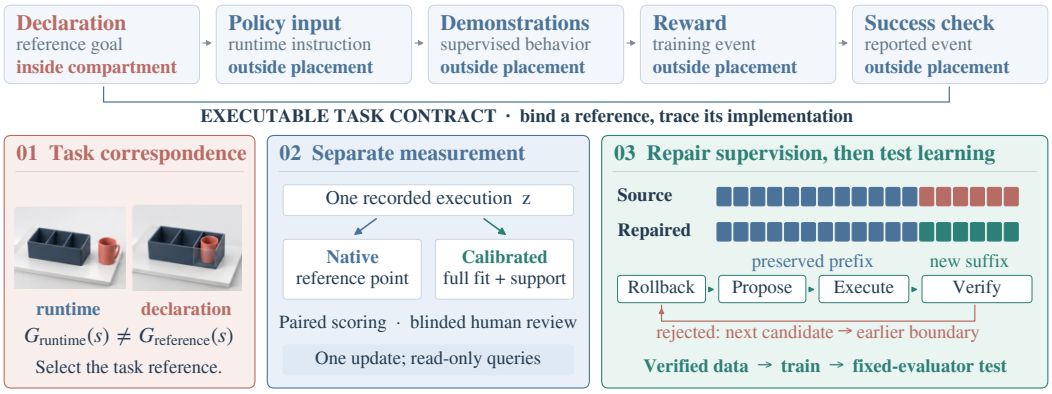

Figure 1: **Task correspondence connects diagnosis, measurement, and repair.** The caddy example illustrates different goals across benchmark components; dashed links denote correspondence checks. Rescoring fixes the execution, whereas CGSR changes supervision before a separate policy test. Placement artwork is conceptual; Figure 3 provides the experimental images.

seed student study tests the acquired-data pipelines. Blinded human review and RoboCasa, SIMPLER, and MINERVA analyses further examine how endpoint interpretation, monitor updates, and policy inputs affect benchmark conclusions.

Our contributions.

- **Executable task contracts** trace declarations, policy inputs, supervision, rewards, and success checks, exposing mismatches through source and execution witnesses.
- **Measurement–learning separation** distinguishes paired rescoring from supervision interventions, with geometric calibration, blinded human review, and query-invariance tests.
- **Contract-guided supervision repair** adapts rollback depth to balance recovery, locality, and cost; fixed-boundary controls and separate student evaluations test these outcomes.

2 Related Work

Benchmark validity and task-conditioned behavior. Robot-benchmark audits examine shortcuts, statistical significance, overfitting, and data-source dependence (Jiang et al., 2026). Instruction-focused evaluations instead test paraphrases, counterfactual goals, and minimal language changes (Kim et al., 2026; Fang et al., 2026; Liu et al., 2026d), while RoboRMBench studies reward-model sensitivity to paraphrasing (Jeung et al., 2026). We examine correspondence among a benchmark's own task representations, including the instruction reaching the policy. This locates implementation mismatches before interpreting policy behavior as capability.

Specifications and success monitoring. Symbolic task definitions and embodied interfaces make goals executable (Srivastava et al., 2021; Li et al., 2024a); related evaluations study spatial robustness, execution constraints, recovery, and outcome judgments (Wang et al., 2026; Mao et al., 2025; Liu et al., 2026b; Navasardyan et al., 2026). SafeManip uses temporal monitors (Huang et al., 2026), and MANIGUARD connects physics-grounded specifications to evaluation and fine-tuning (Peng et al., 2026). Our analysis links the event encoded by a checker to its task reference and caller-update semantics, while human review characterizes the operating characteristics of geometric endpoints.

Demonstration generation and repair. MimicGen transforms demonstrations across contexts, whereas CP-Gen preserves trajectory constraints as geometry and poses vary (Mandlekar et al., 2023; Lin et al., 2025). CGSR begins with a recorded failure under an explicit contract and searches for a verified replacement suffix. Its distinctive experimental question is the trade-off between preserving source actions and recovering additional failures under bounded execution cost. Comparisons with fixed rollback and recollection test this trade-off; policy learning is evaluated separately.

Table 1: **The controlled intervention determines the question answered.** Each comparison holds the indicated quantities fixed and returns evidence relative to its stated task reference.

Question	Fixed	Changed	Output
Correspondence	Scene and trajectory	Task reference	Conflicting goal labels
Measurement	Executed behavior	Completion check	Paired score gap
Monitor semantics	Physics / sensor epoch	Query schedule	Label and state history
Learning	Evaluator and recipe	Supervision	New-policy performance
Repair	Source pool and interface	Rollback / search rule	Yield, locality, cost

3 METHOD

3.1 BINDING A TASK REFERENCE TO ITS IMPLEMENTATION

A contract brings the task's representations into one auditable object. We write

$$\mathcal{C} = (\ell, \mathcal{W}, G, R, \tau), \qquad \ell = (\ell_{\text{decl}}, \ell_{\text{policy}}), \tag{1}$$

where $\mathcal{W}$ specifies objects, assets, fixtures, and regions; G and R define success and reward; and τ records labeling, stopping, and monitor updates. Supervision $\mathcal{D}$ and provenance bind versions, model weights, and observation/action alignment to the contract. Declaration and policy input are recorded separately because a task name alone does not identify the instruction reaching the policy.

From a discrepancy to an executable witness. The audit pins versions, parses task components, and selects candidates through deterministic cues. Manual inspection resolves region ownership, initialization, object aliases, and semantic equivalence; tracing the caller and rescoring shared trajectories then produces an executable witness. In caddy, either revising the declaration to outside placement or revising the runtime to compartment placement restores consistency. Our supervision experiments explicitly adopt the declaration as their governing reference.

3.2 SEPARATING MEASUREMENT FROM LEARNING

A change in reported success can arise without any change in behavior. Let z be an executed trajectory and $Y_G(z;\tau)$ its episode label. Rescoring the same collection $\mathcal{Z}$ isolates a measurement contrast,

$$\Delta_{\text{meas}} = \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} \big[Y_{G_1}(z;\tau) - Y_{G_0}(z;\tau) \big]. \tag{2}$$

A learning comparison instead trains $\pi_j = \text{Train}(\mathcal{D}_j)$ under matched within-stage recipes and evaluates both policies using the same $(G_\star, \tau_\star)$ on paired starts. Rescoring therefore relabels existing behavior, while replacing complete demonstrations changes what the policy learns (Table 1).

Completion is a temporal event. For $t^\star = \min\{t : G(s_t) = 1\}$, first-hit success is $Y_{\text{hit}} = \mathbf{1}[t^\star \le T]$. Persistence also requires $G(s_{t^\star+j}) = 1$ for every $j \in \{0, \dots, k\}$. Our caddy final-window endpoint requires success at all final 50 post-action states of a 600-action execution. These rules distinguish reaching a goal, maintaining it immediately after first hit, and completing it at the end of an episode.

The monitor's update schedule also defines the event. *Query invariance* separates updates $m_t = U(m_{t-1}, x_t)$ from read-only lookups. Cache keys distinguish episode, transition, boundary, and sensor epoch, and reset clears the cache. RoboCasa reward/Gym queries straddle fixture updates; our shadow monitor uses one post-fixture update and keeps their sensor epochs distinct.

3.3 CONTRACT-GUIDED SUPERVISION REPAIR

A failed placement may still contain a useful grasp and transport prefix. Recollecting the entire demonstration discards that behavior, whereas fixing one late rollback point can leave some failures unreachable. CGSR addresses this trade-off by searching from late to earlier admissible phases. Given original actions $a_{0:T-1}$, it constructs

$$\widetilde{a} = \big(a_{0:k-1},\, u_{k:T'-1}\big), \qquad V_{\mathcal{C}}(\text{Exec}(s_0, \widetilde{a})) = 1, \tag{3}$$

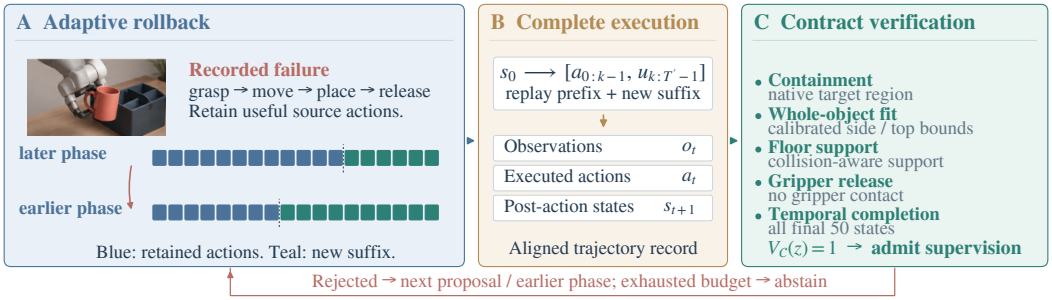

Figure 2: **Late-to-early repair balances locality and recovery coverage. A:** A local failure motivates searching successive rollback phases. **B:** Every candidate replays its retained prefix from the registered start. **C:** Admission checks the full final window and the aligned execution record. The artwork is conceptual; privileged geometry guides data generation only.

Algorithm 1 Contract-Guided Supervision Repair (CGSR)

Input: source $(s_0, a_{0:T-1})$, verifier V_C, bounded proposals, budget B
Output: verified repair with an aligned execution record, or abstention

Diagnose violated clauses; order admissible boundaries $\mathcal{B}$ from late to early
for $k \in \mathcal{B}$ **do**
  for suffix proposal u in the fixed candidate order at k **do**
    if remaining budget cannot cover the candidate: **return** abstention
    $z \leftarrow \mathrm{Exec}(s_0, [a_{0:k-1}, u])$; charge replay and candidate costs
    if $V_C(z) = 1$ and alignment holds: **return** repair and record
  end for
end for
return abstention

where k is a rollback boundary and V_C verifies the complete execution. Source actions are retained, while observations and states are regenerated together during execution.

Bounded search, complete verification. Clause diagnosis orders admissible phases. At each phase, a bounded controller proposes placement, approach, orientation, release, and retreat parameters; unsuccessful local candidates trigger earlier rollback. Every candidate executes from the registered initial state through the unchanged action interface. Caddy admission requires containment, calibrated whole-object fit, floor support, and no gripper contact throughout the final 50 post-action states. Algorithm 1 returns the first verified repair within the proposal family and budget, or abstains.

A joint evaluation of yield, locality, and cost. Verified yield counts eligible sources repaired; locality is the retained fraction k/T, and generated suffix length is $T' - k$. Candidate cost includes rejected attempts and replayed prefixes, divided by verified sources; common screening/setup costs are separate. Retention conditions on successful repair, so comparisons on common-success sources distinguish coverage from locality.

4 Experiments

4.1 Experimental setup

Benchmarks and policies. We parse 130 LIBERO task files and study two caddy tasks sharing one scene template. The released-behavior comparison uses 100 official demonstrations and QueST SFT/RIPT and Qwen2.5-OpenVLA policies (Mete et al., 2024; Tan et al., 2025; Ganai et al., 2026), each executed from 50 starts per task for 600 actions without early stopping. Three frozen adapted Qwen models supply 150 direct placement executions on 50 shared starts. RoboCasa365 provides monitor diagnostics (Nasiriany et al., 2026); SIMPLER and MINERVA provide instruction and temporal comparisons (Kwok et al., 2026; Sendai et al., 2026).

Controlled comparisons. Training arms share initialization, update budget, task sampling, and evaluator within each study. We report mean ± sample SD across training seeds, with shared starts

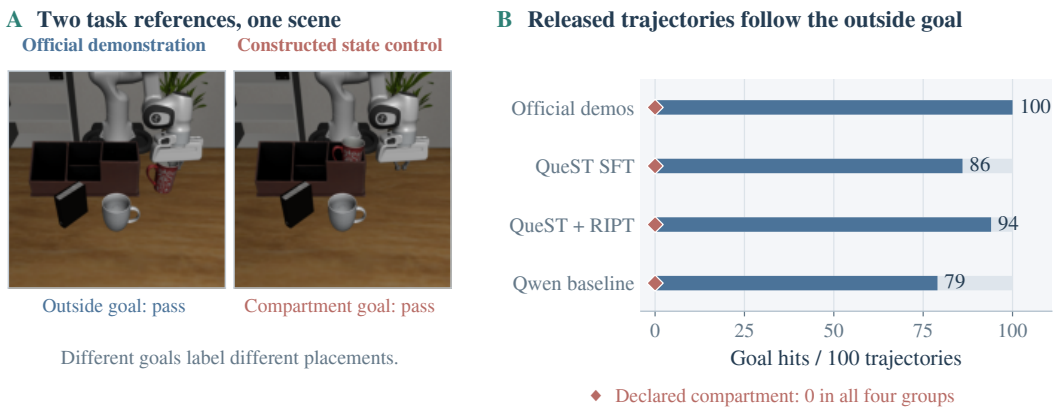

Figure 3: **Released behavior follows the outside-placement runtime task. A:** The official demonstration and an explicitly constructed state illustrate competing task references; the latter is a state control. **B:** Each group contains 100 trajectories scored against both goals. Outside-goal hits are shown in blue; every group has zero declaration-referenced compartment hits.

retained as paired identities. Paired label tests use McNemar comparisons; indicated paired-score intervals use Tango's method (Tango, 1998). Repair cohorts instead report fixed source counts and complete candidate costs. Appendices Q, A.2, and W detail the protocols and selection chronology.

4.2 Task correspondence: tracing declarations into behavior

A consistent runtime can implement a different declared task. The caddy declaration requests placement in the right compartment, but the factory derives policy text from a filename requesting placement to the right *of* the caddy. Reward checks `On(mug, table-region)`; the declaration-referenced goal is `In(mug, caddy-region)`. Both regions exist in the same scene. Tracing source ownership and the policy caller establishes the mismatch illustrated in Figure 3.

All 100 official demonstrations satisfy only the outside goal. The three released policies hit it on 86, 94, and 79 of 100 starts, respectively, with no compartment hit; Qwen's 400-action prefix yields identical counts. Thus the policies follow their actual instructions while the declaration names another goal. Among 14 cue-selected census candidates, ten have consistent encodings, two add an unstated stove-maintenance condition, and two are the caddy conflicts. The complete review, including six later templates with no additional confirmed conflict, appears in Appendices C and T.2.

4.3 Measurement: geometric endpoints and monitor semantics

The completion criterion changes the score on identical executions. Native `In` tests an object reference position. Our geometric endpoint instead checks the cup's side/top envelope, floor support, and release. Physical controls identified a 2.395-mm difference between the white cup's visual bottom and collision support surface; calibration uses the collision bottom at the floor while retaining the other requirements. Eight supported positive controls pass. On 150 directly recorded executions, native and calibrated final-50 success are $65.3 \pm 6.1\%$ and $27.3 \pm 5.0\%$, with 57 native-only passes. Both already exclude gripper contact; their 38.0-point difference concerns geometry and support.

Human review characterizes the two operating points. Two non-author reviewers independently label 105 frozen two-view videos, agreeing on 102 ($\kappa = 0.939$); a third blinded reviewer adjudicates disagreements. Native agrees with 95/105 final labels, and calibrated geometry with 97/105. More importantly, their errors differ: native accepts all 39 human successes and ten human failures, whereas geometry rejects all 66 human failures but misses eight successes (Figure 4). The latter is a conservative operational endpoint. This separate stratified review characterizes endpoint behavior; Appendix W.4 provides sampling and annotation details. Tolerance and clause analyses further show that removing the top-envelope constraint alone rescues none of the 57 direct-record disagreements, because side or support violations remain.

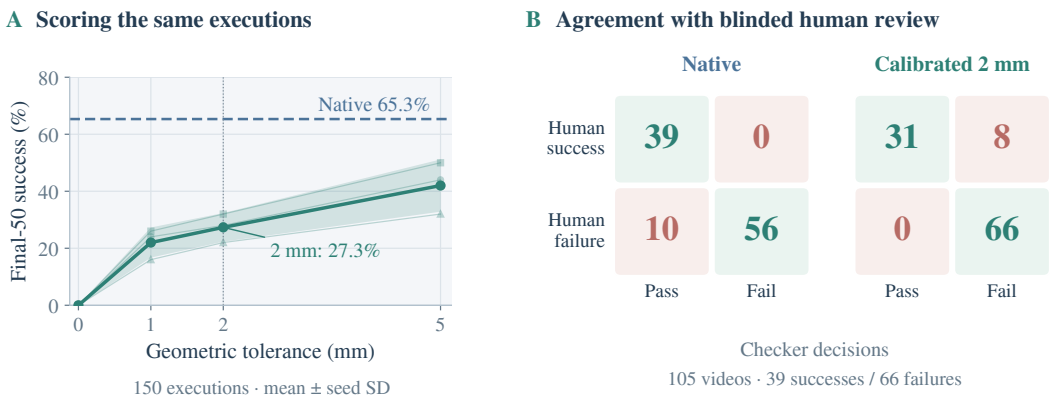

Figure 4: **Endpoint calibration reveals a trade-off in what counts as completion. A:** Tolerance values rescore the same 150 executions; the curve and shading give geometric mean ± seed SD, with individual seeds faintly marked. **B:** Confusion matrices compare each checker with 105 blinded human labels (rows: human outcomes; columns: checker decisions). Diagonal cells mark agreement.



Figure 5: **Native-successful executions can have different geometric outcomes.** These recorded final-Qwen examples show calibrated passes and failures at step 600; all satisfy native contact-free final-50 success. The fixed crops preserve the rendered scenes. Cases are selected post hoc within color/endpoint groups; Appendix M details selection and trace checks.

What the geometry adds. Figure 5 contrasts native-successful executions that pass or fail calibrated geometry. Protrusion can leave a reference point inside the target region while violating whole-object bounds. Conversely, the passing white cup is not upright: the criterion tests enclosure and support, not a prescribed orientation. The paired differences thus reflect visible placement geometry.

An extra read can count progress that never occurred. RoboCasa source-method tests identify query-sensitive labels in nine of twelve directly incrementing counters. Native OvenBroilFish and CoolKettle diagnostics then compare shadow monitors sharing post-update physics. Four reads trigger OvenBroilFish success before a door-opening event, after which its native latch preserves the label (Table 2). CoolKettle's externally advanced clock remains invariant. Caching matches one-read state and label histories at all 333 transitions, with an explicit monitor-update boundary.

4.4 CGSR: RECOVERY, LOCALITY, AND COMPONENT ABLATIONS

Recovery at a lower candidate-execution cost. After development on nine geometry failures, a separately frozen exploratory extension selects 16 previously unattempted sources from recovered phase logs. Seven pass fresh raw-action replay; the remaining nine form a common repair-only cohort. Methods share starts, interfaces, per-boundary proposals, and budget ceilings. CGSR repairs 9/9 failures, compared with 8/9 for fixed pre-place and 6/9 for fixed transport and full recollection (Figure 6). Its 3,964 candidate-execution steps correspond to 440.4 per verified source, versus 881.8

Table 2: **Query scheduling changes the first-success transition.** Native denotes the published Gym path. The other columns share post-update physics; Cache reproduces the specified one-read history.

Task	Controlled episode	Native	Post-update monitor			
			1 read	2 reads	4 reads	Cache
OvenBroilFish	Door stays closed	6	11	6	3	11
OvenBroilFish	Open/close at 4/5	10	15	10	3	15
CoolKettle	Burner initially off	151	151	151	151	151
CoolKettle	Burner on at 1–3	154	154	154	154	154

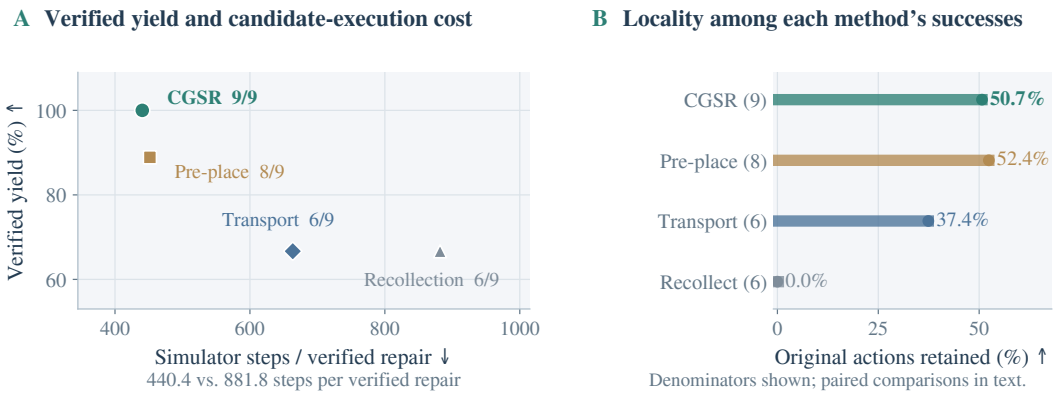

Figure 6: **Adaptive rollback combines repair coverage with action-prefix locality.** The cohort contains nine fresh-execution failures. **A:** Candidate cost includes rejected attempts and prefix replay. **B:** Retention means condition on each method's verified sources, with counts in parentheses. The common-success comparisons in the text separate locality from coverage.

for recollection: a 50.1% reduction. Screening and setup add common overheads of 5,212 and 1,268 steps, reported separately in Appendix A.2.

Adaptivity resolves the fixed-boundary trade-off. CGSR retains 50.7% of original actions and generates a mean 168.6-step suffix. On eight common successes, fixed pre-place preserves exactly the same prefix, but it cannot recover the ninth source. Against fixed transport on six common successes, CGSR retains 15.6 points more source actions and generates 68.8 fewer suffix steps. Adaptivity therefore improves coverage relative to a late boundary and locality relative to an earlier one. This is the design benefit tested by the fixed-boundary comparisons.

In the separate development cohort, removing classification or randomizing source order yields exact ties: all sources share one failure class, and the budget admits every repair bundle. Complete controls and selection chronology are retained in Appendices A.1 and W.5.

4.5 Learning: goal adaptation, acquisition yield, and equal-size controls

A corrected reward needs behavioral support. Replacing the caddy reward with the compartment predicate yields 0/240 successful RIPT rollouts and no parameter changes under the tested one-seed budget; aligning policy text gives the same outcome. With all-zero rewards, leave-one-out advantages vanish. In contrast, 18 action-generated compartment demonstrations enable QueST to reach $60.8 \pm 8.8\%$ native compartment first-hit success after 600 updates, versus zero for source-matched original-data controls. Outside success simultaneously changes from 73.3% to 0.8% (Figure 7A). Both arms receive outside text, so the intervention establishes fixed-goal adaptation to demonstrations. The separate Qwen protocol in Appendix E.3 reaches $70.0 \pm 2.0\%$ native first-hit success with 95 demonstrations and 4,200 updates.



Figure 7: **Supervision changes the attained goal and exploratory pipeline completion. A:** QueST native first-hit means over three seeds and 40 shared starts. **B–C:** The separate acquisition-pipeline study uses 600 updates and 50 reused starts per seed. Paired markers show all reported seeds for recollection (51 demonstrations) and CGSR (58); Table 3 summarizes both endpoints.

Table 3: **The acquired-data CGSR pipeline yields more verified supervision and higher exploratory student completion.** Acquisition uses the same 16 sources; candidate steps exclude common screening/setup. Student success is mean ± sample SD (%) over seeds 0, 3, and 4 on 50 reused starts each. Strict denotes calibrated 2-mm final-50 completion; Native includes no gripper contact. Dataset sizes are 51 and 58.

Method	Same-source acquisition		Student final-50 success	
	Verified ↑	Steps ↓	Strict ↑	Native ↑
Full recollection	9/16	10,510	13.3 ± 1.2	24.0 ± 4.0
CGSR	**16/16**	**6,468**	**26.7** ± 1.2	**47.3** ± 5.8

Acquisition yield under a common source stream. From the extension's 16 sources, CGSR verifies 16 demonstrations and recollection nine, using 6,468 and 10,510 candidate steps. Adding each entire verified yield to a common 42-source base produces CGSR58 and full51. With 600 updates per arm, calibrated final-50 success rises from $13.3 \pm 1.2\%$ to $26.7 \pm 1.2\%$ (Table 3); every reported seed favors CGSR on both endpoints (Figure 7B–C). This exploratory comparison uses previously examined starts and retains dataset size as an acquisition outcome. It therefore evaluates the complete pipeline, with per-seed counts in Appendix A.3, rather than an equal-size supervision intervention.

Learning effects at equal demonstration counts. A separate Qwen2.5-OpenVLA study fixes the task, training recipe, and 600 updates while comparing equal-size data arms (Table 4). Only 42/95 native-selected teachers meet calibrated checks, yet filtering gives no mean native gain at 42 demonstrations. At 52, both paired arms retain the 42 qualified sources; replacing ten failed trajectories with historical verified teachers lowers native completion by 9.3 points. Calibrated replacement effects instead average $+2.0$ points, with seed differences $[-4, +4, +6]$ (Appendix W.8). These replacements are not CGSR outputs, and they change complete trajectories rather than isolating observation, action, or length effects. Together, the studies motivate evaluating acquisition yield, demonstration validity, and downstream learning separately.

4.6 Analysis beyond the caddy case

Temporal completion and instruction alignment. Across 40 MINERVA-on-LIBERO tasks and 50 starts per task, first-hit success is 94.8%, whereas maintaining the native predicate for ten additional same-policy steps gives 84.4%. The 10.4-point reduction has task-cluster interval $[6.2, 15.7]$ and appears across all four suites (Figure 8A). In SIMPLER zucchini, revising the entire policy-facing instruction from a cloth reference to a plate reference raises base-policy strict completion from 11/40 to 26/40, a 37.5-point paired gain (Figure 8B). Because modifiers also change, this is a whole-

Table 4: **Equal-size supervision changes give no uniform native gain.** Qwen2.5-OpenVLA uses 600 updates per arm. Native contact-free final-50 success is mean ± sample SD (%; ↑) over seeds 0–2 on 50 shared starts per seed. Bold marks the best mean within each pair. Paired-repaired uses historical verified teachers, not CGSR outputs.

Comparison	Training data	Demonstrations	Native ↑ (%)
Filtering	Random-matched	42	**35.3** ± 9.5
	Filtered	42	**35.3** ± 5.0
Replacement	Mixed-paired	52	**39.3** ± 6.1
	Paired-repaired	52	30.0 ± 0.0

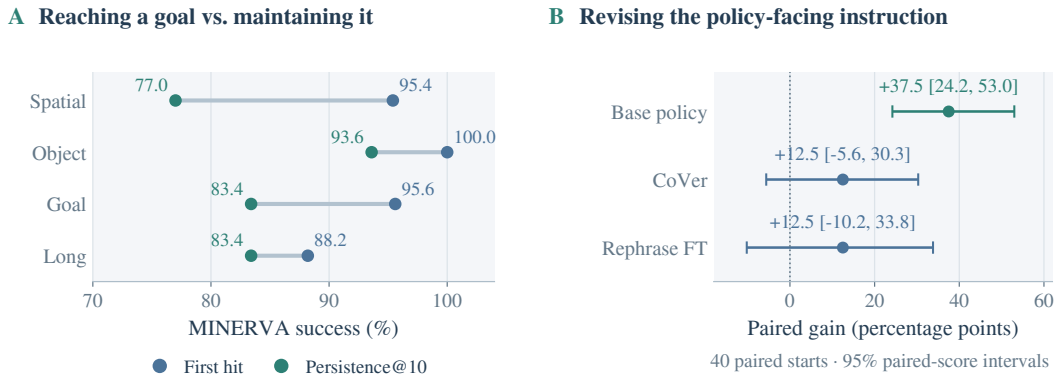

Figure 8: **The counted event and policy input matter beyond caddy. A:** Each MINERVA suite contains 500 executions; persistence@10 adds 0.50 s of the same policy after first hit. **B:** Whole-instruction revision uses 40 paired starts per policy. Bars denote 95% Tango paired-score intervals; only the base-policy interval excludes zero.

instruction intervention. Stricter scoring still preserves CoVer's aggregate advantage across the tested seven tasks; Appendix X reports the full criteria and task-level results.

Practical implications and scope. The appropriate remedy depends on the diagnosed mismatch: align task references, separate monitor updates from reads, calibrate completion criteria, or repair supervision before testing a new policy. Our repair evidence concerns small geometry-failure cohorts with privileged teachers and two caddy tasks sharing a scene template. Human judgments and geometric checks serve as operational references, while query tests establish specific runtime mechanisms. These boundaries identify the next validation targets: additional task templates for repair and independent cohorts for downstream learning. The evidence distinguishes three practical decisions—which task to implement, which event to count, and which supervision to provide—so corrections can be evaluated against their intended outcomes.

5 Conclusion

Executable task contracts link the task a benchmark describes to the event it measures and the behavior it teaches. Tracing these connections reveals declaration/runtime conflicts, substantial endpoint differences on identical executions, and query-dependent monitor histories. CGSR extends the same reference to verified supervision repair, balancing recovery coverage with retained actions and execution cost; its exploratory student comparison separately measures the acquired data's learning value. By keeping these interventions distinct, the framework makes benchmark corrections actionable while preserving a clear interpretation of what each improvement establishes.

Reproducibility Statement

We provide implementation details, experimental protocols, evaluation settings, and additional results in the appendices and supplementary materials. Code and supporting materials will be released to facilitate reproduction of the reported results.

AI Use Statement

Generative AI tools were used to assist with literature search, coding, and language polishing. The authors reviewed and verified the resulting content and take full responsibility for the final manuscript.

References

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael 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, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. URL https://arxiv.org/abs/2212.06817.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. URL https://arxiv.org/abs/2307.15818.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. URL https://arxiv.org/abs/2303.04137.

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. URL https://arxiv.org/abs/2602.17659.

Milan Ganai, Katie Luo, Jonas Frey, Clark Barrett, and Marco Pavone. Self-supervised bootstrapping of action-predictive embodied reasoning. In *RSS*, 2026.

Chengyue Huang, Khang Vo Huynh, Sebastian Elbaum, Zsolt Kira, and Lu Feng. Safemanip: A property-driven benchmark for temporal safety evaluation in robotic manipulation. *arXiv preprint arXiv:2605.12386*, 2026. URL https://arxiv.org/abs/2605.12386.

Wonje Jeung, Sangyeon Yoon, Hyesoo Hong, Yoonjun Cho, Dongjae Jeon, Bumjun Kim, Jean Oh, Youngjae Yu, and Albert No. Same trajectory, contradictory rewards (robormbench): Paraphrase

fragility in vision language reward models. *arXiv preprint arXiv:2609.05401*, 2026. URL https://arxiv.org/abs/2609.05401.

Tianchong Jiang, Xiangshan Tan, Samuel Wheeler, Luzhe Sun, Tewodros W. Ayalew, and Matthew Walter. What are we actually benchmarking in robot manipulation? *arXiv preprint arXiv:2606.04233*, 2026. URL https://arxiv.org/abs/2606.04233.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. HG-DAgger: Interactive imitation learning with human experts. *arXiv preprint arXiv:1810.02890*, 2018. URL https://arxiv.org/abs/1810.02890.

Chanyoung Kim, Minwoo Kim, Minseok Kang, Hyunwoo Kim, and Dahuin Jung. Libero-para: A diagnostic benchmark and metrics for paraphrase robustness in vla models. In *EMNLP*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Jacky Kwok, Xilun Zhang, Mengdi Xu, Yuejiang Liu, Azalia Mirhoseini, Chelsea Finn, and Marco Pavone. Scaling verification can be more effective than scaling policy learning for vision-language-action alignment. *arXiv preprint arXiv:2602.12281*, 2026. URL https://arxiv.org/abs/2602.12281.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. DART: Noise injection for robust imitation learning. In *CoRL*, 2017.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024. URL https://arxiv.org/abs/2403.03181.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, Weiyu Liu, Percy Liang, Li Fei-Fei, Jiayuan Mao, and Jiajun Wu. Embodied agent interface: Benchmarking LLMs for embodied decision making. In *NeurIPS*, 2024a.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024b. URL https://arxiv.org/abs/2405.05941.

Kevin Lin, Varun Ragunath, Andrew McAlinden, Aaditya Prasad, Jimmy Wu, Yuke Zhu, and Jeannette Bohg. Constraint-preserving data generation for visuomotor policy learning. In *CoRL*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, 2023.

Chaoqi Liu, Xiaoshen Han, Jiawei Gao, Yue Zhao, Haonan Chen, and Yilun Du. OAT: Ordered action tokenization. In *RSS*, 2026a.

Lin Liu, Zhicheng Bao, Lu Zhang, Ziying Song, Wu Yang, Shuai Tao, Wulong Liu, and Huchuan Lu. LIBERO-RECOVER: Beyond task success towards failure recovery in robotic manipulation models. *arXiv preprint arXiv:2609.05178*, 2026b. URL https://arxiv.org/abs/2609.05178.

Yicheng Liu, Shiduo Zhang, Zibin Dong, Baijun Ye, Tianyuan Yuan, Xiaopeng Yu, Linqi Yin, Chenhao Lu, Junhao Shi, Luca Jiang-Tao Yu, Liangtao Zheng, Tao Jiang, Jingjing Gong, Xipeng Qiu, and Hang Zhao. FASTer: Toward efficient autoregressive vision-language-action modeling via neural action tokenization. In *ICLR*, 2026c.

Yiwei Liu, Luwei Yang, and Shunbo Lei. One word, different action: A real-robot benchmark for language-conditioned embodied reasoning. *arXiv preprint arXiv:2609.05260*, 2026d. URL https://arxiv.org/abs/2609.05260.

Jianlan Luo, Perry Dong, Yuexiang Zhai, Yi Ma, and Sergey Levine. RLIF: Interactive imitation learning as reinforcement learning. In *ICLR*, 2024.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *CoRL*, 2023.

Yanbo Mao, Jianlong Fu, Ruoxuan Zhang, Hongxia Xie, and Meibao Yao. Beyond success: Refining elegant robot manipulation from mixed-quality data via just-in-time intervention. *arXiv preprint arXiv:2511.22555*, 2025. URL https://arxiv.org/abs/2511.22555.

Atharva Mete, Haotian Xue, Albert Wilcox, Yongxin Chen, and Animesh Garg. QueST: Self-supervised skill abstractions for learning continuous control. *arXiv preprint arXiv:2407.15840*, 2024. URL https://arxiv.org/abs/2407.15840.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *ICLR*, 2026.

Zaruhi Navasardyan, Tatul Danielyan, and Hrant Davtyan. Failbench: How reliable are VLMs at judging robot task success? *arXiv preprint arXiv:2609.03611*, 2026. URL https://arxiv.org/abs/2609.03611.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. URL https://arxiv.org/abs/2405.12213.

Yiyan Peng, Philip Wang, Simon Sinong Zhan, Yiqi Lyu, Zhenyang Ni, Jixin Yan, Fiorelli Wong, Ruochen Jiao, Hang Yin, Xinyu Cao, Huajie Shao, Manling Li, Ruohan Zhang, and Qi Zhu. MANIGUARD: A benchmark and data suite for specification-grounded safety evaluation and improvement of robotic manipulation. *arXiv preprint arXiv:2608.17386*, 2026. URL https://arxiv.org/abs/2608.17386.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. URL https://arxiv.org/abs/2501.09747.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011.

Kohei Sendai, Tatsuya Matsushima, and Yusuke Iwasawa. MINERVA: How small can a manipulation policy be and still solve LIBERO? *arXiv preprint arXiv:2609.03715*, 2026. URL https://arxiv.org/abs/2609.03715.

Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *arXiv preprint arXiv:2206.11251*, 2022. URL https://arxiv.org/abs/2206.11251.

Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, C. Karen Liu, Silvio Savarese, Hyowon Gweon, Jiajun Wu, and Li Fei-Fei. BEHAVIOR: Benchmark for everyday household activities in virtual, interactive, and ecological environments. *arXiv preprint arXiv:2108.03332*, 2021. URL https://arxiv.org/abs/2108.03332.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025. URL https://arxiv.org/abs/2505.17016.

Toshiro Tango. Equivalence test and confidence interval for the difference in proportions for the paired-sample design. *Statistics in Medicine*, 17(8):891–908, 1998. doi: 10.1002/(SICI)1097-0258(19980430)17:8<891::AID-SIM780>3.0.CO;2-B.

Guodong Wang, Chenkai Zhang, Qingjie Liu, Jinjin Zhang, Jiancheng Cai, Junjie Liu, and Xinmin Liu. LIBERO-X: Robustness litmus for vision-language-action models. *arXiv preprint arXiv:2602.06556*, 2026. URL https://arxiv.org/abs/2602.06556.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. URL https://arxiv.org/abs/2304.13705.

A Extended Repair and Learning Evidence

Reading guide. The appendix develops the evidence behind the paper's three interventions: task-reference auditing, same-execution measurement, and supervision repair. This section collects repair locality and the subsequent student comparison. Task definitions and policy interfaces are detailed in Appendices C, D, and Q; direct geometry and monitor tests in Appendix S; and blinded human review and complete learning arms in Appendix W.

A.1 Repair locality and complete execution cost

The locality analysis asks how much source behavior survives repair and what execution cost is required to verify it. It reuses the original checker, source pool, and expansion rule. Elementwise comparison verifies retained action prefixes, while artifact and alignment checks verify the associated observations, states, final-50 clauses, and attempt ledgers. Retention is k/T relative to the original length; generated suffix length $T' - k$ is reported separately from the number of replaced actions. Conditional means are accompanied by their verified-source coverage.

Unlike interactive imitation learning, this repair setting starts from a fixed recorded source. DAgger queries expert labels on learner-induced states (Ross et al., 2011), HG-DAgger lets a human control interventions (Kelly et al., 2018), DART perturbs demonstrations to elicit recovery (Laskey et al., 2017), and RLIF learns from intervention signals (Luo et al., 2024). CGSR instead edits a source under an explicit endpoint contract, with generator validity and subsequent policy learning evaluated as separate outcomes.

Table 5: CGSR's development advantage concerns coverage and cost relative to particular baselines. Classification and random ordering do not separate it on this pool. All attempt costs, including rejections and prefix replay, enter steps per verified source.

Method	Verified	Total steps	Steps/verified	Prefix (%)	Suffix steps
CGSR	9/9	4,833	537.0	47.0	182.6
Fixed pre-place	7/9	4,125	589.3	49.9	165.0
Full recollection	8/9	5,075	634.4	0.0	388.1
No classification	9/9	4,833	537.0	47.0	182.6
Random source ordering	9/9	4,833	537.0	47.0	182.6
Early generation + final filtering	3/9	4,542	1,514.0	25.8	214.7

The common-success comparison explains the apparent locality difference: fixed pre-place and CGSR preserve identical actions and generate equally long suffixes on seven shared successes. CGSR's lower overall retention reflects two additional recoveries from the earlier transport boundary. The corrected final-filter comparator changes the generation boundary as well, making it a coupled implementation comparison. A pure filter contributes no repairs to an already-failed pool. Random source ordering reuses the same successful repair bundles and therefore has the same locality, rather than supplying an independent recovery mechanism.

A.2 Frozen extension to previously unattempted sources

To examine locality beyond the initial development sources, we conducted a separately frozen exploratory extension after the original expansion gate had failed. The 95-source diagnosis comprised 42 passes, nine geometry failures with usable phase boundaries, and 44 geometry failures lacking boundaries in the truncated audit. Recovering full phase logs restored those boundaries after matching source IDs, action counts, and file hashes. Before new repair outcomes, a fixed-string-plus-ID SHA-256 ordering selected eight sources per color. These 16 sources exclude the original nine but remain within the same historical task and dataset. The extension is distinct from both new-scene evaluation and the original gate decision.

The comparison fixes the controller, placement proposals, and checker across adaptive CGSR, fixed pre-place, fixed transport, and full recollection; recollection retains the historical per-color grasp calibration. Each arm receives ceilings of 96 attempts and 96,000 control steps, with at most two

candidates per rollback boundary. A candidate begins at the original initial state, replays its retained prefix, and is accepted only when native containment and 2-mm supported placement hold for every final-50 state. Support includes floor contact and absence of gripper contact. Cost includes rejected candidates and prefix execution; subsequent student training follows a separate protocol.

Full-action replay first determines which saved-state failures persist under fresh execution. The repair-only comparison applies the resulting exclusion rule identically to all methods, while reporting all 16 selected sources. For each successful output, checks verify alignment among actions, observations, and pre/post states, and exact equality of retained source actions. Regenerating the observation and state streams ensures that prefix retention refers specifically to actions, without assuming unchanged images or globally shortest physical repair.

The extension uses one host and a fixed seed, with input-state restoration verified within 10^{-12}. Exact full-trace repeats for one source per color precede the formal comparison. These host-matched checks establish candidate-execution validity; the earlier cross-host restoration failures retain their original status. A second machine independently recomputes arithmetic, prefix equality, and recorded final-50 clauses. This separates independent record verification from physical replication and from invariance under another tokenizer or runtime.

Table 6: Complete extension results. Prefix, replaced fraction and generated suffix are means among successful outputs; steps include all candidates for the stated source pool. The two panels describe overlapping populations, not independent experiments.

Method	Verified	Total steps	Steps/valid	Prefix (%)	Replaced (%)	Suffix
All 16 selected sources						
CGSR	16/16	6,468	404.2	54.4	45.6	167.0
Fixed pre-place	15/16	6,118	407.9	55.6	44.4	162.8
Fixed transport	12/16	7,147	595.6	37.7	62.3	229.2
Full recollection	9/16	10,510	1167.8	0.0	100.0	428.7
Nine fresh-execution failures (one red, eight white)						
CGSR	9/9	3,964	440.4	50.7	49.3	168.6
Fixed pre-place	8/9	3,614	451.8	52.4	47.6	160.9
Fixed transport	6/9	3,981	663.5	37.4	62.6	228.3
Full recollection	6/9	5,291	881.8	0.0	100.0	407.5

Complete accounting includes 19, 18, 21, and 24 attempts for CGSR, fixed pre-place, fixed transport, and recollection, respectively. Raw replay screening costs 5,212 steps, and a two-source, two-repeat smoke check costs 1,268. Together with all four arms, the generator extension uses 36,723 control steps and no training updates. The repair-only panel charges candidates on the common nine fresh failures; the all-source panel also includes the seven replay-passing sources. Screening and setup are disclosed as common overheads alongside these candidate costs.

A source-only filter retains seven unchanged action sequences under fresh replay (7/16 sources, 100% action retention) after the 5,212-step screen. Using them as fresh-validated supervision additionally requires regenerated, aligned observations. Since the filter repairs 0/9 fresh failures, its steps-per-verified-repair ratio is undefined; it is distinct from the historical generation-plus-filter comparator in Table 5. On the six failures repaired by both CGSR and recollection, CGSR preserves 52.3% of source actions and generates 248 fewer suffix steps on average. This same-template comparison covers one red and eight white failures; Appendix A.3 separately examines the later learning outcome.

A.3 Exploratory student comparison after the locality extension

This comparison follows the full acquisition pipeline into student training. Both arms start from 42 qualified base demonstrations and add their entire verified yield from the same 16-source stream, producing CGSR58 and full51 with 58 and 51 demonstrations. Thus dataset size is an output of acquisition, rather than a controlled rollback factor. The reported training seeds are 0, 3, and 4, each using 600 updates and 4,800 draws per color. Each final checkpoint executes 600 steps and 150 policy queries on the same 25 red and 25 white starts per seed. Strict completion combines calibrated 2-mm

Table 7: **CGSR58 leads full51 in the reported three-seed exploratory comparison.** Counts are successes out of the same 50 starts per training seed; the total reuses those starts three times and is not 150 independent test cases. Strict is calibrated 2-mm final-50 success with support and release; native is contact-free native final-50 success.

Arm	Seed 0	Seed 3	Seed 4	Total
Strict / calibrated				
CGSR58	13	13	14	40/150
Full51	6	7	7	20/150
Native contact-free				
CGSR58	27	22	22	71/150
Full51	12	14	10	36/150

geometry, floor support, and release throughout the last 50 steps; native contact-free completion substitutes the native goal-region check over the same window.

CGSR58 exceeds full51 by $+20/150$ strict completions and $+35/150$ native completions. The strict differences are $+7, +6, +7$ per 50 starts across seeds; native differences are $+15, +8, +12$. Because the three seeds reuse test-start identities and the acquired dataset sizes differ, these descriptive effects concern the full pipeline. They do not independently isolate adaptive rollback or establish broad downstream superiority. As in Table 6, candidate cost excludes the separately reported common screening/setup overhead. The archive records acquisition and policy evaluation as distinct stages.

A.4 Complete primary supervision-learning counts

The same-goal supervision intervention provides the equal-size controls in Table 4, complementing the separate acquisition-pipeline comparison in Section 4.5. The goal and update budget are fixed, while five data arms change which demonstrations supply supervision. Figure 9 and Table 8 retain all primary seed-level outcomes, including the larger Mixed arm.

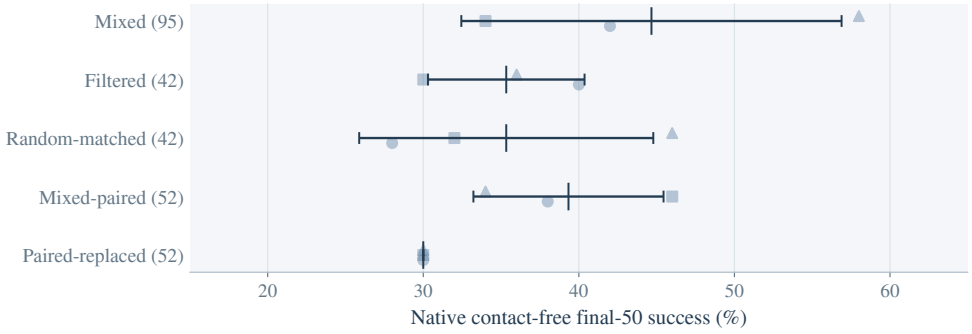

Figure 9: **Same-goal data selection has heterogeneous learning effects.** Native contact-free final-50 rates are evaluated on 50 shared starts per seed. Faint symbols indicate the three training seeds; black marks show mean ± sample SD. The paired-replacement arms use historical verified teachers, not CGSR outputs.

A.5 Audit questions and placement sensitivity

The following tables make the audit's inference boundaries and the unchanged tolerance sensitivity explicit.

The appendices are organized by question. For task conflicts, see the candidate decisions and caddy traces (Appendices C and D). For learning, see reward hits and demonstration training (Appendices H and E). Geometry calibration is in Appendices L and P; instruction selection and execution checks are in Appendices M and N. Query history is in Appendix I. Definitions and reproduction instructions are in Appendices O and Q, with SIMPLER selection history and paired comparisons in Appendix R.

Table 8: Same-task supervision changes do not yield uniform gains. Entries are native contact-free final-50 counts out of 50 shared held-out starts per seed; all 15 primary runs are shown. Geometric endpoint changes and the complete exploratory seed-direction block are in Appendix W.8. These runs use a different budget from the historical 4,200-update adaptation.

Training data	Sources	Seed 0	Seed 1	Seed 2
Mixed	95	21	17	29
Filtered	42	20	15	18
Random-matched	42	14	16	23
Mixed-paired	52	19	23	17
Paired-repaired	52	15	15	15



Figure 10: Task reference and completion requirements answer different questions. **A:** QueST, 18 demonstrations/600 updates, mean native first-hit success over three seeds on 40 shared cases; both arms receive outside text. **B:** Three frozen final Qwen models, 50 shared starts each, with native and calibrated geometric final-50 measured directly in the same executions. Both checks require no gripper contact. In B, symbols identify training seeds, not additional independent starts. Historical rotation-reconstruction limits are described in Appendix L.

Evidence and compute scope. The paper reports completed comparisons with their local source, start and training-seed denominators. Repeated starts, stored-action replays and teacher-controller transitions are not additional independent policy evaluations. Contact outcomes from stale collectors and the unfinished 480-run historical Qwen extension are excluded. Complete workload, interruption and resource histories remain in the experiment records. Each GPU training/evaluation worker uses one device; incomplete historical timing prevents a defensible total-study GPU-hour estimate. Generation, optimizer updates and policy-evaluation actions are reported separately rather than combined into an efficiency score. The locality extension's generator phase uses 36,723 simulator control steps and no training updates; the later student study is separate (Appendices A.2 and A.3).

B Temporal Evaluation and Replay

MINERVA confirmation. We use MINERVA's released `t05_l1_0.54M` checkpoint at revision `1b4fb174` and the authors' code at `64c3cc8`. Each task uses 50 initial states, seeds 1000–1049, hard resets, one action per replan and temporal-ensemble coefficient 0.01. The official horizons are 280/280/300/520 for Spatial/Object/Goal/Long. These settings follow the published L1 protocol, with OSMesa rendering in place of EGL. All 130 runtime BDDL files are byte-identical to those in the source census. After first success, the same policy takes ten further closed-loop steps without reset.

LIBERO's demonstration collector requires first success followed by ten more successful checks. A failure clears the counter, allowing a later successful window. By contrast, its evaluator uses `dones[k] = dones[k] or done[k]` to retain any success and stops when all environments

Table 9: The same recorded caddy behavior satisfies the outside goal, not the declared compartment goal. "Decl.-ref." uses native `In`; all trajectories are scored against both goals.

Source	n	Shipped	Decl.-ref.	Gap (pp)
Official demonstrations	100	100	0	100.0
QueST SFT	100	86	0	86.0
QueST + RIPT	100	94	0	94.0
Qwen2.5-OpenVLA baseline	100	79	0	79.0

Table 10: The audit distinguishes task conflict, monitor behavior and endpoint sensitivity. Each requires a different witness and supports a different inference.

Question	Executable evidence	Interpretation boundary
Task correspondence	Declaration and runtime goals disagree on the same state; actual policy text is traced.	A component conflict, not necessarily policy disobedience or a uniquely determined repair.
Monitor query semantics	Extra reads change history at a fixed sensor epoch.	An interface-dependent event; its frequency in policy rollouts remains unmeasured.
Completion definition	One execution is scored by stated native, geometric or temporal requirements.	A proxy-dependent score gap, not proof that the stricter definition is uniquely correct.
Learning consequence	Within-stage rewards or demonstrations change under a matched budget.	Evidence of reward support or local adaptation, not general conditional instruction following.

have succeeded at least once. MINERVA's wrapper similarly uses `terminated = done or is_success`; its vector environment resets automatically on the next step.[1]

Zero-action follow-up and missing replays. To distinguish continued-policy effects from passive loss, the frozen follow-up selects 20 persistence-loss cases per suite by SHA-256 ordering of suite, task, and episode IDs. Each case is replayed to first success before ten zero actions are applied. First-hit reproduction succeeds on 54/80 cases: 5 Spatial, 20 Object, 20 Goal, and 9 Long. Spatial and Long stop after a selected trajectory fails in two attempts. Because replay failure determines missingness, the completed subset supports only case-specific hold diagnostics, not an overall MINERVA hold estimate.

C LIBERO Task-Definition Census

Suite task counts are 10 Spatial, 10 Object, 10 Goal, 10 LIBERO-10, and 90 LIBERO-90. Placement goals appear in 116/130 tasks, articulation goals in 26, and multiple goal atoms in 20. The supplement's `revision/census/candidate_index.json` records all 14 candidates selected by deterministic cues. It includes task paths, declarations, triggering cues, goals, region owners, source lines and the reasons for each decision. One manual pass checks the complete instruction, region, initial-condition and goal blocks. It accepts equivalent encodings, identifies unstated conditions that must be maintained, and confirms the two compartment/table conflicts. Automated checks verify candidate coverage, source bytes and the explicit conflict/maintenance assertions; they do not estimate missed defects or agreement between annotators. The supplement contains all 130 pinned BDDL files, the census and decision records, and the reused parser/verifier. Run `python run_revision.py` to reproduce and verify the index. The same screening rule can flag both a real conflict and a valid encoding. Table 13 shows one of each from the frozen index judged by a single reviewer. Both use an `On` goal without an `on` or `stack` cue in the instruction, triggering `goal_on_without_on_or_stack_cue`. This rule selects cases to inspect; a person must still judge whether the instruction and goal agree. We retain the original judgments and do not estimate screening recall.

[1] Source links at the evaluated revisions: LIBERO collector, LIBERO evaluator, and MINERVA wrapper.

Table 11: The placement gap persists across the reported nonzero tolerances. Counts are from the same directly recorded runs in Table 45, not new evaluations. Each seed reuses the same 50 starts. All columns require the full final-50 window and no gripper contact.

Training seed	n	Native	Geometry 1 mm	Geometry 2 mm	Geometry 5 mm
0	50	32	12	14	22
1	50	36	13	16	25
2	50	30	8	11	16

Table 12: Goal atoms by predicate over all 130 BDDL tasks.

Predicate	On	In	Close	Open	TurnOn	TurnOff	Total
Atoms	61	63	11	7	8	1	151

The supplement records these paths, judgments, reasons and source hashes in `revision/census/candidate_index.json.gz`. Running `run_revision.py` checks the captured native filename parser and policy calls for the red/white caddy tasks.

D Source and Behavioral Evidence for Task Disagreement

Unstated stove condition. Two files, `libero_10/KITCHEN_SCENE8_put_both_moka_pots_on_the_stove.bddl` and `libero_90/KITCHEN_SCENE8_put_the_right_moka_pot_on_the_stove.bddl`, require `TurnOn(flat_stove_1)` in both the initial state and the goal. Their instructions request pot placement, without asking the robot to turn on the stove. Success therefore also requires preserving an unstated initial condition.

Caddy compartment definitions. The red-mug file, `libero_90/STUDY_SCENE3_pick_up_the_red_mug_and_place_it_to_the_right_of_the_caddy.bddl`, and its white-mug counterpart declare placement in the "right compartment of the caddy". Their goals instead check a region anchored to the table. Both files define the caddy's `right_contain_region`, but neither uses it in the goal.

We evaluate the declared compartment goal with LIBERO's native `In` check. Its implementation compares an object reference position with an approximate region. It does not check containment of the full mesh or release from the gripper.

We score 100 complete demonstration traces. All 50 per task satisfy only the shipped goal. QueST SFT also satisfies only that goal on 47/50 red and 39/50 white starts; RIPT does so on 50/50 and 44/50. Neither reaches the native compartment goal.

Paired rescoring quantifies disagreement between the two task references on identical trajectories. Exact p-values are 1.58×10^{-30} for demonstrations, 2.58×10^{-26} for SFT, 1.01×10^{-28} for RIPT, and 3.31×10^{-24} for Qwen. These tests concern reference-specific labels, whereas method rankings and the prevalence of conflicts require different comparisons.

The criteria set before demonstration scoring require at least 80 shipped-goal hits, at least 20 more shipped-only than compartment-only traces, exact paired $p < 0.05$, and the same direction in both tasks. Each released QueST checkpoint must have at least 20 shipped hits, a net excess of ten shipped-only trajectories and exact $p < 0.05$. Qwen uses those criteria and must also favor the shipped goal in both tasks.

Figure 3 shows the corresponding red-cup demonstration and state-placement control.

E Qwen Policy Evaluation and Matched Supervision

We evaluate the released `libero-qwen25-openvla` direct-action baseline. This is the baseline, rather than the reasoning model refined by R&B-EnCoRe. The source revision is `1da651f2`, the model-repository revision is `76cd50d8`, and the checkpoint SHA-256 begins `920274d38fbb`.

Table 13: Two cases selected by the same screening rule: one instruction–goal conflict and one valid encoding. Paths are relative to the pinned LIBERO BDDL root. Initialization atoms come from source and do not establish physical poses.

Task and instructions	Source initialization and scored goal	Region ownership and frozen judgment
`libero_90/STUDY_SCENE3_pick_up_the_red_mug_and_place_it_to_the_right_of_the_caddy.bddl` **Declaration:** "pick up the red mug and place it to the right compartment of the caddy." **Default policy text:** "pick up the red mug and place it to the right of the caddy."	**Initialization** (`:init`, 112–117): `On(desk_caddy_1, study_table_desk_caddy_init_region)`; `On(black_book_1, study_table_desk_caddy_front_left_contain_region)`; `On(porcelain_mug_1, study_table_porcelain_mug_init_region)`; `On(red_coffee_mug_1, study_table_red_coffee_mug_init_region)`. **Goal** (`:goal`, 119–121): `And[On(red_coffee_mug_1, study_table_desk_caddy_right_region)]`.	The scored `desk_caddy_right_region` belongs to `study_table` (lines 71–81). The available `right_contain_region` belongs to `desk_caddy_1` (82–84), compiling to `desk_caddy_1_right_contain_region`. **Confirmed language–goal conflict:** the declared instruction asks for a compartment, but the goal selects the table region. The available compartment region is unused. The policy receives a separate instruction requesting placement outside.
`libero_90/STUDY_SCENE1_pick_up_the_yellow_and_white_mug_and_place_it_to_the_right_of_the_caddy.bddl` **Declaration:** "pick up the yellow and white mug and place it to the right of the caddy." **Factory-derived text:** the same string. We have not captured a policy execution for this example.	**Initialization** (`:init`, 78–82): `On(desk_caddy_1, study_table_desk_caddy_init_region)`; `On(black_book_1, study_table_black_book_init_region)`; `On(white_yellow_mug_1, study_table_white_yellow_mug_init_region)`. **Goal** (`:goal`, 84–86): `And[On(white_yellow_mug_1, study_table_desk_caddy_right_region)]`.	The scored `desk_caddy_right_region` belongs to `study_table` (lines 38–48); `right_contain_region` belongs to `desk_caddy_1` (49–51). **Equivalent region encoding; excluded as conflict:** the instruction asks for placement to the right of the caddy, which the table region implements. An unused compartment region does not by itself imply a conflict.

The protocol records full hashes, including that of the vector-quantized (VQ) action tokenizer. Qwen and QueST use the same two tasks. Qwen's predefined criteria match QueST's and additionally require both tasks to favor the shipped goal.

We follow the release's `prismatic` preprocessing: vertical image flipping, 224-pixel Lanczos resizing, a center crop retaining 0.9 of image area, and bilinear resizing. Actions use `libero_90` unnormalization and the official gripper binarization/sign inversion. Center cropping, 50 trials and ten settling actions follow evaluator defaults; the README example uses no crop and 20 trials. We compare our transforms with the official implementation on 20 asymmetric images and seven gripper boundary cases.

The released LIBERO-90 evaluator allows 400 actions. We use 600 to match QueST, and also score the first 400 actions as a predefined sensitivity check. These prefixes yield 79/100 shipped and 0/100 native-`In` successes (paired exact $p = 3.31 \times 10^{-24}$). Full evaluations use one GPU and total 2.43 episode-hours, excluding loading.

The synchronous baseline cannot import its unavailable asynchronous reasoning dependency. We replace that import with a shim that raises an error if the asynchronous client is instantiated. We also remove an unused eager training-data import and apply the recorded trusted-tensor loading patch in LIBERO. Extra wrapper-level action clipping is removed after the first action is generated, before any policy action is executed. Cases, seeds, transforms, metrics and decision rules stay fixed. Source diffs, hashes and complete predicate traces are included in the artifact.



Figure 11: White-mug demonstration and explicit state-placement control, using official `demo_0` at left. The panels retain the original scene images and distinguish the runtime and declaration-referenced goals. The right panel is not a policy execution.

On the red task, Qwen has 32/50 shipped successes and 0/50 native-`In` successes. White-task counts are 47/50 and 0/50. Overall percentages, with Wilson 95% intervals, are 79.0 [70.0,85.8]% for the shipped goal and 0.0 [0.0,3.7]% for native `In`. Both goals are checked on every trajectory.

E.1 Matched LoRA training

We fine-tune the 1.26B-parameter Qwen2.5-OpenVLA base with rank-16 LoRA on attention and multilayer-perceptron projections. Its 8.80M trainable parameters exclude the vision backbone and multimodal projector. The two conditions use the same nine source demonstration IDs per task. Original demonstrations contain 1,946 transitions; compartment demonstrations contain 5,500. Inverse-length sampling gives each of the 18 demonstrations equal probability within its condition. Both conditions use the released image flip, DINOv2+SigLIP transform, LIBERO-90 q01/q99 action normalization, gripper conversion and eight-action VQ window.

All six runs use seeds 0–2, 600 optimizer steps, effective batch size 16 and learning rate 2×10^{-4}. We use 30 warmup steps, gradient clipping at 1 and the final checkpoint, with no selection based on performance. A preliminary loss/gradient check makes no weight updates and precedes protocol freeze. Each adapter executes 600 steps on official indices 30–49 in both tasks. Native-`In` first hit is primary; final-50 and shipped first hit are secondary. A recorded path correction occurs before constructing the environment, preserving checkpoints, cases, seeds, metrics and budgets.

E.2 Matched supervision results

The baseline and six adapters each run 40 episodes of 600 steps: 280 episodes and 168,000 actions in total. The baseline reaches the shipped goal on 34/40 cases and never reaches native `In`. Original-data seeds have shipped counts $(27, 27, 30)/40$ and no native-`In` first-hit or final-50 successes. Compartment-data seeds never reach the shipped goal. Their native first-hit counts are $(0, 5, 4)/40$, and final-50 counts are $(0, 3, 3)/40$. Across seeds, first-hit success is $7.5 \pm 6.6\%$, and final-50 success is $5.0 \pm 4.3\%$. These SDs use three training seeds, each evaluated once.

First-hit gains over matched controls are 0, 5, and 4 cases across seeds (exact two-sided $p = 1, 0.0625, 0.125$); final-50 gains are 0, 3, and 3 ($p = 1, 0.25, 0.25$). Seed-wise reporting preserves the shared task–state identities. Qwen uses indices 30–49 and QueST 10–29, so the cross-model difference also changes evaluation cases. In this initial stage, both training and evaluation use

filename-derived outside instructions, and training omits the inference crop. The demonstration intervention changes actions, lengths, and observations jointly; the absence of execution repeats leaves training and execution variation combined.

E.3 Expanded Qwen training and confirmation

We expand the collection to all 50 source demonstration starts per task, using the same privileged teacher and robot actions. Of 100 attempts, 46 red and 49 white trajectories pass native success. The original-data control uses the same source IDs. The compartment dataset contains 29,881 transitions and the original dataset 10,412. Teacher starts have no exact simulator-state overlap with development or either confirmation block (time excluded; tolerance 10^{-6}). All five failed attempts remain in the artifact.

In this stage, training and evaluation both receive the BDDL compartment instruction explicitly. Every policy call records the default and supplied text. Training now uses the inference crop. Stored and runtime RGB images are 128 pixels before the released 224-pixel transform. The controller executes all eight VQ-decoded actions before replanning; we compare its first action with the unmodified predictor and clear action queues at reset. The policy selects actions using only RGB and instruction text. Goal labels, poses and contacts are logged for evaluation and do not control the robot. Every episode takes ten settling actions and 600 policy actions, without stopping early at success.

Before development results, we specify three configurations in order: 128-pixel/eight-action, 128-pixel/four-action and 256-pixel/eight-action. Each is tested after 600 and then 1,200 additional updates. We select the first ten-case result with at least eight first hits and seven final-50 successes. The first configuration meets this rule after 600 further updates, scoring 8/10 on both measures. All six models continue their respective 600-update adapters for 600 more updates, with fresh AdamW at 10^{-4}. These 1,200-update models are then evaluated on the same 40 pairs at indices 10–29. Seed-0 development also completes 1,200 additional updates, but those later weights are not selected.

After the 40-case evaluations, we specify checkpoints at 1,000/2,000/3,000 additional updates. All six 1,200-update adapters resume with saved AdamW moments, batch and microbatch size 16, learning rate 10^{-4} and clipping at 1. To save memory, we project vocabulary logits only at supervised next-token positions, preserving the full causal context and masked cross-entropy objective. Two checks without weight updates agree in loss within 1.2×10^{-7} and trainable-gradient relative L_2 within 1.01% under BF16. The 2% tolerance checks numerical agreement, not bitwise identity. Interrupted optimization records are retained; none supplies a confirmation checkpoint.

Development uses official indices 0–4 per task. First-hit/final-50 counts at the three checkpoints are 7/7, 6/6 and 9/9 out of ten. The predefined rule selects the first checkpoint with at least nine first hits and eight final-50 successes. Thus 3,000 additional updates, or 4,200 total, are selected before confirmation results. Every seed uses the same budget and controller. Confirmation uses 25 states per task at indices 5–9 and 30–49, separate from development and the earlier 40-case block. Some states appeared in historical audits: they are withheld from this stage's model selection, but are not globally unseen. All three seeds share the 50 cases.

Table 14: Qwen confirmation counts. Each triple gives training seeds 0/1/2. The unadapted baseline is run once with the controller and explicit BDDL instruction used in that stage. CF means no gripper contact.

Total updates / n	Arm	Shipped	Native first-hit	CF native final-50
1,200 / 40	Unadapted	21	1	0
	Original	33, 33, 35	0, 0, 0	0, 0, 0
	Compartment	0, 0, 0	22, 27, 24	21, 26, 22
4,200 / 50	Unadapted	27	3	1
	Original	47, 42, 41	0, 0, 0	0, 0, 0
	Compartment	0, 0, 0	35, 36, 34	32, 36, 30

The baseline and six continued-training models execute 50 cases each, totaling 350 episodes and 210,000 actions. Compartment-data first-hit counts are $(35, 36, 34)/50$, and contact-free final-50

counts are $(32, 36, 30)/50$: seed-wise rates of 70/72/68% and 64/72/60%, respectively. Means and sample SDs are 70.0 ± 2.0 and $65.3 \pm 6.1\%$. Within-seed exact paired first-hit p-values are 5.82×10^{-11}, 2.91×10^{-11}, and 1.16×10^{-10}; final-50 values are 4.66×10^{-10}, 2.91×10^{-11}, and 1.86×10^{-9}. Each checkpoint is executed once, so uncertainty includes execution variation. Interpretation remains within stage because instructions, data, crops, controllers, budget, and cases change across stages.

F SIMPLER Completion-Window Sensitivity

The same saved SIMPLER traces support the one-, three-, and five-step window comparison in Table 15. Rescoring these traces changes the completion definition without adding policy executions.

Table 15: SIMPLER success counts after verifying matching resets and action prefixes. Intermediate simulator states were not hashed.

Policy	Position	Strict@1	Strict@3	Strict@5
Base π_0	50	44	40	40
Released CoVer	79	71	62	61
Reconstructed CoVer	74	74	68	65
Total	203	189	170	166

G SmolVLA Continuation and Contact Analysis

The independent 60-case SmolVLA screen finds first-hit/persistent counts of 15/5 on Spatial, 18/17 on Object, and 18/12 on Goal. Among 51 first hits, 17 are lost and 14 later recover. Object's single loss falls below the predefined three-loss expansion threshold, showing that continuation losses differ across suites and predicate families.

Replaying each Spatial first hit and then applying a seven-dimensional zero action isolates losses that can occur without further policy motion. Five of ten continuation losses recur under this hold, and four later recover. Across the 12 unsuccessful states in those cases, bowl–plate contact is always lost; only two cross the 3-cm horizontal threshold, and none violates the height condition. At the first failure of each case, horizontal distance remains below 3 cm. Contact loss therefore accounts for the initial predicate transition in these held-action examples.

H Reward Support and Demonstration-Based Adaptation

H.1 Demonstration-supported native-goal adaptation

The reward-only null motivates collecting demonstrations from source starts 0–9 of each caddy task. Each collection restores the initial state once, then follows robot actions throughout. An 80-action red-task or 75-action white-task replay establishes the grasp; privileged simulator feedback then guides lifting, transport over the compartment, lowering, opening, and retreat. This teacher information is excluded from student evaluation. Nine of ten attempts succeed per task, providing 18 traces that satisfy native `In` without mug–gripper contact throughout their final 50 states. Both failed attempts remain recorded. These demonstrations establish action-based reachability of the native goal; their collection did not require full-object geometric containment.

QueST fine-tuning retains the upstream stage-1 token-prediction loss and frozen action tokenizer. AdamW uses learning rate 10^{-5}, batch size 64, clipping at 100, and 600 updates. The original-data control matches the 18 source IDs and update budget, and both arms use their final checkpoint. Actions, lengths, and regenerated observations change together, so the comparison tests the full demonstration intervention rather than individual data factors or equalized collection cost.

The seed-0 pilot uses official indices 10–29 in both tasks: 40 episodes of 600 policy steps. The evaluator then applies 50 zero-motion, open-gripper actions. We check native `In` at any step, at step 600 and throughout this external contact-free hold. A later analysis also checks contact-free

Table 16: Training on original versus compartment demonstrations. Percentages are mean ± sample SD across three training seeds, each run once; SD therefore combines training and execution variation. First hit and contact-free final-50 both use native `In`. Compare within rows: protocols and cases differ between stages. Appendix E.3 lists case indices; geometry calibration is separate.

Policy / training setting	Original first-hit	Compartment first-hit	Compartment final-50
QueST, 18 demos / 600 updates	0.0 ± 0.0	60.8 ± 8.8	59.2 ± 7.6
Qwen, initial 18 / 600	0.0 ± 0.0	7.5 ± 6.6	5.0 ± 4.3
Qwen, expanded 95 / 1,200	0.0 ± 0.0	60.8 ± 6.3	57.5 ± 6.6
Qwen, continued 95 / 4,200	0.0 ± 0.0	70.0 ± 2.0	65.3 ± 6.1

native success at step 600 and through the final 50 policy steps. These states appeared in historical audits, but were not used for this collection or tuning; training starts are checked for exact-state overlap. Evaluations refresh observations after assigning the initial state and taking ten settling steps. The upstream `reset_ori` returned observations from before its final state assignment. Earlier evaluations are therefore not pooled with these as matched comparisons.

Five stable successes in the first ten red pilot evaluations, against zero in both controls, motivate adding training seeds 1 and 2 to both arms. All other settings remain fixed. The resulting seeds share 40 states and inference seeds, supporting seed-wise outcome reporting rather than additional task-level replications.

Compartment-data seeds 0/1/2 have native-`In` first-hit counts $(21, 24, 28)/40$, compared with $(0, 0, 0)/40$ for original-data controls. Their final-50 counts are 21, 23 and 27, versus zero in every control. External open-gripper hold counts are $(21, 24, 27)/40$ versus $(0, 0, 0)/40$. Mean first-hit success is $60.8 \pm 8.8\%$ and final-50 success is $59.2 \pm 7.6\%$. Execution repeats show that these SDs include more than training variation. Seed 1 loses final-50 success on white state 17. Seed 2 retains native `In` on white state 13, but continuing gripper contact disqualifies both step-600 completion and the external hold.

H.2 Execution repeats for every QueST checkpoint

One control runs the compartment-data seed-0 checkpoint eight times on indices 10–29. A separate predefined study uses indices 30–49 and runs each of the three original-data and three compartment-data checkpoints three times. We fix weights, evaluator, case IDs, official state bytes and recorded parent-policy seeds. The older reruns do not verify identical settled controller states or policy inputs; subprocess reset randomness differs from the recorded parent seed. They are therefore repetitions of fixed cases and checkpoints, without confirmed identical inputs. All 18 outputs contain 40 complete 650-step traces and pass runner/checkpoint hash checks. Original-data models score 0/40 for native first hit and final-50 in every run. Compartment-data counts are $(24, 25, 20, 27, 23, 30, 26, 29, 28)/40$ for first hit and $(23, 22, 18, 23, 23, 30, 24, 25, 28)/40$ for final-50. We express SDs within and between checkpoints in successful-case counts. They describe these three checkpoints and three runs, rather than population variance components.

Across compartment checkpoints, first-hit success averages 25.8/40. The between-checkpoint SD is 2.5 cases and the root-mean-square within-checkpoint SD is 2.7 cases. Final-50 success averages 24.0/40, with corresponding SDs of 2.6 and 3.0 cases.

H.3 Compartment-reward RIPT from compartment-data initialization

Before results, we specify two starting SFT checkpoints and three RIPT training seeds per checkpoint. Both conditions use compartment environments, the compartment-data reset prefix, 15 updates, 16 rollouts per update and RLOO groups of eight. Only the starting checkpoint differs, and each run uses its final checkpoint. These seeds replicate RIPT training from fixed initializations, not the earlier demonstration training. Pre- and post-training evaluations share 40 official task–state pairs (indices 30–49), a 600-step horizon and a separate 50-step hold. Each starting checkpoint is evaluated once; its result is reused for all three post-training comparisons. A loader correction precedes optimization and leaves cases and budget fixed.

Table 17: RIPT with compartment reward, starting from original-data or compartment-data SFT. Success counts use 40 shared states. Update and tensor entries give seeds 0/1/2. A "mixed" update contains both successful and unsuccessful reward outcomes.

Initialization	Pre lang.-ref.	Post lang.-ref.	Mixed updates	Changed tensors
Original demos	0	0, 0, 0	0, 0, 0	0, 0, 0
Compartment demos	25	36, 24, 32	15, 15, 15	77, 77, 77

Compartment-data initialization supplies both reward outcomes in all 15/15 updates. Across the three seeds, final-minibatch gradients are nonzero in 14/15, 14/15, and 15/15 updates, and each checkpoint changes 77 tensors. This is consistent with RLOO's dependence on within-group reward differences; a small floating-point residual in a centered batch mean alone is insufficient evidence of learning. Original-data initialization produces neither rewarded samples nor nonzero updates and remains at 0/40 native-`In` success.

After RIPT, compartment-initialized first-hit counts are $(36, 24, 32)/40$. Relative to the shared pre-training execution, gain/loss counts are 14/3, 12/13, and 12/5, with exact paired $p = 0.0127, 1, 0.1435$. Final-50 counts are $(36, 21, 26)/40$, versus zero from original-data initialization. Reward-support criteria hold in 3/3 seeds, but the behavioral threshold holds in 2/3. This distinction matters because the baseline is executed once on shared states. Tensor-identical original-data checkpoints also produce 36, 33, and 28 shipped successes under repeated per-case seeds, while native `In` stays at zero. Execution variability therefore remains a competing explanation for some post-training score changes.

H.4 Reward-only training

For reward-only training, we start from released QueST SFT weights and replace only the two BDDL goal atoms with native `In`. Optimization, demonstrations and filename-derived policy text stay unchanged. RIPT uses 15 updates, 16 rollouts per update, seed 0, 600-step horizons and learning rate 10^{-6}. Dynamic sampling is disabled: otherwise all-zero groups could be resampled indefinitely. Before results, we require ten native successes and a ten-case gain over unchanged SFT on the fixed 100-state evaluation before adding seeds 1–4. The same-budget shipped-reward control is specified after seeing the compartment diagnostics and is an exploratory training check.

Table 18: Same-budget RIPT development runs. Each arm counts hits of its own training goal, so reward hits do not describe the same physical event. The shipped-reward control was specified after observing the compartment null result.

Training reward	Updates	Rollouts	Reward hits	Changed tensors
Compartment `In`	15	240	0	0/338
Shipped table region	15	240	226	77/338

Both arms use the public SFT checkpoint, seed 0, batch size two, eight rollouts per RLOO group, 20 PPO epochs, learning rate 10^{-6} and a 600-step limit. Dynamic sampling is disabled in both to avoid indefinite resampling of all-zero compartment groups. Only the BDDL directory and output paths differ, although reward-based stopping can make sampled action/state paths diverge. Each arm has one training seed.

The log records the gradient norm of only the final minibatch in each update. All 15 compartment-reward values are zero, and every model tensor is unchanged. The shipped-reward control logs three nonzero values and changes 77 tensors. We compare all 338 tensor names, shapes and values exactly, removing only the distributed-training module prefix. This distinguishes parameter changes from serialization differences.

We evaluate the compartment-reward checkpoint on the same 100 official pairs as SFT, using 600 actions without goal-based stopping. Under the shipped check, 85 pairs succeed in both runs and three fail in both; seven succeed only for SFT and five only for the compartment-reward checkpoint. Predicate traces differ in 96/100 pairs despite identical weights, showing execution variation. Neither

run reaches native `In`. The shipped-reward control yields 47/50 red and 45/50 white shipped successes, and 0/100 native successes. Compared with earlier SFT, seven pairs change in each direction: aggregate success is unchanged despite tensor updates.

I RoboCasa Query-Invariance Experiments

Table 19: Unmodified source methods evaluated on mocked sensor sequences. Entries are final source-method labels under 1/2/4 queries per transition on the controlled sequences.

Task	Controlled sequence	Original	Once/transition
PreheatPot	250 hot-pot transitions, then vegetable inserted	F/T/T	F/F/F
RinseCuttingBoard	50 hot-water transitions, then faucet off	F/T/T	F/F/F
OvenBroilFish	Six fish-on-rack, closed-door transitions	F/T/T	F/F/F
CoolKettle	75 burner-off transitions, then move kettle	F/F/F	F/F/F
WipeTable	Six contacts at one position, then release	F/F/F	F/F/F
SanitizeSink	Six contacts at one position, then release	F/F/F	F/F/F

PreheatPot requires 500 calls, RinseCuttingBoard 100, and OvenBroilFish ten accumulated calls checked before the next increment. Querying twice at the same physical state can therefore meet a duration threshold earlier. The cached variant runs the extracted method once per transition and returns that answer for later queries. All cached monitor states and labels match the complete single-query traces.

CoolKettle updates its timer in `update_state`, separately called once per transition. The wiping methods reject repeated contact positions. After the initial results, we extend CoolKettle to 150 cooling transitions and give the wiping controls five contact locations spaced 3 cm apart. All three then succeed under every query count with identical traces. They are query-invariant in these tests for both positive and negative final labels.

Remaining direct-counter methods. After the six initial cases, we specify tests for every remaining method that directly increments a counter. JuiceFruitReamer uses six qualifying transitions; PrepareVeggiesForSteaming uses 25 followed by counter placement; RinseFragileItem uses 50 followed by rack placement. ClearClutter, DrainVeggies and PrepareVegetableRoasting each use six washing transitions followed by placement. All yield F/T/T under 1/2/4 queries. CheeseMixing's original `_detect_stirring` helper tracks the previous object position, preventing repeat queries from counting the same motion. With 3-mm movements, two transitions fail and three succeed under every query count. In total, nine direct-counter methods are sensitive and three invariant; CoolKettle is a separate external-clock control. Every cached state and label trace matches the single-query reference. The juicing test uses a fixed 180-degree x-axis rotation. Its orientation stub agrees with SciPy's quaternion conversion for that input only.

At revision `4f8a2980`, the inventory finds 353 directly defined `_check_success` methods. Seventeen write instance state and twelve directly increment counters. Inherited calls and indirect mutations are not screened, so these counts do not estimate defect prevalence. Success is queried by RoboCasa reward and, after stepping, by the Gym wrapper and collection loop. Robosuite calls reward in its post-action path. Any cache must identify the intended update boundary and be cleared at episode initialization.

J Extended MINERVA Results

A later endpoint check finds 1838/2,000 successful states at the tenth follow-up (91.9%), versus 1,687 windows that remain successful throughout. The last-state check allows intermediate failures; the predefined persistence measure does not. Neither counts eventual completion under the collector's restart-after-failure rule.



Figure 12: **Completion requirements change reported success across the audited settings.** Each pair compares the recorded relation or first-hit event with the displayed completion check. SIMPLER uses the frozen five-state endpoint, including its original motion constraint; SmolVLA and MINERVA use ten continued-policy steps. Counts and denominators remain specific to each setting. Figure 8 separately shows the instruction-revision effects.

Table 20: Prespecified MINERVA audit. "Published ref." uses the released result under a different renderer; only Hit, Persist@10, and Lost are paired within our run.

Suite	Published ref.	Hit	Persist@10	Lost
Spatial	96.8	95.4	77.0	92
Object	99.6	100.0	93.6	32
Goal	97.4	95.6	83.4	61
Long	89.2	88.2	83.4	24
Overall	95.75	94.8	84.4	209

Table 21: MINERVA persistence statistics with Wilson 95% intervals. Rates are percentages. Episode-level intervals are conditional on these 40 tasks and are not uncertainty over unseen tasks.

Suite	Episodes	Hit [CI]	Persist [CI]	Rel. reduction	Recovery
Spatial	500	95.4 [93.2,96.9]	77.0 [73.1,80.5]	19.3	84
Object	500	100.0 [99.2,100.0]	93.6 [91.1,95.4]	6.4	32
Goal	500	95.6 [93.4,97.1]	83.4 [79.9,86.4]	12.8	27
Long	500	88.2 [85.1,90.7]	83.4 [79.9,86.4]	5.4	18
Overall	2,000	94.8 [93.7,95.7]	84.4 [82.7,85.9]	11.0	161

Table 22: MINERVA results for each task (50 episodes each). Hit counts any native success; P@10 requires the predicate to hold throughout the next 10 steps while the policy continues. Lost counts hits that fail this window. S/O/G/L denote Spatial, Object, Goal, and Long.

Task	Hit	P@10	Lost	Task	Hit	P@10	Lost
S0	50	24	26	G0	50	50	0
S1	48	40	8	G1	48	48	0
S2	50	47	3	G2	42	39	3
S3	50	48	2	G3	46	46	0
S4	46	37	9	G4	49	39	10
S5	47	37	10	G5	50	50	0
S6	47	39	8	G6	45	40	5
S7	46	41	5	G7	50	50	0
S8	47	35	12	G8	48	42	6
S9	46	37	9	G9	50	13	37
O0	50	40	10	L0	43	36	7
O1	50	50	0	L1	45	45	0
O2	50	49	1	L2	45	44	1
O3	50	49	1	L3	47	47	0
O4	50	50	0	L4	37	28	9
O5	50	30	20	L5	50	50	0
O6	50	50	0	L6	29	25	4
O7	50	50	0	L7	49	49	0
O8	50	50	0	L8	48	46	2
O9	50	50	0	L9	48	47	1

K FIXED-TASK OBSERVATION SYNCHRONIZATION

At RIPT-VLA revision `440990e`, `reset_ori` captures an observation before assigning an official initial state. Nested wrappers discard the regenerated observation and keep the preceding frame buffer. Both released QueST evaluation and RIPT training request this mode. Our change returns the regenerated observation and rebuilds the stack at size one, without extra simulator steps. Instructions, goals and later observation processing stay unchanged.

Before confirmation results, we choose one non-caddy task from each consecutive ten-task block of LIBERO-90, using the smallest task-name SHA-256. This gives nine tasks across six scenes. Indices 0–9, both released SFT/RIPT checkpoints and both reset settings give 360 trajectories of 600 actions. We fix weights and official state vectors. Pre-reset and policy-sampling seeds match between reset settings within a checkpoint, but differ between checkpoints. All outcomes are retained. Development includes 16 episodes with a prior caddy configuration and 16 with original task files; neither enters effect estimates. An earlier interface failure produced no outcomes. Confirmation completes within the 3,600-second cap on two shared RTX 4090 GPUs, without training.

Table 23: All reset-confirmation results, with ten cases per cell. L uses the pre-initialization observation; S uses the synchronized observation. Asterisks mark tasks that fail an additional observation-pairing check. Their outcomes remain in the matrix as descriptive results.

Task	Instruction	SFT L	SFT S	RIPT L	RIPT S
8	Open top drawer and put bowl in it	8	9	8	10
10	Put black bowl on cabinet	10	10	10	10
28*	Close top drawer	10	10	10	10
30*	Put black bowl on plate	8	9	10	10
40	Put frying pan on cabinet shelf	10	10	10	9
56	Pick up butter and put it in tray	10	10	10	10
69	Put chocolate pudding left of plate	7	8	10	10
72	Put white mug on plate	8	8	9	9
89	Put right book under cabinet shelf	9	10	10	10
Total	90 cases per checkpoint and arm	80	84	87	88

All 180 within-checkpoint pairs have equal initial simulator-state-vector hashes. Every legacy observation differs from the post-initialization observation, while every synchronized observation matches it. First action-token sequences differ in 179/180 pairs. However, observation hashes before and after initialization differ between reset settings in all 40 pairs for tasks 28 and 30 (Kitchen Scene 5). A later two-reset diagnostic on task 28 reproduces equal state vectors but different model-body positions and images. LIBERO stores fixture positions in `sim.model.body_pos`, outside the restored state vector. Equal state vectors therefore do not ensure identical worlds. These two tasks cannot isolate the effect of observation synchronization.

In the full matrix, RIPT exceeds SFT by 7.8 percentage points with L and 4.4 with S. Exact two-sided paired p-values are 0.092 and 0.219, or 0.185 and 0.438 after correcting for two comparisons. The change in advantage is -3.3 points. A 10,000-resample bootstrap over tasks gives a 95% interval of $[-7.8, 1.1]$; clustering by six scenes gives $[-7.5, 1.1]$. These intervals describe the matrix and do not resolve its pairing limitation. Synchronization changes SFT success by 4.4 points ($p = 0.388$) and RIPT by 1.1 ($p = 1.000$). The pooled change is 2.8 points. Both pooled success and the method-advantage change fall below the predefined 5-point material-consequence threshold.

On the seven tasks with matching observations across settings, SFT succeeds on 62/70 with L and 65/70 with S; RIPT succeeds on 67/70 and 68/70. This later diagnostic retains the method ordering and does not replace confirmation. The incorrect observation return is verified, but neither a material change in the comparison nor an ordering reversal is established. Effects on training remain untested.

L Behavioral calibration and aligned-supervision controls

L.1 Independent geometric placement criterion

The placement criterion operationalizes full containment, floor contact, and release. It is fixed before the new calibration outcomes and permits any cup orientation. These explicit physical conditions provide a reproducible measurement reference; comparison with human judgments is a separate validation step.

We derive the compartment boundaries from the caddy's compiled collision boxes: its floor, outer walls and divider. This definition does not use the native `right_contain_region` site. In caddy-local coordinates, the lower and upper corners are $(-0.06284, -0.20320, 0.01381)$ and $(0.06277, -0.07424, 0.16730)$ metres. We transform the cup's visual-mesh and collision-box vertices into this frame. Checking the convex-hull vertices is sufficient for containment in this rectangular volume. Let $E_t(\epsilon)$ indicate full-envelope entry with spatial tolerance ϵ, and let $P_t(\epsilon) = E_t(\epsilon) \wedge F_t \wedge \neg H_t$, where F_t is cup–floor contact and H_t is cup–gripper contact. Entry uses $\max_{t \le 600} E_t$; supported/released final-50 placement requires $\prod_{t=551}^{600} P_t = 1$. We fix 2 mm as the primary tolerance and 0/1/5 mm for sensitivity checks. Passing certifies only the listed geometry and contact conditions during the observed window. It does not establish a stationary or upright pose, force equilibrium, or real-world completion.

Existing final-Qwen trajectories. The frozen audit includes six final-checkpoint executions of 50 cases each and the 50-case unadapted baseline. The baseline saved native labels and contacts, but no actions or positions, so its 50 geometric labels are unavailable. All 300 trained-checkpoint trajectories can be audited. We replay their actions without cameras to reconstruct rotations that were not saved during the original runs. Each replay matches all 600 native `In` labels, gripper-contact flags and saved cup positions. Maximum position error is zero, within the acceptance tolerance of 10^{-6} metres, so no trained-checkpoint trajectory is excluded. This agreement checks the replay against the saved traces; it does not recover an independent observation of the original rotations.

Of the 105 compartment-data episodes that hit native `In`, 46 (43.8%) never place the full cup inside the compartment and 55 (52.4%) never also satisfy floor contact and release. These are false positives relative to our specified proxies, rather than a unique interpretation of the verbal task. At the frame level, 24,920 of 35,694 native-true frames fail full-envelope entry (69.8%). These dependent frames do not increase the independent sample size. The native contact-free final-50 check passes 98 cases; the supported/released geometric check passes only 34. Mean full-envelope entry is $39.3 \pm 6.4\%$, supported/released first hit $33.3 \pm 4.6\%$, and supported/released final-50 $22.7 \pm 3.1\%$. These sample

Table 24: Geometry checks on the original final-Qwen executions. Each count is out of 50; triples list training seeds 0/1/2, with one execution each. Native final-50 also requires no gripper contact. Geometry uses the fixed 2-mm tolerance.

Endpoint	Original-data models	Compartment-data models
Native `In` first hit	0,0,0	35,36,34
Full-envelope entry	0,0,0	21,22,16
Supported/released placement first hit	0,0,0	18,18,14
Contact-free native final-50	0,0,0	32,36,30
Supported/released geometric final-50	0,0,0	13,11,10

SDs summarize three original checkpoint executions and combine training and execution variation. The original-data models score zero on every endpoint.

Table 25: Tolerance sensitivity on the same compartment-data traces. Counts are out of 50 and list seeds 0/1/2. Original-data models score zero at every tolerance.

Tolerance	Full-envelope entry	Supported/released hit	Supported/released final-50
0 mm	2,6,3	0,0,0	0,0,0
1 mm	19,13,12	15,10,12	9,8,7
2 mm	21,22,16	18,18,14	13,11,10
5 mm	24,25,17	23,25,17	22,25,14

The choice of spatial tolerance affects the measured success rate. Supported/released final-50 placement averages 16.0% at 1 mm and 40.7% at 5 mm, compared with the native rate of 65.3%. No sustained supported placement passes at zero tolerance. We retain the primary 2-mm value rather than adjusting it after seeing the results. Across these checks, compartment supervision improves placement, but hitting native `In` remains substantially easier than placing the full cup inside the volume.

L.2 Language and reward aligned at the original budget

We test whether aligning both language and reward restores learning at the original budget. Starting from the same released QueST SFT checkpoint, we change the two policy-facing instructions to match the BDDL compartment declarations before building dataset and runner task embeddings. We record the actual descriptions and embedding hashes; the goals already use native `In`. The remaining settings stay fixed: seed 0, 15 updates of 16 rollouts, RLOO groups of eight, a 600-step horizon, 20 PPO epochs, learning rate 10^{-6}, no dynamic sampling and the legacy training reset. We specify this follow-up after the original null result and before seeing its own outcomes.

The aligned run completes all 15 updates with 0/240 reward hits; a separate initial diagnostic also scores 0/16. Every logged final-minibatch gradient norm is zero. Comparing all 338 model tensors against public SFT finds zero changed tensors, maximum absolute difference zero and relative L_2 difference zero. Thus language alignment does not produce reward hits in this fixed protocol. This one-seed result does not establish that reward revision generally fails, separate all exploration and reset effects, or show that demonstrations are the only successful route. We do not use a new post-training behavioral evaluation to support this null result.

L.3 Final-checkpoint execution repeats

Before examining new outcomes, we select the final seed-0 compartment-data Qwen adapter and its matched original-data adapter. Each receives two additional executions on the same 50 official task–initial-state pairs; the original final confirmation is the first execution. We change only episode seeds, using fixed offsets of one and two million. We retain the weights, source evaluator, compartment text, 128-pixel rendering, image crop, eight-action execution and 600-step horizon. We check source and adapter hashes and the logged geometry. These repeats measure execution variation for one

training seed. They add no training replications and do not estimate all sources of uncertainty in the three-seed result.

Table 26: Three executions of each selected final-Qwen seed-0 adapter. Triples count successes out of 50 in the original run and two new runs. Geometry uses the fixed 2-mm tolerance.

Endpoint	Original data	Compartment data
Native `In` first hit	0,0,0	35,33,34
Contact-free native final-50	0,0,0	32,27,32
Full-envelope entry	0,0,0	21,14,17
Supported/released geometric final-50	0,0,0	13,9,7

Across the three executions of the compartment-data adapter, native first-hit success is $68.0 \pm 2.0\%$ (range 66–70%), native contact-free final-50 is $60.7 \pm 5.8\%$ (54–64%), full-envelope entry is $34.7 \pm 7.0\%$ (28–42%), and supported/released full-envelope final-50 is $19.3 \pm 6.1\%$ (14–26%). These means and sample SDs describe repeated executions of one fixed adapter, rather than variation across training seeds. The original-data adapter scores zero on every endpoint in all three executions. The repeats preserve the advantage of compartment supervision and the gap between native success and geometric placement. They do not establish broader task competence or separate population-level variance components.

L.4 Direct geometry logging in new QueST executions

We run the prespecified final seed-0 QueST pair on the original 40 official task–initial-state pairs, using indices 10–29 for each task. We retain the batched evaluator, checkpoints, seeds, observations and action loop. Passive instrumentation records geometry without changing actions or stopping conditions. All 40 official-state and checkpoint hashes match in each arm, and geometry and native traces agree on gripper contact at all 650 steps. We analyze the first 600 policy steps and exclude the 50-step external diagnostic suffix. These are new executions. They do not supply missing geometry for the old QueST trajectories, which saved neither positions nor actions.

Table 27: New QueST seed-0 executions. Counts are out of 40, with geometry measured at the fixed 2-mm tolerance. Each selected checkpoint runs once on the two tasks that share a scene template.

Endpoint	Original data	Compartment data
Native `In` first hit	0	25
Full-envelope entry	0	16
Supported/released placement first hit	0	13
Contact-free native final-50	0	24
Supported/released geometric final-50	0	11

Geometric final-50 success is 27.5%, compared with 60.0% for native contact-free final-50. The matched original-data model scores zero. This fixed-checkpoint diagnostic finds the same measurement gap in the other architecture, within the same scene template. It does not compare architectures or test a new scene.

L.5 Paired-instruction diagnostic

We also freeze a small instruction diagnostic before seeing its outcomes. It uses the same final seed-0 Qwen pair. A fixed hash rule selects five official initial-state indices (30, 34, 39, 45, 48) for each task. Each model runs both external-placement and compartment text on these ten cases, for 40 new episodes. We retain the weights, episode seeds, 128-pixel observation, eight-action controller and 600-step horizon. We save the actual policy-input text and verify that all ten paired initial policy RGB hashes match within each model. There is no optimization or selection based on results.

To pass the prespecified joint check, a model must satisfy native contact-free final-50 `On` under external text and native contact-free final-50 `In` under compartment text on the same RGB-verified

Table 28: Paired instruction diagnostic, with each count out of ten. Final-50 requires the native goal and no gripper contact throughout the last 50 policy steps. These proxies do not check full geometry or human judgments of completion.

Training data	Instruction	`On` hit	`On` final-50	`In` hit	`In` final-50
Original	External	10	3	0	0
Original	Compartment	10	2	0	0
Compartment	External	0	0	7	5
Compartment	Compartment	0	0	5	4

start. Both models score 0/10. The diagnostic therefore does not show that either adapter selects and completes both requested behaviors. The observed goal pattern remains associated with the training data under both texts. Ten cases from one scene template per model cannot establish general language insensitivity, rule out conditional capability, or demonstrate broader competence. We retain this null separately from the positive matched-supervision result.

M Asset Calibration and Controlled Task Labels

The joint tests below ask whether a policy completes both requested goals from the same start, once under each instruction. We use three versions of the compartment check: native, calibrated geometry and original visual geometry. In all three, exterior placement requires native `On` and no gripper contact throughout the final 50 steps. The calibrated and original-visual versions change only the compartment check. Their joint results therefore combine two different proxies; they do not independently certify full physical placement for both goals. Native `On` checks whether the body origin lies in a region and vertical band. It does not require table contact, full-object containment, an upright pose or a stationary pose. We also report a stronger exterior auxiliary: throughout the final 50 steps, the body origin must lie inside the native site's XY rectangle, the cup's full visual XY envelope must be separate from the caddy box envelope, and the cup must touch the table without gripper contact.

M.1 Physical positive controls and support geometry

Positive controls reveal a specific asset-calibration issue. At the fixed 2-mm tolerance, the initial check accepts four red-cup controls but rejects all four white-cup controls. Each control places an upright cup at one of four fixed yaw angles, releases it, and lets simulation settle; these are measurement controls, not policy executions. The white cup's visual mesh extends 2.395 mm below its collision support surface, so floor-supported placement can violate the visual lower-Z boundary. The discrepancy arises because contacts and rendering use different geometric representations.

The floor-calibrated criterion is defined before new data selection or student outcomes. It retains the full visual/collision envelope at the horizontal sides and upper boundary, but uses the collision envelope at the floor, with the same 2 mm tolerance, floor contact, and release requirements. All eight positive controls then pass throughout the final 50 steps. The correction accommodates the support-surface mismatch while continuing to reject side or top protrusions. The original full-visual endpoint remains separately recorded. Both endpoints measure their stated constraints; uprightness, stationary equilibrium, human judgments, and real-world completion require additional evidence.

All 500 available original and repeated Qwen executions pass the 600-step checks of native labels, contacts and positions. We preserve the original full-visual traces and recompute floor-calibrated labels from the saved bounds and contacts. These are replays of existing episodes, rather than 500 additional independent evaluations. The final three compartment-data models average $27.3 \pm 5.0\%$ on calibrated final-50 placement, compared with $22.7 \pm 3.1\%$ under the original full-visual check and 65.3% under native contact-free final-50. Of 105 native-hit episodes, 43 never achieve calibrated full-volume entry and 51 never also satisfy floor contact and release.

The paired final-50 comparison contains 150 compartment-data executions: 41 pass both native `In` and calibrated placement, 57 pass native alone, 52 pass neither, and none passes calibrated placement alone. Thus 57/98 native positives (58.2%) are excluded by geometry. The three checkpoints

Table 29: Floor-calibrated checks on existing Qwen trajectories. Counts are out of 50; triples list original training seeds 0/1/2, with one execution each. All original-data controls remain zero. Table 24 retains the original full-visual results.

Endpoint	Compartment-data models
Native `In` first hit	35,36,34
Support-compatible full-volume entry	23,23,16
Support-compatible supported/released first hit	19,21,14
Support-compatible supported/released final-50	14,16,11

share 50 physical starts. For the selected seed-0 adapter, its original execution and two repeats give calibrated counts of 14,14,10 out of 50, against 13,9,7 under the full-visual endpoint. The asset correction narrows the native/geometric difference without removing it, while compartment supervision improves both geometric endpoints.

Against the matched all-zero controls, exact two-sided tests give $p = 1.22 \times 10^{-4}$, 3.05×10^{-5} and 9.77×10^{-4} for the calibrated contrasts at seeds 0/1/2, respectively. Each test conditions on one trained checkpoint. It does not treat the reused states as independent training replications.

The qualitative examples in Figure 5 are final-Qwen trajectories at step 600, rendered with native OSMesa cameras and the fixed right-bay crop. Selection uses the minimum identity hash within each cup-color and endpoint group after outcomes are known. All 600 native/contact/position and geometric labels are verified against saved traces, and each displayed failure also fails at the shown frame. These selected illustrations describe the mechanism of disagreement rather than its prevalence.

M.2 Audit of the supervision targets

The supervision audit compares 95 retained compartment demonstrations with 95 source-matched original-data controls. It restores each of the final 50 saved pre-action states and runs simulator forward computation before measuring geometry, thereby auditing dataset states rather than reconstructing the controller's post-action history. All 95 compartment demonstrations pass native final-50 `In`, but 36 pass full-visual geometry and 42 pass floor-calibrated geometry (36/46 red, 6/49 white). Native selection therefore admits geometrically weak placements. No original-data demonstration passes compartment placement; attributing student errors to teacher quality requires the separate learning intervention.

We then build a teaching set with stronger placement checks. We retain all 72 action-generated external/compartment attempts, including failures and a fixed-source retry that releases the cup above the compartment. After the initial reset, every trajectory follows robot actions. We check 51 final saved pre-action states per attempt: calibrated geometry for the compartment, and native contact-free `On` for the exterior. Nine sources per color qualify for both goals, yielding 36 shared demonstrations. Each goal pair shares the initial simulator state, model XML and both initial camera images. No exact initial object pose overlaps with the planned confirmation cases, although those cases have appeared in historical audits and are not globally unseen. State restoration and native/gripper checks pass for all 72 attempts. Cached post-step bounds can differ by up to 1.624 mm from bounds recomputed after restoring the saved states. We retain those cached bounds as diagnostics rather than claiming exact action replay.

M.3 Released-policy instruction intervention

We freeze 20 historical official starts, ten per color, and run the released SFT and RIPT checkpoints under both external and compartment text, yielding 80 executions. Within each checkpoint, only the policy-facing instruction changes. We verify both initial camera arrays, all proprioceptive inputs, saved native flattened-state bytes, weights, inference settings and case randomness. We retain native top-five sampling, temperature one, the eight-action horizon, 32-step skill block and 600-action episode. These historical starts share a scene template; they are not new confirmation cases or training replications.

CPU replay reconstructs all 80 episodes from saved float32 actions, reproducing every initial array byte and all 48,000 native flags. A collector bound to the active simulator records geometry and contact, and a frozen auditor independently verifies each reconstructed frame.

A matched 600-step replay gives identical native flattened-state bytes and geometry with camera observables enabled or disabled, permitting camera-free reconstruction. Native state serialization contains time, joint positions, and velocities, but omits controller/actuator state and derived poses/contacts; the original executions also lack per-step pixels and states. Consequently, reconstruction validates the saved channels and new measurements, with no direct comparison to those missing originals. It performs no policy inference and adds no new evaluation cases.

Table 30: Released policies under changed instructions. Each cell counts final-50 successes out of the same 20 starts. Exterior placement uses native `On` without gripper contact; compartment placement uses calibrated geometry, floor contact and release. These different proxies do not independently certify physical completion of both goals.

Checkpoint	Instruction	Exterior native proxy	Calibrated compartment
SFT	External	7	0
SFT	Compartment	4	0
RIPT	External	18	0
RIPT	Compartment	13	0

Neither released checkpoint completes both instructions under the native, mixed, or operational-auxiliary joint endpoints. Each 0/20 rate has Wilson 95% interval [0,16.1]%, reflecting substantial uncertainty about conditional capability. Exterior-goal success under exterior rather than compartment text differs by +15 points for SFT and +25 for RIPT, with gain/loss/tie counts 6/3/11 and 7/2/11 and paired bootstrap intervals [-15,45] and [0,50]. The compartment comparisons contain 20 ties, for which a degenerate bootstrap interval cannot establish an exactly zero effect. The stronger exterior auxiliary passes 7/20 and 17/20 under exterior text, respectively; compartment and joint counts remain zero. These results delimit the tested conditional behavior without supporting an architecture ranking or general language-insensitivity claim.

M.4 Frequency-matched native label intervention

We compare three ways of labeling the same 36 source-paired physical demonstrations, starting from the released QueST model. A uses the correct goal text. B uses compartment text for both behaviors. C independently permutes the correct text within each sampled batch and color. C preserves A's exact sampled prompt frequencies; B does not. Across 640,000 sampled transitions per arm, C mismatches approximately 43.91% of labels, rather than exactly half. For three training seeds, we use native action-token negative log-likelihood, batch 64, learning rate 10^{-4} and the final checkpoint at 10,000 updates. Tensor checks confirm that the 118 action-autoencoder entries remain identical to the released model. Changes are confined to 151 permitted parameter entries and 60 native batch-normalization buffers. Two seed-0 policies reuse their previously trained final checkpoints; the other seven are newly trained.

Before evaluation, we freeze 20 new physical starts produced by native reset, ten per color. We apply source-based validity filters and rule out exact state overlap with official starts and training sources. We independently verify restoration using both camera arrays, proprioception and native flattened-state bytes. The starts vary cup and caddy translations within one scene template; object orientations and the robot configuration stay fixed. They do not test different layouts and are not certified absent from all historical project states. The first stage uses all nine fixed checkpoints, both instructions and one execution seed, for 360 episodes of 600 actions. We retain native top-five sampling, eight-action execution and 32-step skills. Temperature one is reconstructed from the source/configuration, rather than directly recorded as a runtime attribute.

The independent sample consists of 20 physical cases. A–C is the primary comparison; A–B is auxiliary. For each training seed, we report gains/losses/ties and a conservative, nondegenerate 95% paired-difference interval using Bonferroni-adjusted Clopper–Pearson bounds. Means and ranges across seeds are descriptive. Reusing the starts does not yield 60 independent cases.

We initially defer two further execution seeds, totaling 720 episodes. The frozen decision rule calls for inspecting saved trajectories, inputs and randomness records if effects reverse across seeds, depend on one case or seed, or joint-zero results leave target selection and execution variation unresolved. We run the full paired supplement only when those repeats can answer a concrete remaining question. Repetition within the same template cannot establish independent generalization.

All nine checkpoints finish 40 episodes without changes to inference tensor states, and all three supervisors exit normally. The independent CPU auditor checks all 216,000 saved primitive frames, recomputes every summary and verifies the actual initial-input/embedding bytes, action clipping and checkpoint-file identities. This checks the stored bounds and contacts against fixed assets. It does not replay the complete simulator/actions or supply missing per-step camera and proprioception arrays. Native flattened state omits controller, actuator and derived-cache state. Source inspection confirms that this collector is constructed after restoration.

Table 31: First-stage QueST label comparison. Mixed joint counts are out of 20 shared cases per training seed. A uses correct labels, B uses compartment labels for both behaviors, and C permutes labels while preserving their frequencies. Intervals are conservative 95% paired-difference bounds in percentage points for each seed. A–B has the same gains/losses/ties and intervals in this stage.

Training seed	A	B	C	A–C gain/loss/tie	Difference interval (pp)
0	1/20	0/20	0/20	1/0/19	$[-19.6, 27.9]$
1	0/20	0/20	0/20	0/0/20	$[-19.7, 19.7]$
2	0/20	0/20	0/20	0/0/20	$[-19.7, 19.7]$

Only one seed and one case favor A on mixed joint completion: the descriptive mean difference is 1.7 percentage points, with range 0–5. Native joint counts are 1/20 for each A seed and zero for every B/C seed, while the stronger operational-auxiliary joint check passes none. Mixed completion counts for the indicated external/compartment goal are A: (4/4,4/0,3/0), B: (0/0,4/0,4/1) and C: (0/0,3/0,7/0). Under compartment text, wrong exterior-goal attainment is (0,0,0) for A, (1,3,4) for B and (1,3,9) for C. These fixed-seed observations do not establish a reliable benefit of correct labels or the ability to complete both requested goals.

The result activates the prespecified one-case/one-seed trigger. Inspection identifies A0's only mixed positive as white-cup reset seed 2026100204. A1/A2's native joint positives occur at white-cup seed 2026100209: their compartment runs maintain floor contact and no gripper contact throughout the final 50 steps, but never pass calibrated containment. For these cases, the failure concerns the spatial envelope, rather than release or floor contact. A0's exterior run also fails the stronger auxiliary check, limiting its physical interpretation.

The saved actions and source seed/reset calls cannot tell us whether A0's mixed positive repeats under different execution randomness, because runtime RNG states and sequences were not saved. We therefore run both deferred execution seeds for all nine checkpoints: 720 additional episodes, reported separately as an exploratory diagnostic of execution variation. The first-stage outcomes and inferential units remain fixed.

Table 32: Exploratory execution repeats, authorized separately. Each row reuses the same 20 cases; training and execution seeds do not enlarge the independent sample. Counts use the mixed joint check. A–C intervals are conservative 95% paired-difference bounds in percentage points.

Execution	Train	A	B	C	A–C g/l/t	95% CI (pp)
20260918	0	0/20	0/20	0/20	0/0/20	$[-19.7, 19.7]$
20260918	1	0/20	0/20	0/20	0/0/20	$[-19.7, 19.7]$
20260918	2	0/20	0/20	0/20	0/0/20	$[-19.7, 19.7]$
20260919	0	0/20	0/20	0/20	0/0/20	$[-19.7, 19.7]$
20260919	1	0/20	1/20	0/20	0/0/20	$[-19.7, 19.7]$
20260919	2	0/20	0/20	0/20	0/0/20	$[-19.7, 19.7]$

All nine checkpoints finish 80 further episodes with unchanged inference tensor states, and all three supervisors exit normally. The independent CPU auditor recomputes all 432,000 stored primitive

frames and every summary. It verifies initial input/embedding bytes against the first stage and checks all 18 checkpoint–execution groups and actual file identities. Runtime records show temperature 1.0 before and after each worker. They do not supply the missing first-stage record or a measurement for every inference call. We seal the follow-up audit implementation after launch but before inspecting endpoints, using the original frozen measurement and statistical rules.

A and C score 0/20 for every training seed under both added execution seeds (Table 32). A–C gains/losses/ties are always 0/0/20, with conservative 95% bounds [-19.7,19.7] points. B scores 1/20 only under execution 20260919 and training seed 1, at white-cup reset seed 2026100200. That case also passes the native and operational-auxiliary joint checks. Its A–B comparison is 0/1/19, with interval [-27.9,19.6] points; the other A–B rows have 20 ties. These comparisons establish neither superiority nor equivalence, and do not show an exactly zero effect.

Under execution 20260918, native joint counts are A=(2,1,1)/20, B=(0,0,0)/20 and C=(0,0,1)/20 across training seeds. Under 20260919, they are A=(0,1,0)/20, B=(0,1,0)/20 and C=(0,0,0)/20. The stronger operational-auxiliary check passes no A/C group and only B's single positive.

Two cases show why the geometry threshold matters. For A0 at the original white-cup reset seed 2026100204 under execution 20260918, the native joint check remains positive and compartment placement passes the prespecified 5-mm sensitivity check, but fails the primary 2-mm check. C2 at white-cup seed 2026100203 under the same execution also passes native joint and 5 mm, but fails 2 mm. We retain the frozen primary criterion. The only first-stage coherent-label mixed positive therefore fails to repeat under either added execution seed. This answers the local repeat question for these 20 cases and this training budget; it does not identify a unique RNG mechanism or establish general incapability. We do not reconstruct missing RNG states/sequences, full simulator/controller state, or complete per-step camera/proprioception records. The earlier fixed-goal supervision improvement remains a separate result and does not establish conditional reliability here.

M.5 Controlled instruction-label comparison

The two Qwen conditions share the same 36 physical demonstrations, images, actions, source identities, sampled-index sequence, seed-0 adapter and saved AdamW state. The adapter already has 4,200 updates. One condition gives each demonstration its correct external or compartment text; the other gives compartment text to both behaviors. We record tokenized prompts and sampling identities, so only the training labels differ. Each run completes 1,000 further updates, and we evaluate only the final checkpoint. Two earlier matched attempts were stopped after 100 logged updates each because measured runtime exceeded the allocated cap. They have no policy evaluations and provide no selected checkpoints.

We test both models under both instructions on 20 official physical starts, ten per color, for 80 episodes. The 600-step horizon, 128-pixel camera, crop, eight-action execution and per-case randomness stay fixed. Initial simulator-state and first policy-RGB identities match across both texts and models for all 20 starts. Joint success requires contact-free `On` throughout the final 50 steps under external text and the specified compartment endpoint under compartment text from the same start. Exterior success remains a native-predicate proxy.

Table 33: Controlled Qwen label comparison. Each instruction and joint count is out of 20 physical cases. Exact paired tests compare case pairs; frames do not add cases or training replications.

Final-50 endpoint	Aligned labels	Conflicting labels	Exact p
External text: native `On`	12	7	–
Compartment text: native `In`	6	1	–
Compartment text: support-compatible placement	4	0	–
Joint native goals	3	1	0.625
Mixed joint: calibrated compartment	2	0	0.500
Mixed joint: visual compartment	0	0	1.000

Native joint success is 15% with aligned labels and 5% with conflicting labels, with Wilson intervals [5.2,36.0]% and [0.9,23.6]%. The paired difference is 10 percentage points, with paired bootstrap interval [-10,30] points. The calibrated joint difference is also 10 points, supported by only two

discordant positives. Neither joint comparison establishes a label-treatment advantage. Both original full-visual joint rates remain zero. This single-seed test controls the training-label difference within one template, but does not establish reliable completion in both directions or a generally effective learning repair. We keep its weak result separate from the earlier positive fixed-goal supervision comparisons.

M.6 ACTION-REPRESENTATION DIAGNOSTICS

To examine action fidelity, we replay all 36 teacher trajectories from 18 shared starts, either directly or after frozen native action encoding/decoding. Every eight steps, the oracle encodes the teacher's next 32 actions, pads with repeated final actions when needed and executes the first eight decoded actions. The oracle knows the teacher sequence; it does not predict actions from observations or language. Both modes restore the same world state once and follow the same fresh-controller initialization recipe. Historical OSC goals, reference orientation, and interpolator state were not serialized or restored. Some raw-action replays fail their historical qualifications. Reanalysis of the recorded geometry passes.

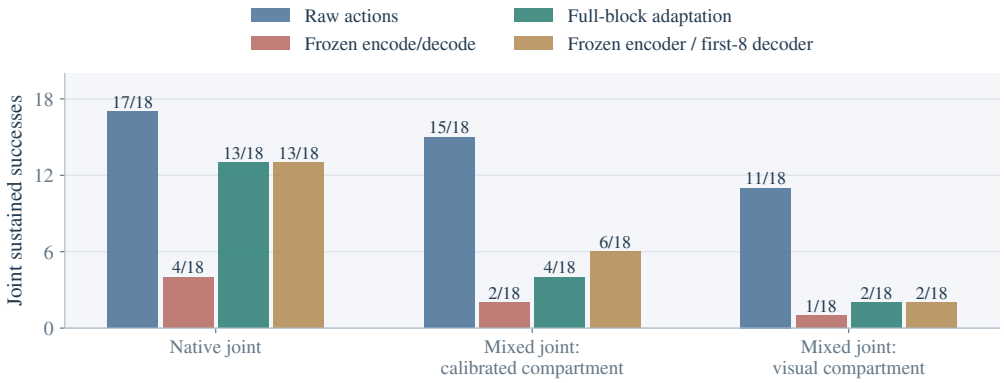

Figure 13: Action-reconstruction diagnostic on 18 shared teacher starts. Joint success requires the contact-free exterior native goal and the specified compartment check throughout the final 50 steps. Both adaptations use 28 training and eight held-out source-paired demonstrations; all failures are included. Teacher oracles replay known actions and are not learned-policy tests or new task templates.

Raw actions produce joint native 17/18, calibrated 15/18 and original-visual 11/18 successes. Frozen encoding/decoding reduces those counts to 4/18,2/18,1/18. Thus the action-compression chain loses control fidelity on these teacher trajectories. This does not identify the sole cause of weak learned behavior. A learned policy may compensate through feedback, so the teacher oracle is not a policy upper bound.

We also train native stage-0 reconstruction from the released checkpoint, fixing 10,000 updates, learning rate 10^{-4} and batch 64. Source-ID hashing holds out two goal-paired sources per color, leaving 28 training and 8 representation-validation demonstrations. These are not policy test cases. Every non-autoencoder tensor remains unchanged. The final-checkpoint oracle achieves 13/18 native, 4/18 calibrated and 2/18 original-visual joint successes; calibrated held-out success is 0/4. It fails the predeclared gate of 10/18 overall and 3/4 held-out successes, so we do not launch the policy follow-up.

In a separate diagnostic, we freeze the encoder, quantizer, observations and prior, and train only the decoder/output head with first-eight-action L1 reconstruction. We retain the initialization, split, learning rate, batch, seed and final selection after 10,000 updates. Exactly 74 tensors change and 264 remain identical. The oracle achieves 13/18 native, 6/18 calibrated and 2/18 original-visual joint successes, with 1/4 calibrated held-out pairs. It also fails the gate. Because both trainable parameters and loss horizon change, we cannot attribute the outcome to either alone. Neither adaptation establishes reliable physical control.

N COMPLETED FREQUENCY-MATCHED LABEL FOLLOW-UPS

The completed follow-ups compare correct and permuted labels under fixed protocols within each architecture. The primary mixed joint check requires native `On` without gripper contact for exterior placement and calibrated 2-mm placement with support and release for the compartment. Each condition must hold throughout the final 50 steps under its requested instruction from the same start. We retain 20 physical cases per checkpoint as the sample unit. Frames, training seeds and repeated executions do not add independent cases.

N.1 FIXED-BUDGET QUEST FOLLOW-UP

Correctly labeled A0 and frequency-matched permuted C0 each continue from their existing 10,000-update checkpoints for 10,000 more updates. Both use the same 36 demonstrations, 640,000 sampled transitions and prompt frequencies; C mismatches 281,028 sampled labels. We test the final checkpoints under both instructions on the 20 frozen starts with execution seed 20260926: 80 episodes and 48,000 saved steps. The extra budget is fixed before outcomes, with no intermediate checkpoint selection.

Table 34: QueST follow-up after 20,000 total updates per arm. Counts are out of 20 shared cases. Exterior and compartment placement use the two different components of the mixed check.

Labels	Exterior	Compartment	Mixed joint
Coherent A0	4/20	1/20	0/20
Permuted C0	3/20	0/20	0/20

The joint comparison has 0/0/20 gains/losses/ties and conservative 95% difference interval [-19.7,19.7] percentage points. It does not meet the frozen expansion rule, so we launch no additional training seeds or execution repeats for this pilot. The strict source-action gate also remains failed. This exploratory budget increase identifies neither a label benefit nor an architecture ceiling.

An independent audit of saved outputs verifies source/checkpoint identities, continuation records, matched samples and embeddings, initial-array dtype/shape/bytes, finite actions/states and every saved summary. Training, evaluation and auditing all exit normally. The audit does not rerun physics or independently reconstruct missing raw-camera/proprioception transformations and full geometric predicates. This study uses one training seed and one execution seed; it is not an equivalence test.

N.2 FREQUENCY-MATCHED QWEN CONFIRMATION

We next compare labels from the released Qwen2.5-OpenVLA baseline, which uses a Qwen2.5-0.5B language backbone and rank-16 LoRA. Correct-label A and permuted-label C share the same 36 physical demonstrations, released base model, initial adapter tensors and sampled transitions. Each of three training seeds receives 600 optimizer updates, with microbatch four, gradient accumulation four, learning rate 2×10^{-4} and 30 warm-up steps. We evaluate the fixed final update for all six models.

C permutes prompts independently within each microbatch and cup color while preserving A's exact prompt multiset. Mismatch fractions are 26.65%, 25.67% and 26.65% across seeds, each based on 9,600 samples. Gradient accumulation does not expand the permutation group. This intervention therefore changes a smaller fraction of labels than QueST's batch-64 permutation. Native 128-pixel RGB, vertically flipped 224-pixel inference crops and eight-action execution match the training/evaluation setup. Each episode includes ten zero-action settling steps followed by 600 actions.

All six checkpoints run both instructions on the same 20 frozen starts under execution seed 20260920, totaling 240 episodes. The saved-output audit verifies matched initialization, samples, prompt frequencies, checkpoint identities and registered frozen parameters. It checks canonical raw observations/states, initial bytes under both instructions, complete traces and all 75 saved native crops per execution. The unregistered VQ action-tokenizer tensors are absent from the before/after full-training fingerprints. Asset hashes, evaluation mode, no-gradient encoding and optimizer exclusion support their fixed status at the source level, but do not retrospectively certify their tensor states.

Table 35: Qwen confirmation. A uses correct labels; C permutes them while preserving their frequencies. Counts are out of 20 shared cases. Gains/losses/ties and conservative 95% intervals compare A–C mixed joint completion within each training seed.

	A indicated		C indicated		Mixed joint			
Train	Exterior	Comp.	Exterior	Comp.	A	C	g/l/t	95% CI (pp)
0	10/20	1/20	6/20	2/20	0/20	1/20	0/1/19	$[-27.9, 19.6]$
1	16/20	2/20	9/20	0/20	2/20	0/20	2/0/18	$[-18.8, 34.9]$
2	11/20	1/20	12/20	0/20	1/20	0/20	1/0/19	$[-19.6, 27.9]$

The mixed joint differences across training seeds are $-5, +10, +5$ percentage points, and every interval includes zero. Native joint counts are (0,3,4)/20 for A and (1,1,1)/20 for C. Geometric qualification therefore changes which native successes count as completed placement. This single-execution study within one template establishes neither a stable advantage of correct labels nor a zero effect.

A primary joint positive activates the frozen follow-up for all six models under execution seeds 20260921 and 20260922. At the cutoff for this study, that 480-episode exploratory repeat is paused and incomplete. We retain the primary outcomes and do not analyze or pool partial repeat outputs. Appendix P decomposes the completed primary endpoints without adding executions.

N.3 Native action-roundtrip control

We separately test Qwen action encoding/decoding without learned-policy inference. A fixed eight-trajectory pilot activates evaluation of all 36 source demonstrations, giving 44 diagnostic executions but only 36 unique sources. Roundtrip actions achieve 5/18 exterior and 4/18 compartment target positives; the extra 50-step holds preserve those counts. Exterior qualification uses the stronger operational proxy, rather than native `On` alone.

Two cases that succeeded under source-history-matched raw actions fail after the roundtrip: red `demo_0` for exterior placement and red `demo_1` for the compartment. The latter still passes native `In` but fails calibrated geometry. These counterexamples show loss of execution fidelity in the roundtrip chain. They do not identify a unique VQ mechanism or explain every learned-policy failure.

The saved-array audit checks actions/states, normalization, padding, gripper processing, clipping, trace errors and source/input identities. It correctly rejects a deliberately modified action copy. Forty VQ weights/buffers have matching fingerprints before and after the diagnostic. Endpoint and hold checks recompute saved primitive summaries; they do not rerun geometry or physics. This diagnostic tensor check does not fill the missing full-training fingerprints, establish a label effect or demonstrate held-out reliability.

O Operational definitions and artifact verification

Table 36 offers four reproduction entry points, from inexpensive numerical checks to full runs. The uniform 240-rollout upload includes every case and the quantities needed to recompute endpoints. It omits redundant raw actions, per-call inputs and large per-step flattened states; hashes for the original full records and portable records remain separate. Full weights and licensed meshes must be reconstructed. We do not claim anonymous external hosting until access has been checked.

The four earlier caddy checks retain their v15 scope. Later table recomputation, source checks and six-episode replay each have their own documented scope. `REPRODUCE_V19.md` lists the new direct-recording and native-call entries and dependencies. Replay under an alternate asset prefix still uses the existing environment and linked original assets. Historical final-Qwen rotations remain reconstructed; geometry for the historical baseline50 run remains unavailable. Full historical teacher HDF5 and policy weights must be reconstructed separately. We have not verified native reconstruction on a clean machine, anonymous hosting or reviewer access without signing in.

SIMPLER motion and grasp. SIMPLER's frozen `PutOnInSceneEnv.evaluate` measures source displacement as $d_s = \|p^{xy}_{s,t} - p^{xy}_{s,0}\|_2$. The baseline $p_{s,0}$ is saved after reset settling,

Table 36: Four ways to reproduce the evidence, from table recomputation to full training. The review supplement indexes the inputs and commands. Passing a numerical or source check does not show that models, assets or the full study can be reconstructed.

Layer	Entry and dependencies	Input ⟶ output	What has been checked
Numerical recomputation	`python run_all.py`; `python run_revision.py`. Python ≥3.10 standard library; no GPU or simulator.	Saved predicates, geometry and contact bounds, and selected record fields, all checked against the manifest ⟶ caddy/MINERVA/support tables, completion analyses, SIMPLER paired tables and the frozen candidate index in `outputs/`.	Both entries passed the v17 checks after clean extraction. The later `run_v19.py` check recomputes the new direct-recording and native-call results from saved records. These commands do not regenerate missing observations or dynamics.
Source-method query tests	`python run_all.py --only robocasa`. The portable core includes a limited NumPy substitute for these fixed test fixtures; native scripts require NumPy.	Pinned RoboCasa methods extracted as syntax trees and specified mocked sensor values, queried 1/2/4 times per transition ⟶ success labels, counterexamples and comparisons of monitor state. The three source checks are `audit_queries.py`, `check_remaining_counters.py`, `verify_query_state.py`.	The archive includes completed source-method checks and earlier checks in independent processes. They do not test robot physics, estimate how often physical trajectories are affected, or verify a patch deployed in the native environment.
Selected physical replay	Earlier checks use `python run_checks.py`; the new six-case check uses `replay_portable.py`, documented in `REPRODUCE_V19.md`. Both require pinned LIBERO/robosuite/MuJoCo, Torch, NumPy, SciPy, official assets and `LIBERO_CONFIG_PATH`.	Four earlier caddy sequences and six new 600-action cases ⟶ checks of reset and settled state, comparisons of saved measurements, and run records. Separately, `revision/replay_simpler_motion.py` replays four SIMPLER sequences using its pinned runtime with CUDA rendering.	The earlier caddy check matched 2,400 steps; the SIMPLER check covered 600 saved actions with reset and label checks. Six new caddy episodes then passed replay, including six checks under an alternate asset prefix, in the existing native environment (Appendix S.2). Replaying these episodes adds no independent policy cases.
Full evaluation and training	The README and pinned protocols describe reconstruction. Qwen confirmation uses `run.sh experiment.py train` or `evaluate`; teacher collection uses `revision/teacher-recipe/collect.py`.	Reconstructed experiment directories, asset paths, full upstream weights and tokenizers, licensed meshes and original HDF5 ⟶ trained weights, policy trajectories and training/evaluation logs. Source and checkpoint hashes, together with earlier run records, identify the intended experiment.	Complete model and asset reconstruction has not been independently verified. The compact archive cannot run the full study by itself. Earlier successful jobs do not establish that this layer has been reproduced.

rather than at the first policy action. The flag `moved_correct_obj` is $(d_s > 0.03\,\mathrm{m}) \wedge (\forall j \neq s : d_j < d_s)$: the source must move more than the threshold and farther than every other object; ties fail. Evaluation recomputes these flags from current displacement and does not accumulate them. `MoveNearInSceneEnv._initialize_actors` sets the baseline `episode_obj_xyzs_after_settle` after settling and unlocking motion.

The current flag `is_src_obj_grasped` calls `WidowX.check_grasp`. Its defaults are a minimum impulse of 10^{-6} and a maximum angle of $60°$. Both finger–object net contact impulses must

meet the magnitude threshold and lie within that angle of the signed finger-opening directions. There is no gripper-width threshold.

Each evaluation with a current grasp increments `consecutive_grasp`; a non-grasp clears it. The diagnostic becomes true after five evaluations. Separate episode-statistic flags accumulate grasp by logical OR. They differ from the current grasp value used in q_t. `reset` clears the consecutive counter, and `_initialize_episode_stats` clears the flags. The artifact's `DEFINITIONS.json` pins these three source files and function locations with SHA-256.

Completion monitor boundaries. SIMPLER's strict collector clears its stability streak at episode reset. After each completed control transition, it reads `evaluate` once and computes q_t. A true result increments the streak; a false result clears it. The collector accepts after five consecutive control states at 5 Hz. The accumulated episode grasp flag and native consecutive-grasp diagnostic are not substitutes for the current no-qualifying-two-finger-grasp condition. Reading the source evaluator itself advances grasp history, so this protocol adds no extra reads.

For LIBERO, a new collector is constructed after each reset/restoration and settling, then samples labels after every executed action. First hit means the first true post-action state in steps 1–600; final-50 checks all states 551–600. No-op holds are reported separately. Predicate and contact lists start empty for each episode. The geometry collector binds its model/data, body/geometry indices and asset envelopes to the current simulator. A hard reset replaces that simulator and therefore requires a new collector.

Table 37: Simulated duration of persistence checks. Frequency denotes the control frequency, rather than inference speed or video frame rate. For n sampled states, the time from first to last is $(n-1)/f$; k additional transitions take k/f.

Endpoint	Frequency	Window and simulated duration
MINERVA first-hit persistence@10	20 Hz	First-hit state plus ten follow-ups; 0.50 s from hit to final follow-up
Caddy student final-50	20 Hz	Post-action states 551–600; 2.45-s sample span, 2.50-s action-window duration
Caddy teacher suffix-50	20 Hz	Fifty saved pre-action states; 2.45-s sample span, shifted one transition from student sampling
Caddy no-op hold-50	20 Hz	Fifty additional actions; 2.50 s after the policy/placement endpoint
SIMPLER strict five-state window	5 Hz	Five consecutive control states; 0.80-s sample span, 1.00-s action-window duration

Native LIBERO defaults to 20 Hz, and the caddy evaluators retain that setting. The captured MINERVA environment also defaults to 20 Hz and forwards it to LIBERO. SIMPLER follows the fixed 5-Hz protocol. The relevant source locations are in `revision/timing-source` and `revision/contract-source`. Final Qwen replans every eight actions, or 0.40 s at 20 Hz. This affects how often the model is queried, while predicates are still sampled after every action. The RoboCasa mocked-sensor harness uses discrete transitions without assigning them a measured physical duration.

The RoboCasa source tests use a monitor protocol; they do not patch every native environment. Each case starts with zero counters, false completion flags and empty position histories (`audit_queries.instantiate`; the remaining-counter initializer keeps its declared helper histories). After physical/sensor transition t, the harness advances an external clock once if present, calls the source success method once and saves the label and full monitor state. Additional queries at t return this saved result without calling the method. A new episode creates a fresh monitor and clears the cache; a reset-time read does not consume a transition. `audit_queries.run_case` defines this boundary. `verify_query_state.equal_reference` compares all scalar, list and array histories under 1/2/4 queries.

Four-trajectory independent-process verification. Before verification, we reuse the frozen identity-hash rule to select one calibrated pass and one failure for each cup color (`geometry-examples-plan.json`, fixed 17 September). The identities are red: corrected seed 2/index 39 and seed 0/index 43; white: corrected seed 2/index 47 and seed 1/index 48. The arti-

fact runs `audit_four.py` in a separate CPU process with the original saved 600-action sequences and pinned official initialization materials. At each step it recomputes native, contact, full-envelope and calibrated labels, and compares bounds/positions within 10^{-6} m.

After each replay, separate probes check simulator replacement under hard reset and whether the collector binds the current simulator. Placing these probes afterward keeps them from changing the replay world. The receipt includes complete logs, world XML, reset/settled states, recorded primitives and a per-field difference list. We also copy the RoboCasa source suite to a fresh temporary directory and run its checks in independent processes. Neither policy inference nor training is used.

All 2,400 replayed steps match the original native, contact and geometry labels. Cup-position error is zero and the difference list is empty. Both collectors bind the replacement simulator in every hard-reset probe. An initial verification attempt placed an extra reset before replay, which changed static fixture XML. Moving the probe after replay restores the frozen single-reset protocol. We retain the failed attempt and fixture-pose differences; no original result changes. All three independent RoboCasa checks exit normally.

P EXISTING-DATA STRATIFICATION AND ENDPOINT DECOMPOSITION

P.1 CUP-SPECIFIC GEOMETRY AND TOLERANCE SENSITIVITY

Table 38: Historical final-Qwen geometry results by training seed and cup color. Each checkpoint uses the same 25 starts per color. CF native final-50 requires the native predicate and no gripper contact throughout the final 50 steps. The tolerance columns apply floor-calibrated geometry with support and no gripper contact over the same window. The primary tolerance is 2 mm; the other columns show sensitivity. Geometry is reconstructed from actions (Appendix L).

Arm	Seed	Cup	Native first-hit	CF native final-50	0 mm	1 mm	2 mm	5 mm
control	0	red	0/25	0/25	0/25	0/25	0/25	0/25
control	0	white	0/25	0/25	0/25	0/25	0/25	0/25
control	1	red	0/25	0/25	0/25	0/25	0/25	0/25
control	1	white	0/25	0/25	0/25	0/25	0/25	0/25
control	2	red	0/25	0/25	0/25	0/25	0/25	0/25
control	2	white	0/25	0/25	0/25	0/25	0/25	0/25
corrected	0	red	15/25	15/25	0/25	9/25	11/25	15/25
corrected	0	white	20/25	17/25	0/25	3/25	3/25	7/25
corrected	1	red	15/25	15/25	0/25	8/25	9/25	15/25
corrected	1	white	21/25	21/25	0/25	5/25	7/25	10/25
corrected	2	red	14/25	14/25	0/25	6/25	9/25	12/25
corrected	2	white	20/25	16/25	0/25	2/25	2/25	4/25

At the fixed 2-mm tolerance, red-cup counts are 11,9,9 out of 25 (36–44%), and white-cup counts are 3,7,2 out of 25 (8–28%). Both tasks improve over their matched all-zero controls. Red cups account for 29 of the 41 calibrated successes across the three checkpoint executions. That is a descriptive count, not 75 independent starts per color.

Tolerance changes the magnitude of the placement effect: seed-wise totals are 12,13,8 at 1 mm and 22,25,16 at 5 mm, with zero at exact zero margin. Compartment supervision improves the measured placement outcome in every task and seed at 1/2/5 mm. The analysis retains the prespecified 2-mm primary criterion, separating sensitivity to tolerance from selection of a favorable threshold.

P.2 TARGET ATTAINMENT VERSUS SUSTAINED COMPLETION

We explore where failures occur using only the completed Qwen confirmation under execution seed 20260920. Correct-target first hit uses native `On` under external text and native `In` under compartment text. Wrong-target hit uses the opposite predicate. Both can occur during one execution; `label_endpoints.json` records their first-step indices for each case. Qualified hits and final-50 completion require contact-free native `On` outside and calibrated supported placement inside. A native compartment hit can therefore fail geometric qualification.

Table 39: How the 240-episode Qwen label experiment reaches or fails each completion check. A uses coherent labels; C uses frequency-matched permuted labels. Each seed and instruction uses the same 20 starts. Joint counts starts that succeed under both instructions; the count appears in both instruction rows for convenience and is counted only once.

Arm	Seed	Instruction	Correct hit	Wrong hit	Qualified hit	Final-50	Hit, not final-50	Joint
A	0	external	16/20	0/20	16/20	10/20	6/20	0/20
A	0	compartment	2/20	3/20	1/20	1/20	1/20	0/20
A	1	external	17/20	0/20	17/20	16/20	1/20	2/20
A	1	compartment	3/20	0/20	3/20	2/20	1/20	2/20
A	2	external	12/20	0/20	12/20	11/20	1/20	1/20
A	2	compartment	6/20	2/20	2/20	1/20	5/20	1/20
C	0	external	11/20	3/20	11/20	6/20	5/20	1/20
C	0	compartment	6/20	4/20	5/20	2/20	4/20	1/20
C	1	external	13/20	3/20	13/20	9/20	4/20	0/20
C	1	compartment	2/20	8/20	0/20	0/20	2/20	0/20
C	2	external	14/20	0/20	14/20	12/20	2/20	0/20
C	2	compartment	4/20	7/20	1/20	0/20	4/20	0/20

Table 40: Paired zucchini strict-success differences. Gain/loss/tie counts use the original and revised instructions on the same 40 resets; differences and Tango 95% intervals are in percentage points.

Policy setting	Gain/loss/tie	Difference	95% interval	Exact p
Base π_0	15/0/25	37.5	[24.2, 53.0]	6.10×10^{-5}
Reconstructed CoVer	9/4/27	12.5	[-5.6, 30.3]	0.267
Rephrase-finetuned π_0	13/8/19	12.5	[-10.2, 33.8]	0.383

Endpoint decomposition distinguishes target attainment from qualified, sustained completion. For coherent-label A, exterior hit-to-completion losses are 6,1,1 out of 20 across seeds, and compartment losses are 1,1,5. For permuted-label C, the corresponding losses are 5,4,2 and 4,2,4. A has more exterior first hits in seeds 0/1 but fewer in seed 2, while compartment first hits show no consistent advantage. In A2, six compartment hits narrow to two qualified hits and one final-50 completion, locating losses at both geometric qualification and persistence.

Joint counts remain A=(0,2,1) and C=(1,0,0), each from the same 20 paired cases. The decomposition shows target-attainment differences hidden by the joint check, but does not establish a stable label benefit. Hitting a target remains a behavioral proxy rather than an independent test of language comprehension. We reuse cases across seeds and introduce no pooled significance test or larger independent n.

P.3 Paired instruction-effect intervals

For the three frozen 40-pair zucchini comparisons, we report approximate 95% Tango paired-binomial score intervals for the corrected-minus-original strict-success probability difference (Tango, 1998, equations (24)–(26) and (33)). These intervals invert the score test with the discordance probability estimated under each hypothesized difference; concordant pairs and zero off-diagonal cells are retained. The earlier normal intervals already used paired variances; the score intervals refine the finite-sample interval calculation. Exact McNemar tests remain unchanged, and the intervals are marginal rather than multiplicity-adjusted.

Q Caddy task implementation and evaluation protocols

We use LIBERO commit `8f1084e3132a39270c3a13ebe37270a43ece2a01`. The executable-contract definition and caddy instance are given in Section 3. To check which instruction reaches the policy, `revision/language-routing/verify.py` runs the captured filename parser, checks the actual QueST and Qwen callers, and asserts the exact strings for both colors. The snapshot includes complete BDDL files and the benchmark factory. The native success and reward code is in `revision/contract-source`: `base_predicates.py` (`In`, lines 57–59; `On`, 62–76), `site_object.py` (`in_box`, 36–56), `bddl_base_domain.py` (`reward`, 165–189), and

`collect_demonstration.py` (hold counter, 87–94). These files match the upstream revision, with no tracked differences. `revision/CONTRACT_SOURCES.json` lists SHA-256 hashes and function locations. The unscaled reward is one when the selected predicate is true and zero otherwise. Optional reward scaling changes its value but not which states receive reward.

Table 41: Protocols for the main caddy comparisons. E is the exact exterior instruction produced from the filename; C is the BDDL compartment instruction defined in Section 3. RGB refers to native resolution before released preprocessing. Learned policies run for 600 actions without stopping at first success; Qwen also uses ten settling actions. Execution counts measure work performed. Reusing starts across seeds or repeats does not add independent starts.

Contrast	Actual text	Data / updates per arm	Controller / starts	Seeds / executions	Endpoint / record
Released QueST / Qwen	E	Public checkpoints; no training	Released QueST / Qwen native; 100 official starts	One checkpoint each; 300 executions	Both native first-hits; direct labels
Reward-only / aligned	E / C	No new demos; 15 updates × 16 rollouts; RLOO group 8	Released QueST; legacy reset	Training seed 0; 240 per arm	Reward support and tensors; direct logs
Matched QueST	E	18 source-matched demos; 600	Native skill controller; indices 10–29, 40 starts	Train 0/1/2; 240 original executions	Native first-hit; direct labels
Initial Qwen	E	18 demos; 600	Initial released action interface; indices 30–49, 40 starts	Train 0/1/2; 280 incl. base	Native first-hit; direct labels
Expanded Qwen	C	95 demos; 1,200 total	128 RGB, eight actions/replan; indices 10–29, 40	Train 0/1/2; 280 incl. base	Native first-hit; saved actions/positions
Final Qwen	C	95 demos; 4,200 total	Same 128/eight controller; 5–9 and 30–49, 50	Train 0/1/2; 350 incl. base	Native first-hit / contact-free final-50; geometry reconstructed for 300
Early direct geometry	C for Qwen; E for QueST	Fixed final seed-0 pair; zero updates	Qwen: same 50; QueST: same 40	Two extra Qwen executions/arm (200); one QueST/arm (80)	Direct full-visual traces; Qwen calibrated labels later reconstructed
New direct calibration	C	Three frozen final adapted Qwen models; zero updates	Same 128/eight controller; 25 red + 25 white shared starts	Train 0/1/2; 150 new executions; no original-data controls	Same-run native/calibrated final-50; direct pose/bounds/contact (Appendix S.1)
Qwen label confirmation	Paired E and C	36 physical demos; 600 per A/C arm	Native 128/eight; 20 translated physical starts	Train 0/1/2; 240	Mixed joint and decomposed endpoints; direct primitives
Fixed-target skill diagnostic	C only	18 compartment demos from the same 36 pool; 600 / 9,600 sampled windows	Same initialization, crop/controller; same 20 starts	Train seed 0; 20 executions, disjoint 7/7/6 workers	Native/qualified hit; calibrated final-50; direct primitives (Appendix T.4)

Several choices change between the 18-demo/600-update and 95-demo/4,200-update Qwen stages: instruction routing, data coverage, image crop and controller. Their difference cannot isolate training budget. Within each stage, the compartment and original-data arms use the same source IDs and update budgets. We select the final checkpoint on a separate ten-case development block before seeing its confirmation results. The overlap check distinguishes teacher, development, earlier confirmation and final confirmation starts. Final training seeds reuse 50 starts, rather than providing 150 independent starts. Appendix N lists initialization, sampling frequencies, optimizer settings and execution seeds for both architectures' label comparisons. Their released batch sizes differ, so the permutations also differ in strength.

Teacher actions and saved-state timing. The portable script `revision/teacher-recipe/collect.py` runs the same action-only teacher used in the experiments. It restores the official demonstration XML and first saved state, then reuses the first 80 red or 75 white grasp actions. The teacher lifts, transfers and lowers the cup toward $(x_s, y_s, z_r + 0.20)$, $r + (0, 0, 0.20)$ and $r + (0, 0, 0.01)$ metres, respectively; r is the native containment-site center. These phases allow 50/80/80 actions. Translation commands are `clip((goal-cup)/0.05,-0.35,0.35)`, rotation commands are zero, and the gripper stays closed (+1). Each phase ends after an action if the error measured before that action is below 8 mm. The formula scales a position error into a command; it does not guarantee 5 cm of cup motion. The native controller and contacts determine actual motion. The teacher then sends 15 release actions,

20 upward retreat actions with z-command 0.3, and 50 open-gripper hold actions with zero motion (gripper -1). A demonstration is accepted when native `In` is true at all final 50 post-action checks. The 95 accepted demonstrations were not filtered by the geometric criterion.

Measurement timing distinguishes teacher-state auditing from student evaluation. HDF5 stores states and observations before each action, but reward and predicate entries describe the following state; the two readings are therefore one transition apart. The 42/95 teacher count audits saved suffixes against calibrated geometry, support, and release, and was not a collection filter. The matched original-data arm uses the same 46 red and 49 white source identities. Dataset hashes, source IDs, and failed collection attempts remain available in the record index.

How geometry was recorded and reconstructed. The original final-Qwen runs saved actions, mug positions, native labels and gripper contact, but did not save orientation. Geometry for the 300 trained-model runs is therefore reconstructed by replaying saved actions. We accept a reconstruction only when every recorded position, native label and gripper flag matches; the original rotations still have no direct check. In separate tests, 200 extra Qwen executions and 80 QueST executions record bounds and contacts during execution using fixed seed-0 models. For the extra Qwen compartment model, the full-visual final-50 counts are 9/50 and 7/50, compared with contact-free native counts of 27/50 and 32/50. QueST gives 11/40 versus 24/40. Matched controls give zero. The extra Qwen floor-calibrated counts, 14/50 and 10/50, come from a later saved-action support audit; they do not directly observe the original rotations.

The later confirmation runs all three frozen adapted models and directly records native labels, rotations, visual bounds, collision bottom and floor/gripper contacts in 150 new executions (Table 45). Each seed uses the same 50 starts. Native and calibrated rates are compared on these same new executions; no original-data control is rerun. Independent replays of six preselected episodes check the new recording process. They do not recover rotations missing from historical runs.

Table 36 lists executable entry points and asset requirements. Geometry remains unavailable for the old 50-case unadapted final-Qwen baseline because it lacks actions and positions. We exclude the paused, partially completed 480-execution label extension.

R SIMPLER EXCLUSIONS AND SENSITIVITY TO MOVEMENT RANKING

How the tasks and starts were selected. The seven tasks and 140 starts were already used to screen the released CoVer implementation, its fixed-reference reconstruction and candidate-0 selection. The earlier protocol prohibited later changes to selector rules or parameters and used results to decide whether to expand evaluation. Its original freeze timestamp was not independently retained. We froze the strict endpoint on 13 September 2026 at 18:40 UTC+8, after inspecting positional outcomes and grasp defects but before running strict evaluation. The confirmation thus reuses inspected cases. It supports a comparison within this corpus, rather than a test independent of task and method selection.

Table 42: Why 37 trajectories that hit the native target failed the frozen strict criterion. We inspect the five-state window beginning at the first native hit. Motion combines the 3-cm and displacement-ranking checks; persistence requires the target relation to continue, with incomplete windows at the execution limit counted as failures. Grasp means a detected qualifying two-finger grasp. Rows separate the observed combinations of failed checks; they do not identify causes.

Failure pattern	Base	Released	Reconstructed	Total
motion	2	0	0	2
motion + persistence	1	1	1	3
motion + grasp	1	0	0	1
persistence	3	4	4	11
grasp	0	8	3	11
grasp + persistence	3	5	1	9
Total	10	18	9	37

Among the 37 excluded trajectories, the five-state window beginning at the first native hit fails the motion condition in 6 cases, the no-qualifying-two-finger-grasp condition in 21 and positional continuity in 23. These conditions overlap, so their counts cannot be added. Table 42 gives the mutually exclusive combinations by policy. A failure in this first window does not explain all later failures in the trajectory. The JSON record lists each excluded case, first-hit index, observed window length, failed conditions and whether a later window passes each relaxed criterion.

The frozen motion check requires both $d_s > 3$ cm and $d_s > d_j$ for every other object. The second requirement makes the source move farther than every distractor, although the instruction does not ask for this ranking. We keep it in the frozen result and remove only the ranking in a sensitivity analysis. That analysis retains absolute source motion, the target relation, no detected qualifying two-finger grasp and five consecutive states. The resulting event lies between the frozen-strict and no-motion events. When these two labels agree, they determine the absolute-only label without displacement data. Four cases have unequal labels and need reconstruction: base tennis-ball seeds 1010/1015/1016 and released zucchini seed 1001. A separate native process restores the exact setup and reset, then executes each saved 150-action trace. Reset image, state and proprioception hashes match, as do all recorded motion, grasp, relation and strict labels. We then record source and distractor XY displacement. These displacements come from replay, not from the original execution.

Removing the ranking requirement adds 3 base and 1 released successes, giving 43/62/65 successes per 140. Released CoVer versus base has gain/loss/tie counts of 32/13/95 and a +13.57 pp difference (exact $p = 0.006609$); reconstructed CoVer versus base has 36/14/90 and +15.71 pp ($p = 0.002602$). The frozen-strict counts are 35/14/91 and 38/13/89. Removing the entire motion condition also gives 43/62/65 on this corpus, but tests a different event. Removing only the no-grasp check gives 40/69/68; requiring the complete conjunction for one state gives 44/71/74. These are post-hoc sensitivity analyses with unadjusted exact paired tests. We report both the frozen result and the result without a ranking requirement that the instruction does not specify.

Ranking-removal analysis reuses recorded outcomes, adding zero training or policy inference. Its four SIMPLER replays execute 600 saved actions within the original 140 paired task–state sample. Likewise, the earlier four-case caddy check replays 2,400 actions. These executions verify existing comparisons rather than enlarging their case denominators.

Table 43: SIMPLER results by task after removing displacement ranking. Each task has 20 matched starts; R/C denote released/reconstructed CoVer. A gain is success by CoVer alone, a loss is success by base alone, and a tie is agreement. These rows split the same 140 pairs by task; no additional task-wise significance tests are performed.

Task	Pairs	Base	R	C	R gain/loss/tie	C gain/loss/tie
Carrot on plate	20	10	7	7	2/5/13	2/5/13
Eggplant in basket	20	14	17	18	4/1/15	5/1/14
Red Bull on plate	20	3	9	12	8/2/10	9/0/11
Spoon on towel	20	2	4	7	3/1/16	6/1/13
Stack cube	20	0	3	4	3/0/17	4/0/16
Tennis ball in basket	20	8	17	16	10/1/9	10/2/8
Zucchini (plate target)	20	6	5	1	2/3/15	0/5/15
Total	140	43	62	65	32/13/95	36/14/90

CoVer's aggregate advantage is distributed unevenly across tasks. Both variants have positive differences on five of seven tasks under both endpoints, with negative differences on carrot and zucchini. After removing movement ranking, tennis-ball contributes +9/+8 and Red Bull +6/+9 successes for released/reconstructed CoVer, within total differences of +19/+22. The complete corpus therefore supports an aggregate benefit alongside task-specific regressions.

Table 44: SIMPLER results by task under the frozen criterion, including displacement ranking. Each task has 20 matched starts; R/C denote released/reconstructed CoVer. A gain is success by CoVer alone, a loss is success by base alone, and a tie is agreement. These rows split the same 140 pairs by task; no additional task-wise significance tests are performed.

Task	Pairs	Base	R	C	R gain/loss/tie	C gain/loss/tie
Carrot on plate	20	10	7	7	2/5/13	2/5/13
Eggplant in basket	20	14	17	18	4/1/15	5/1/14
Red Bull on plate	20	3	9	12	8/2/10	9/0/11
Spoon on towel	20	2	4	7	3/1/16	6/1/13
Stack cube	20	0	3	4	3/0/17	4/0/16
Tennis ball in basket	20	5	17	16	13/1/6	12/1/7
Zucchini (plate target)	20	6	4	1	2/4/14	0/5/15
Total	140	40	61	65	35/14/91	38/13/89

S Direct Geometry and Native Monitor Validation

S.1 Directly recorded placement across three training seeds

We run each of the three frozen adapted Qwen checkpoints from the same 25 red and 25 white starts. This gives 150 executions from three training seeds, but only 50 distinct starts. We report seeds separately and do not treat the 150 executions as iid samples. There is no new original-control arm or optimizer update. The 30 preselected pilot records remain part of the 150. Expansion depends on checks of recorded fields, resets, recomputed labels and replay, rather than success rate, significance or agreement with historical results.

After each `env.step`, the collector reads native labels, contacts and geometry at the same boundary, without an extra `sim.forward`. Each hard reset attaches a new collector to the current simulator. We save rotations, visual bounds, collision bottom and floor/gripper contacts. Replay inputs include the world, initial and settled physics/controller states, actions and RNG. Recomputing endpoints from these records passes for all 150 executions, as do hashes for episode identity, world, initial state and RNG. Only the simulator reference is omitted from the controller snapshot.

Table 45: Directly recorded Qwen placement results and tolerance sensitivity; all entries are counts. Hit means native `In` is true at least once over 600 actions. N requires native `In` and no gripper contact at every final-50 action. $V@2$ checks the full visual envelope, support and absence of gripper contact over the final-50 window at 2 mm. $G@t$ uses the floor-calibrated envelope at t mm over final-50 actions, with the same support and contact requirements. Both combines a seed's 25 red and 25 white starts.

Seed	Cup	n	Hit	N	$V@2$	$G@2$	$G@0$	$G@1$	$G@5$
0	Red	25	15	15	11	11	0	9	15
0	White	25	20	17	2	3	0	3	7
0	Both	50	35	32	13	14	0	12	22
1	Red	25	15	15	9	9	0	8	15
1	White	25	21	21	2	7	0	5	10
1	Both	50	36	36	11	16	0	13	25
2	Red	25	14	14	9	9	0	6	12
2	White	25	20	16	1	2	0	2	4
2	Both	50	34	30	10	11	0	8	16

The native and calibrated counts, $N/G@2$, are 32/14, 36/16 and 30/11 out of 50 on the same runs. Their averages across seeds are 65.3%/27.3%, with gaps of 36/40/38 percentage points. Both colors show a gap in every seed, though its size and uncertainty differ. The full-visual counts are 13/11/10. Floor calibration mainly changes which white-cup placements pass. Results at 0/1/2/5 mm show how this proxy depends on tolerance; they do not establish a universal physical tolerance.

Table 46: Native-minus-calibrated differences on paired new executions. $00/01/10/11$ encode $(N, G@2)$; gain/loss/tie mean N-only, G-only and agreement. Δ and paired Tango score 95% intervals are in percentage points. Intervals and exact two-sided McNemar p values are exploratory, marginal and unadjusted. Comparisons use 50 pairs per seed or 25 per seed/color cell. We neither use independent-proportion intervals nor pool the 150 executions as iid samples.

Seed	Cup	n	00	01	10	11	gain/loss/tie	Δ	Tango 95% CI	Exact p
0	Red	25	10	0	4	11	4/0/21	16.0	[0.5, 34.7]	0.125
0	White	25	8	0	14	3	14/0/11	56.0	[35.2, 73.3]	1.22×10^{-4}
0	Both	50	18	0	18	14	18/0/32	36.0	[24.1, 49.9]	7.63×10^{-6}
1	Red	25	10	0	6	9	6/0/19	24.0	[7.5, 43.4]	0.031
1	White	25	4	0	14	7	14/0/11	56.0	[35.2, 73.3]	1.22×10^{-4}
1	Both	50	14	0	20	16	20/0/30	40.0	[27.6, 53.8]	1.91×10^{-6}
2	Red	25	11	0	5	9	5/0/20	20.0	[4.0, 39.1]	0.062
2	White	25	9	0	14	2	14/0/11	56.0	[35.2, 73.3]	1.22×10^{-4}
2	Both	50	20	0	19	11	19/0/31	38.0	[25.9, 51.8]	3.81×10^{-6}

S.2 Replaying the same episodes

Saved-action replay tests the accuracy of direct recording independently of new policy inference. Six preselected episodes, index 5 in each seed/color cell, replay their 600 actions in separate native processes. World XML and initial/settled controller states match exactly, with zero maximum error in positions, bounds, collision bottoms, qpos/qvel, and initialization arrays. The largest computed SO(3) difference is 6.39×10^{-6} degrees from roundoff; all contact, predicate, and geometry labels match. Frozen tolerances are 10^{-6} m and 10^{-4} degrees, with exact label/contact agreement. Rotation-matrix comparison handles the equivalence of opposite quaternion signs.

Two initial restoration attempts fail before any policy action because settled qpos differs by $2.98 \times 10^{-5} > 10^{-6}$. A replay-only amendment then uses the collector's native construction, reset, initialization, and settling sequence, including derived states and observables. The episode selection, recordings, and tolerances remain fixed, and both failed attempts remain documented. Successful amended replay validates the new direct trajectories; historical missing rotations remain outside its evidence.

A relocation check moves the LIBERO source, configuration, and asset prefix within the same environment. All 99 Python files and linked asset bytes remain unchanged, and the six replays pass the same physics, controller, and label checks. XML hashes agree after normalizing only the documented LIBERO/RoboSuite asset namespaces; original exact-XML records remain available. SHA-pinned safe JSON restores Python/NumPy/Torch-CPU RNG without public-pickle loading. Saved-action replay uses neither policy inference nor CUDA physics, so captured CUDA RNG is unused. The check establishes relocation compatibility for these episodes, separately from clean-machine reconstruction of dependencies and assets.

The new direct records and the historical reconstruction retain separate provenance. Historical reconstructed rates are 39.3% spatial containment, 22.7% full-visual placement, and 27.3% calibrated placement, without direct access to their original rotations. Earlier direct full-visual counts are 9/7/11 against 27/32/24 native counts. Although the new 65.3%/27.3% means equal the historical means, that numerical agreement does not verify the missing rotations. Training contrasts continue to use their historical controls. Native/local hashes and counts agree exactly, with interval-bound roundoff at most 2.85×10^{-14} percentage points.

S.3 What the SIMPLER grasp detector measures

We construct four prespecified controls in native SIMPLER/SAPIEN to test the released grasp detector, without policy inference or sensor stubs. A saved-action sequence supplies the two-finger positive control. Joint drives hold the robot while the object remains dynamically simulated; we do not lock the object or force its pose at each sample. Each control must provide five observations after native 0.2-s control transitions (500-Hz physics; 0.80-s sample span). We read contacts and impulses

at the final physics substep. The contact proxy requires the sum of contact-point impulse norms to exceed 10^{-6}. Native grasp detection instead uses both fingers' net impulses and opening-angle conditions.

Table 47: Grasp and contact checks in native constructed states. Counts refer to observations from one selected construction, not independent episodes. External support is a thresholded upward-contact-impulse proxy. None of the accepted controls satisfies the task's target relation or source-motion condition.

Constructed state	Detected grasp	Robot contact	External support
Free external support	0/5	0/5	5/5
Two-finger grasp	5/5	5/5	2/5
Single-finger contact	0/5	5/5	5/5
Gripper housing/palm	Unavailable at the five-state criterion		

In five consecutive observations, one finger contacts an externally supported zucchini while native `check_grasp` remains false. Thus, absence of a detected two-finger grasp does not imply absence of robot contact. This is not a demonstrated full-task false positive: none of the accepted controls meets the target relation or motion requirement, and single-finger contact with an externally supported object is not necessarily a placement failure. The two-finger control also has external support for its first two observations, then loses it for the last three. Six frozen palm/housing constructions fail to sustain contact for five states; one briefly records `gripper_bar_link` contact before slipping. We retain every failed trace.

The experiment executes 153 control transitions, including replay of a 98-action historical construction, and records 60 observations over twelve construction attempts. During replay, the reset-state hash and all four historical task signals match. Contacts measured during replay are new observations, rather than recovered historical measurements. A dependency-free check recomputes impulse magnitudes and angles, contact proxies, accepted-state conditions and source/input/protocol hashes. This experiment adds no training or benchmark policy episodes.

This calibration changes no CoVer or language-intervention success counts. The old 420-trace corpus lacks finger and palm measurements. All 166 frozen-strict successes stop before the original 150-step budget. Replaying those saved prefixes with a contact-free criterion could test sensitivity within the prefixes, but could not establish failure over the full horizon. That comparison would need a separately frozen full-horizon protocol. Here, the controls support more precise language for the existing grasp proxy.

S.4 How native RoboCasa calls change the success counter

We run the native reward/Gym path for OvenBroilFish, whose counter advances on queries, and CoolKettle, whose clock advances outside the query method. We select them from source dependencies before observing native results. The experiments use real MuJoCo scenes and native helpers, with no policies or substituted sensors. Layout 15, style 34 and seed 0 are fixed. Each task has two reset episodes with privileged preparation, followed by zero 12-dimensional controls. The diagnostics execute 333 controller transitions and 666 native queries: 333 reward reads before fixture update and 333 Gym reads after it.

Kitchen `_post_action` reads base reward before `update_state`; Gym reads success after the native step. The two calls therefore observe different sensor epochs within one controller transition. Alongside this unchanged path, we run separate monitor histories with one, two, four, or one-plus-three-cached reads at each declared post-update boundary. We restore only task monitor fields; all histories share physics and native helpers. The cached history calls the original method once, then returns the saved label. It never reuses a label across a fixture update or episode reset. This experiment defines a one-read post-update reference in a shadow monitor. It does not install a reward/Gym repair or preserve every historical pre-update reward event.

The native calling sequence explains the earlier Oven success: both published callers advance the counter, so Gym reports first success at transition 6 rather than the one-read reference's 11 (Table 48). With four reads, the door-opening case reaches success at 3. At transition 4, `door_closed` becomes

Table 48: First transition reporting success. n counts controller transitions and excludes preparation. Native is the published Gym path; 1/2/4/Cache are post-update monitor histories. Oven episode 0 stays qualified, while episode 1 opens/closes the door at 4/5. CoolKettle episode 0 starts with the burner off; episode 1 keeps it on for transitions 1–3.

Case	Episode	n	Native	1	2	4	Cache
OvenBroilFish	0	12	6	11	6	3	11
OvenBroilFish	1	16	10	15	10	3	15
CoolKettle	0	151	151	151	151	151	151
CoolKettle	1	154	154	154	154	154	154

false, yet the existing native latch retains `success_timer=10` and returns four true labels because the threshold check precedes the state check. It remains true when the door closes at 5. One-, two-, and cached-read histories instead reset at 4 and first succeed at 15/10/15.

Across 28 Oven states, two reads change the last returned label on 10 states and monitor state on 21; four reads change them on 20/22 states, with maximum counter differences of 5/10. Caching removes these differences. All histories agree on 305 CoolKettle states, where `update_state` advances the clock independently of success queries. The comparison includes Oven's `success_timer` and CoolKettle's `cooldown_timer, cooling_done`; both CoolKettle resets begin at zero/false. Shadow monitors share native sensor values and `qpos, qvel, ctrl, act` and time hashes. This isolates read scheduling at a shared physical state, rather than testing separate-process replay or every model field.

The first Oven scene initializes, but five fixed static offsets fail to establish rack contact. We retain its exception and protocol records. Because the recorder appended preparation data only after a successful return, it omitted the five attempted positions. A recorded amendment settles the same candidates for 100 simulator substeps (0.2 seconds at the native 0.002-second timestep) and saves native inputs and settled poses. The first candidate qualifies in both episodes. These preparation substeps are excluded from controller-transition and query counts. After preparation, the fish is neither repositioned nor constrained. Records retain reset monitors, scene XML and the unchanged monitored-environment identity.

We pin unchanged source to RoboCasa `4f8a2980def7` and robosuite `5ce6643f3092` (1.5.2), retaining full revisions and Kitchen/Gym/core/task/helper hashes. The runtime uses Python 3.10.12, MuJoCo 3.3.1, NumPy 2.2.5 and CPU OSMesa. The restricted object pool contains Objaverse `fish_0`/`plate_0` and available Lightwheel kettles (sampled `Kettle006`), with native category scaling and predicates unchanged. Modern fixtures and Lightwheel objects come from fixed IIFAN mirror revisions. Box was unreachable, so we could not independently check their claimed byte identity with official Box assets. Selected ZIP members pass CRC, size and SHA-256 checks, but the complete archives are not verified. Fish/plate, accessories and textures use official Hugging Face revision `1b92c3d02ca4`. Style 34 needs four generated textures even with augmentation disabled. We compute the ordinary-texture archive SHA without an upstream expected hash. Manifests retain the full pins and records.

Standard-library checks recompute table entries, callers, compared sensors and states, labels, monitor histories and source/input/protocol hashes. The original 9/12 source inventory still uses mocked sensors; CoolKettle supplies an additional clock control. The one-read history is our operational reference, rather than independently certified semantic completion. This experiment tests one query-sensitive task through the native reward/Gym path. It does not cover the collector or every wrapper, and its correlated states cannot estimate defect rates in policy trajectories.

T Statistical Sensitivity and Validation Controls

T.1 Paired SIMPLER results and the influence of each task

For the four CoVer-minus-base comparisons, we compute marginal approximate 95% Tango paired-binomial score intervals on the unchanged 140 task/start pairs (Tango, 1998). We continue to

base significance statements on two-sided exact conditional McNemar tests. Frozen strict@5 is the main endpoint. The no-ordering@5 and task-removal analyses are post-hoc, exploratory and unadjusted. All intervals are conditional on these seven selected tasks and reused starts; they do not describe performance on unseen tasks. This analysis adds no training, policy inference or benchmark trajectories.

Table 49: Paired differences in strict success between CoVer and base. Every comparison uses $n = 140$. Differences and Tango intervals are in percentage points. Ties include cases where both policies succeed and where both fail.

Endpoint	CoVer arm	Gain/loss/tie	Δ	Tango 95% CI	Exact p
Frozen	Released	35/14/91	15.0	[5.4, 24.6]	0.003802
Frozen	Reconstructed	38/13/89	17.9	[8.2, 27.4]	0.000621
No-order	Released	32/13/95	13.6	[4.3, 22.8]	0.006609
No-order	Reconstructed	36/14/90	15.7	[6.0, 25.3]	0.002602

To show how tasks affect the aggregate, we remove each of the seven tasks in turn from each comparison, leaving 120 pairs. All 28 estimated differences stay positive, though some exact tests no longer reject zero difference. Removing tennis-ball gives frozen released $+7.5$ points ($p = 0.1755$) and no-order released $+8.3$ ($p = 0.1214$). Removing Red Bull gives no-order reconstructed $+10.8$ ($p = 0.0596$). Together, tennis-ball and Red Bull contribute 18 of 21 and 20 of 25 net gains for frozen released/reconstructed CoVer, and 15 of 19 and 17 of 22 without ordering. These signed contributions describe the sample; they are not causal fractions or independent replications. We retain both negatively affected tasks in the aggregate.

Table 50: Results after removing each task. Cells give paired differences in percentage points and exact McNemar p values on the 120 remaining pairs. R/C mean released/reconstructed CoVer minus base. `revision_v20/stats_simpler.json` retains gain/loss/tie counts and intervals for all 28 comparisons.

Removed task	Frozen R	Frozen C	No-order R	No-order C
Carrot	20.0 (0.0003)	23.3 (2.54×10^{-5})	18.3 (0.0005)	20.8 (0.0002)
Eggplant	15.0 (0.0096)	17.5 (0.0025)	13.3 (0.0166)	15.0 (0.0096)
Red Bull	12.5 (0.0237)	13.3 (0.0195)	10.8 (0.0410)	10.8 (0.0596)
Spoon	15.8 (0.0066)	16.7 (0.0037)	14.2 (0.0115)	14.2 (0.0137)
Stack cube	15.0 (0.0114)	17.5 (0.0031)	13.3 (0.0195)	15.0 (0.0114)
Tennis ball	7.5 (0.1755)	11.7 (0.0336)	8.3 (0.1214)	11.7 (0.0336)
Zucchini	19.2 (0.0006)	25.0 (9.25×10^{-6})	16.7 (0.0022)	22.5 (6.57×10^{-5})

An approximate confidence interval and a discrete exact test can give different answers near the threshold. In the seed-0 red-cup cell of Table 46, four of 25 pairs pass only the native check and none passes only the calibrated check. The Tango interval is [0.5, 34.7] points, but exact McNemar conditions on the four discordant pairs and gives $2/2^4 = 0.125$. The interval's exclusion of zero therefore does not establish significance under our exact-test convention. We retain both results; failure to reject also does not establish equality. The recomputation uses the standard-library interval implementation from the direct-geometry analysis. Checks against published examples and all task-cell totals pass, and deliberately corrupted counts are rejected.

T.2 Applying the frozen rules to other source templates

Before judging new cases, we freeze the existing deterministic candidate rules, use the BDDL declaration as the instruction reference, and specify the source fields, conservative categories and acceptance criteria. We select cases using pinned census metadata and path-hash ranking, without using new outcome labels. We exclude previously judged candidates, caddy tasks, recorded reset/cross-scene evaluation tasks and their scene prefixes. This leaves 27 eligible LIBERO-90 tasks. We select six cases from six scene templates: two containment, two support and two articulation declarations. A known caddy conflict and an equivalent region encoding serve as positive and negative controls. We inspect every selected case, including those that do not trigger the candidate rule.

Two agents record separate judgments before comparing them. The selecting agent knows the control identities. The second receives complete B01–B08 source packets with filenames, prior categories and control flags hidden. Both agents have manuscript context, the source can reveal task identities, and the caddy control is recognizable. Thus, these are separate agent judgments with partial label hiding, rather than fully blind or third-party validation. All 130 sources had already been mechanically parsed. The exposure manifests guide exclusions but cannot prove that no agent previously saw the six sources. We retain the frozen protocol, both unedited judgment records, source packets and uncertainty fields.

Table 51: New source-review cases and known controls. Line numbers refer to complete B01–B08 source packets. Each row lists the declared relation, scored target and both agents' category. Agreement about source code does not verify physical task completion.

ID	Declared event / scored target	Category (both)	Key lines
B01	Bowl right of plate / table-owned plate-right region	Equivalent; control	3,28,65,69
B02	Stove off / `Turnoff(stove)`	Source-consistent	3,60,64
B03	Bottle in bottom drawer / cabinet-owned bottom region	Source-consistent	3,59,87,91
B04	Microwave closed / `Close(microwave)`	Source-consistent	3,75,79
B05	Bowl in top drawer / cabinet-owned top region	Source-consistent	3,64,96,100
B06	Pan on stove / stove-owned cook region	Source-consistent	3,39,65
B07	Middle bowl atop cabinet / cabinet-owned top side	Source-consistent	3,61,91,98
B08	Mug in caddy compartment / table-owned exterior region	Conflict; control	3,72,83,120

None of the six new cases yields a confirmed declaration–goal conflict or an unstated requirement to maintain an initial condition. Both controls receive their expected categories, and the agents agree on all eight categories. This shows that the source-review steps can be applied beyond the caddy template. It does not measure detector recall, inter-annotator reliability, defect prevalence or physical transfer. We retain two uncertainties: B01's equivalent encoding assumes the initialized reference plate and an unverified direction convention, rather than arbitrary plate motion; B07's middle-object alias can be traced in source, but its physical ordering is unverified. This source-only review does not check asset extents, predicate thresholds, full-object containment, release or persistence. Mechanical checks confirm parser-derived goals and cues, source hashes and line bounds, and reject a deliberately substituted wrong object/owner. These checks establish traceability, not the correctness of each semantic judgment.

T.3 Geometry controls and reproduction in new environments

Before constructing states, we freeze the existing 2-mm rule, two cup colors, a 10-mm protrusion margin, a bounded gripper-placement search and expected checks. For each color, we construct a supported positive, side protrusion, top protrusion and gripper contact. We first run them in the original runtime, then in separately installed simulator virtual environments. Each run has eight constructions and 260 controller transitions, giving 16 constructions and 520 transitions in total. Repeating these templates adds no independent scene coverage. At the construction boundary, all checks match their expected values in both runs (Table 52). Side protrusion retains floor support. Top protrusion also loses floor contact, and gripper contact also fails spatial and support checks, so these two controls do not isolate one clause. We retain ten subsequent controller steps, but they are not final-50 policy or persistence trials. Fresh records save MuJoCo geometry IDs and contact pairs, allowing independent recomputation of floor/gripper flags for constructions, searches and traces. The original run lacks this ID-based check.

Each negative control is rejected at the constructed snapshot. It need not remain negative after simulation resumes. In the next ten steps, top protrusion returns to calibrated placement in 9/10 red and 8/10 white states; white side protrusion returns in 7/10. Both runtimes give the same results. The gripper constructions have minimum contact distances of -45.47 mm for red and -42.69 mm for white, indicating deep interpenetration. They test collision detection after privileged state construction, rather than realistic release trajectories or persistent failures.

We replay two caddy cases in the newly installed LIBERO environment: seed 0/red/index 5 and seed 1/white/index 5. Each provides 600 compared actions plus ten initializing settling steps, for 1,200 compared actions and 20 settling transitions. Their native/calibrated final-50 labels are (0,0) and (1,0). Before the new replay, we add the white case from the previously fixed six-case set

Table 52: Checks at the constructed snapshot, identical across red/white cups and original/fresh runtimes. Side/top/bottom indicate passing the calibrated spatial checks; floor/gripper indicate contact. Expected calibrated placement is 1 for the supported positive and 0 for each negative. Every result matches its expectation.

Construction	Side	Top	Bottom	Floor	Gripper	Calibrated
Supported positive	1	1	1	1	0	1
Side protrusion	0	1	1	1	0	0
Top protrusion	1	0	1	0	0	0
Gripper contact	0	0	1	0	1	0

because its known outcome illustrates a native-only success. These selected cases do not form a rate sample. Normalized world XML, initial/settled controller snapshots, positions, bounds and physical states match. All labels match, there is no first divergence, and the maximum angular discrepancy is 6.39×10^{-6} degrees, within the frozen tolerance. These checks validate selected newly recorded episodes; they do not observe rotations missing from historical runs.

In a separate newly installed RoboCasa environment, we rerun the unchanged OvenBroilFish and CoolKettle diagnostics. They execute 28+305=333 controller transitions and 666 native reward/Gym queries, with two reset episodes per task. Cached shadow labels and complete monitor histories match the one-read reference on 333/333 states. All eight captured source fields per case, including object utilities, match their file pins. This reproduces the native diagnostic. It does not compare old and new physics at every state or deploy a reward/Gym repair.

Both simulator stacks use separate new virtual environments and copied, pinned Python sources and configuration on the original host, with original assets. Fresh LIBERO uses the declared Torch 2.7.1 CPU build for action replay. We retain imported module roots, dependency inventories, world XML, full run records, failed installs and launches, and runtime/input amendments. Full-scope standard-library checks of saved records pass, and deliberately corrupted side/contact records are rejected. The four reproduction layers remain separate (Table 36). We have not verified execution on an independent machine, full asset/weight reconstruction, third-party validation or anonymous signed-out download. These controls and replays add neither optimizer updates nor learned-policy evaluation episodes.

T.4 Fixed-compartment learning with the label study budget

Before training or evaluation, we freeze one Qwen seed-0 diagnostic using the label study's initialization, rank-16 LoRA, crop/preprocessing, optimizer, sampling schedule, controller and final-update checkpoint rule (Table 41). We take all 18 correct-text compartment demonstrations from the same 36-demo pool, retaining observations, actions and source identities; equal-demo sampling mass is one. Training completes 600 updates/9,600 windows (4,753 red; 4,847 white), with no instruction mismatches or changes to frozen parameters. Initialization digests match the old coherent seed-0 arm. The global budget matches, but filtering increases compartment exposure and changes the task mixture. Per-target gradient budgets therefore differ. The 95-demo/4,200-update adaptation uses a separate protocol.

We evaluate the final adapter with compartment text on the same 20 frozen starts (ten/color), execution seeds and canonical inputs. Each run uses ten settling steps and 600 actions without stopping at first success. A collector attached to the current simulator records post-action geometry without extra forwarding. Three workers split the manifest 7/7/6, and all exit successfully. Checks pass for initial inputs, protocol and adapter hashes, case identities and order, and raw, worker and merged metadata. We independently recompute every label at 0/1/2/5 mm over all 12,000 frames. Floor/gripper flags also match decoded contact pairs, using the frozen XML geometry inventory and captured native IDs. Amendments, training archives, GPU snapshots and logs remain available. We have not checked whether serial and parallel execution produce identical actions.

The policy reaches 8/20 native hits, 6/20 qualified hits and 3/20 calibrated final-50 completions, showing limited single-target completion in both colors. Two native-hit episodes never qualify; three qualifying episodes fail final-50. There is no step where the calibrated check is true and the

Table 53: Fixed-compartment results for one seed. Qualified hit and geometric final-50 use the unchanged 2-mm floor-calibrated placement proxy, including support and absence of gripper contact. Native final-50 also excludes gripper contact. Each color uses ten frozen starts, reused from the label study rather than new independent scenes.

Color	n	Native hit	Qualified hit	Geometric final-50	Native final-50
Red	10	4	4	2	3
White	10	4	2	1	4
Both	20	8	6	3	7

native check false. For comparison, the old mixed coherent seed 0 arm gives compartment counts of 2/1/1 out of 20 and joint success of 0/20; these measure different events. Nonzero fixed-target completion does not establish a strong skill ceiling or conditional target selection. The single seed and architecture, changed mixture and exposure, supervision quality and execution leave label effects unseparated from other causes. We observe failures to reach, qualify and sustain placement, but cannot identify their causes. Conditional-label effects remain unestablished, and we do not expand seeds based on these outcomes.

U Post-hoc saved-record audit

U.1 Post-hoc geometry-clause sensitivity

We next re-read the 150 direct V19 executions (600 post-action frames each) through an adapter tied to the frozen recorder, evaluator and per-file hashes. Before any new decomposition, the adapter reproduced the retained 2-mm native/contact-free and calibrated counts in every seed–color cell. Table 54 then removes only the top-envelope clause. Side bounds, collision-bottom tolerance, floor support, absence of gripper contact, and the final-50 window remain unchanged. This is a post-hoc sensitivity analysis; the original calibrated result remains primary.

Table 54: Direct V19 outcomes by training seed and cup color. N is native-`In` with no gripper contact throughout the final 50 actions; G is the original 2-mm calibrated criterion; no-top removes only its top-envelope clause. Counts are out of 25 per color and 50 for Both.

Seed	Cup	N	G	No-top	Native-only rescued
0	Red	15	11	11	0
0	White	17	3	3	0
0	Both	32	14	14	0
1	Red	15	9	9	0
1	White	21	7	7	0
1	Both	36	16	16	0
2	Red	14	9	9	0
2	White	16	2	2	0
2	Both	30	11	11	0

Removing the top clause rescues 0 episodes because every top failure among native-only outcomes co-occurs with another retained failure. Both N and G already require no gripper contact, so their disagreement is not caused by adding a release requirement. The binary clauses do not identify a millimetre-scale protrusion distance, and we do not infer one.

V Action-interface checks on supervision validity

This section reports supplementary action-interface checks on whether qualified supervision remains valid when executed. It keeps three objects distinct: a saved execution receipt, a fresh execution of the saved raw actions, and a matched fresh execution of actions reconstructed by a frozen codec. A disagreement between the first two is replay uncertainty. A codec-associated claim uses the second and third in separate fresh environments under the same XML, flattened MuJoCo state, simulator

Table 55: Overlapping failed-clause counts among native-only V19 episodes. S/T/B/F denote side, top, collision-bottom and floor-support failures. Patterns are mutually exclusive episode-level combinations; the clause columns overlap.

Seed	Cup	Side	Top	Bottom	Support	Exclusive patterns
0	Red	4	0	0	0	S:4
0	White	13	8	0	9	S+T+F:8; F:1; S:5
1	Red	6	0	0	0	S:6
1	White	14	11	0	11	S+T+F:11; S:3
2	Red	5	2	0	2	S:3; S+T+F:2
2	White	14	11	0	11	S+T+F:11; S:3

seed, fresh-controller initialization recipe, execution horizon, and checker. Historical OSC goals, reference orientation, and interpolator state are not serialized or restored. The preserved V27 sections that follow retain the complete human, learning, repair, cost, and negative-result evidence.

Action representations and closed-loop execution. Robot policies commonly emit temporally extended continuous chunks or discrete latent actions (Zhao et al., 2023; Chi et al., 2023; Shafiullah et al., 2022; Lee et al., 2024; Mete et al., 2024). Generalist policies and VLAs make the execution boundary consequential because a predicted block may be decoded, clipped, and only partly executed before the next observation (Brohan et al., 2022; 2023; Kim et al., 2024; Octo Model Team et al., 2024; Black et al., 2024). Recent tokenizers explicitly optimize action compression and downstream control efficiency (Pertsch et al., 2025; Liu et al., 2026a;c); the released policy and adaptation systems reused here instantiate related choices (Tan et al., 2025; Ganai et al., 2026). Their downstream evaluations establish that representation quality matters, while our question is different: which executed task condition changes under one registered interface? We therefore analyze physical clause traces under a fixed block size, executed prefix, and replanning rule. This prevents a historical eight-action replay from being presented as if it were the four-action student evaluator, and it keeps the QueST 32/8 SkillVAE path separate from both Qwen paths. Our empirical increment is a clause-level, matched physical intervention rather than the general observation that compression loses information.

V.1 Registered action paths

The action path is part of the run identity rather than an interchangeable implementation detail. Table 56 records the block and execution horizons relevant to the paper-facing results. The Qwen source audit uses the frozen native eight-action tokenizer and executes all eight decoded physical actions. E2 policy evaluation instead executes four actions before replanning and is therefore reported as a separate runtime. The QueST SkillVAE consumes 32 continuous-action labels and executes eight decoded actions before replanning. All A1 physical replays clip actions to $[-1, 1]$ and use the source-registered model, normalization, gripper conversion, reset, and controller code.

Table 56: Each result is bound to one action path. B is the coded or predicted block and H the executed prefix before replanning.

Family	Evidence path	B	H	Scope
Qwen	native-codec source audit	8	8	raw/decoded replay
Qwen	E2 learned-policy evaluator	8	4	policy outcomes
QueST	frozen SkillVAE	32	8	raw/decoded replay and policy

The exhaustive caddy population is frozen in protocol `a1_caddy_action_audit_protocol_v4.json`, which deduplicates 238 group memberships to 112 source HDF5/group pairs. The source membership, actions, codec weights, task, checker, tolerance, block size, and horizon remain fixed.

V.2 Saved-to-fresh replay uncertainty

The two caddy booleans use the saved reference available to the complete A1 audit; they are not reconstructed from aggregate success rates. For the support-qualified bit, saved strict success is 42/95

Table 57: Saved-to-fresh agreement is replay evidence, not a codec effect. Exact physical equality compares the complete endpoint object or per-frame clause matrix stated in the row.

Population	Sources	Agreement	Compared object
Caddy mother pool	95	95/95	native success bit
Caddy mother pool	95	72/95	support-qualified bit
Caddy repaired	17	17/17	prior raw endpoint

and fresh raw strict success is 45/95: 40 sources fail both, 13 change from failure to success, 10 change from success to failure, and 32 succeed in both. Thus 72/95 is agreement, not a success count. The 17 repaired rows compare a fresh raw replay with the prior complete raw physical audit. These results establish that task-relevant saved and fresh labels can differ before any action encoding is applied.

V.3 Complete raw-to-decoded source audit

Across the 112 unique caddy sources, exact raw-versus-decoded physical endpoint equality occurs for 0/112 sources under the Qwen native codec and 0/112 under QueST SkillVAE. Mean per-source action MAE is 0.011809 for Qwen and 0.020584 for QueST. The maximum errors are dominated by the gripper channel. These action and endpoint differences do not by themselves establish a success change, a direction of effect, or a gripper mechanism.

The previously reported historical teacher oracle is a separate 18-start diagnostic of the frozen QueST 32/8 action path. Raw actions obtain 17/18 native, 15/18 calibrated, and 11/18 original-visual joint successes. Frozen encoding and decoding obtains 4/18, 2/18, and 1/18. Every eight steps the oracle encodes the next 32 teacher actions, pads only at the end, and executes the first eight decoded actions. V28 independently verifies this implementation identity but cannot recompute the aggregate because the local handoff lacks the full per-frame oracle artifact. Both modes restore the same start once and use the same fresh-controller initialization recipe. Because the oracle knows the teacher sequence and does not infer actions from observations, it measures action-interface fidelity rather than student-policy performance.

V.4 Historical Qwen native-codec 8/8 paired diagnostic

The paired diagnostic was frozen before completion of the 72-attempt audit and contains all 12 attempts whose stored and fresh-raw physical endpoints disagree. For each attempt, three new environments restore the same XML, initial state, and deterministic seed and execute raw versus decoded actions in AB/BA/AB order. Raw endpoints are identical across the three repeats for every attempt; decoded endpoints are also repeat-stable. The selection is development based, so its rates are not population estimates.

Table 58: Decoded-minus-raw clause direction in 36 historical Qwen native-codec 8/8 physical executions. No-gripper-contact never changes; geometry and support clauses change in both directions.

Clause/count	Decoded higher	Equal	Decoded lower
Native final-50	0	27	9
Support placement, 2 mm	9	12	15
Floor contact	15	18	3
No gripper contact	0	36	0

Only 6/36 pairs preserve the complete tuple of the four final-50 counts. The strict 2-mm endpoint holds for 12/36 raw executions and 6/36 decoded executions. Two raw failures become decoded successes, so the result is not a monotone degradation. No-gripper-contact equality in all 36 pairs shows that the observed strict changes are not created by an added release condition, even though the maximum action error lies in the gripper channel. This historical 8/8 diagnostic is not interface-matched evidence for the Qwen 8/4 student.

V.5 FROZEN MOTION–GRIPPER CROSSOVER PROTOCOL

For action (m_t, g_t), the four conditions are raw motion/raw gripper (A), decoded motion/decoded gripper (B), decoded motion/raw gripper (C), and raw motion/decoded gripper (D). Each source–interface group uses separate fresh environments under the same restored state, fresh-controller initialization recipe, execution window, clipping, and checker. The Qwen intervention decodes eight actions and executes four before the next block; QueST encodes 32 and executes eight. Qwen 8/8 arrays from the historical audit are not reused. The source rule, development/held-out split, and intervention construction are frozen before outcome access.

Table 59: The crossover distinguishes motion, gripper, and interaction effects; it does not provide a deployable policy repair.

Condition	Motion	Gripper	Question
A	raw	raw	Does fresh raw replay satisfy the task?
B	decoded	decoded	What changes through the full interface?
C	decoded	raw	What is restored by teacher gripper actions?
D	raw	decoded	What is restored by teacher motion actions?

The frozen run completes 12 sources, two interfaces, and four conditions for 96 fixed-action executions with zero policy-training updates. All action arrays, traces, and endpoint summaries are hash-bound; an independent recomputation on a second machine reproduces the 96-row file and summary byte for byte.

The development split suggests a pose-only recovery, but the pattern does not replicate on confirmation. Qwen has no confirmation source where A succeeds and B fails; its decoded condition instead converts two raw failures to successes. QueST has one such drop, and both C and D recover it. Thus neither motion alone nor gripper alone is isolated as a stable mechanism. Even a positive teacher-component restoration would remain a mechanism test because a deployed policy does not observe the teacher action; it cannot be described as a usable repair without a separate deployment-time predictor and validation.

V.6 BOUNDARIES FOR THE LEARNING CLAIM

E2 retains its complete heterogeneous results below. The replay analyses do not explain those learning outcomes because they do not retrain a student under an interface-corrected control. Conversely, the learning outcomes do not validate a codec mechanism because multiple data and closed-loop factors remain. CGSR remains a generator intervention with an exact tie against random source ordering at the development gate. The RoboCasa query-count and binding controls remain separate interface examples and are not combined with the caddy action population.

W COMPLETE SUPERVISION AND REPAIR STUDIES

This appendix separates the frozen supervision follow-ups from the historical fixed-target adaptation. Here RQ1 denotes task correspondence, RQ2 measurement and learning, and RQ3 diagnosis-conditioned repair under common cost accounting. All counts below come from saved machine-readable records. The task-source audit supports RQ1; same-trajectory checker comparisons support the measurement part of RQ2; generator receipts support CGSR executability for RQ3. None of those categories substitutes for a fixed-evaluation learning result. A generator success means that a complete robot-action execution passed the stated checker; it is not a policy-evaluation success.

Demonstration generation already reuses or transforms successful behavior, as in MimicGen and constraint-preserving generation (Mandlekar et al., 2023; Lin et al., 2025). Prefix reuse and geometric transformation are therefore not new capabilities claimed here. The repair comparison below concerns acceptance, failure diagnosis and complete execution validation under a shared budget.

Table 60: Pose/gripper component cross under the interface-matched schedules. Each cell is strict final-50 successes out of six hash-selected sources. Development and confirmation sources were frozen before execution; all outcomes are retained.

Phase	Interface	A: raw/raw	B: dec./dec.	C: dec./raw	D: raw/dec.
Development	Qwen 8/4	4/6	2/6	1/6	4/6
Development	QueST 32/8	4/6	0/6	0/6	4/6
Confirmation	Qwen 8/4	2/6	4/6	3/6	2/6
Confirmation	QueST 32/8	2/6	1/6	2/6	2/6

Table 61: Component recovery among confirmation sources where A succeeds and B fails. "Pose only" means D recovers while C does not; "gripper only" means C recovers while D does not. These are mechanism interventions using teacher components, not deployable policy repairs.

Interface	A pass/B fail	Pose only	Gripper only	Both	Neither
Qwen 8/4	0	0	0	0	0
QueST 32/8	1	0	0	1	0

W.1 Contract-Guided Supervision Repair

CGSR maps a failed source trajectory to an accepted repair or an explicit abstention. Let $B_d = (b_1, \ldots, b_m)$ be the rollback order for diagnosis d, from the latest admissible boundary to earlier boundaries. At each b_i, a deterministic bounded search proposes low-dimensional placement, approach, orientation, release, and retreat parameters. Prefix replay, rejected candidates, validation re-execution, simulator steps, planner time, wall time, and any manual intervention are charged to the attempt ledger.

Diagnosis	Rollback order	Repair action
Pass	none	Retain the verified trajectory.
Wrong goal	divergence → pre-grasp	Replan the target and transport; never relabel old actions.
Geometry	pre-place → transport → grasp	Search target pose, yaw, approach, and entry direction.
Release/time	pre-release → pre-place → transport	Search release timing, retreat, and post-release control.
Ambiguous	none	Abstain and retain the reason.

A candidate is accepted only after reconstruction from the registered initial state through the common robot action interface. Direct object placement, observation concatenation across inconsistent states, and validation of only the originally failed clause are forbidden. The resulting training artifact stores aligned pre-action observations and actions, post-action states, source and pair identities, code and configuration hashes, and the checker receipt. This design makes a successful repair a verified demonstration, not evidence that a trained policy will improve.

W.2 Learning and evaluation protocol

E2 holds the task reference, Qwen2.5-OpenVLA initialization, rank-16 LoRA recipe, 600 optimizer updates, red/white task sampling, and frozen evaluation fixed. The post-action mother pool provides five views: Mixed (95 sources), Filtered (42), Random-matched (42), Mixed-paired (52), and Paired-repaired (52). The paired arms share the 42 qualified sources and ten failed sources whose historical teacher/reference replacements passed the recorded checks. These replacements are not CGSR outputs and do not support an automatic-repair efficiency claim. The comparison estimates the effect of replacing complete supervision trajectories; it does not isolate observation, action, and length changes inside a replacement.

Model selection and final testing use disjoint frozen starts. The only saved checkpoint is the adapter at update 600, so no outcome-based checkpoint choice is permitted. Primary seeds are 0, 1, and 2. Additional seeds for the paired arms are exploratory and must be reported together rather than selecting the best seed.

The later 16-source acquisition-to-student comparison uses its own verified data and fixed training/evaluation protocol (Appendix A.3).

W.3 Post-action mother-pool recertification

We re-executed all 95 native-selected caddy demonstrations from their recorded initial states with their complete original action sequences. Inputs remain pre-action, while the corresponding native predicate, full-envelope geometry, floor contact, gripper contact, and 2-mm support-qualified label were recorded after every action in one uninterrupted execution. No source HDF5 was overwritten.

Table 62: Post-action recertification of the V27 caddy mother pool. "Qualified" requires the frozen 2-mm support-qualified condition at every final-50 state.

Task	Sources	Native final-50	Qualified final-50
Red mug	46	46	36
White mug	49	49	6
Total	95	95	42

The aggregate 42/95 matches the historical saved-suffix audit, but membership differs. White source 19 changes from fail to pass and white source 34 from pass to fail. Source 19's historical restored suffix extended approximately 2.05 mm beyond the inner-y boundary and passed 2/50 frames; uninterrupted replay passes 50/50. Source 34 changes from approximately 1.32 mm beyond the boundary and 50/50 passing frames to approximately 3.46 mm and 0/50 at 2 mm, while remaining 50/50 at 5 mm. We therefore froze an amended post-action split before any V27 optimizer update instead of silently reusing the historical identities.

The resulting arms contain 95 Mixed sources (46 red, 49 white), 42 Filtered sources (36, 6), 42 Random-matched sources (36, 6), and 52 sources in each paired arm (37, 15). The paired arms cover the 42 qualified sources and ten failed sources with verified historical teacher/reference replacements. These are not CGSR outputs, and the paired comparison does not measure automatic repair efficiency. It also does not cover all 53 failed sources. Every external link was dereferenced and every action sequence was checked against state and observation lengths before training.

W.4 Independent human final-50 validation

E1 contains 105 outcome-blind selected executions: 30 historical V19 executions, 15 E2 primary cells, 42 caddy Qwen student executions, and 18 caddy QueST student executions. The student videos support only a checker-versus-human comparison on observed executions, not a policy-arm performance comparison. The historical subset was stratified equally by training seed, cup color, and hidden checker stratum; each student-policy item was selected by the preregistered hash rule, one episode per completed experiment–arm–seed cell. Each opaque item showed all 50 saved post-action states from agentview and sideview. Rendering restored recorded states without policy inference or a physics step. Reviewers did not receive model or arm identity, source identity, checker output, or manuscript results.

The displayed instructions fixed the caddy direction convention before labeling. Left and right were judged only from agentview (the upper half of each video and upper montage row): the large compartment on screen right was the task's "right compartment." Sideview was used only to confirm containment and could not redefine direction; a mug merely outside the caddy on its right side was a failure. This rule is recorded as guidance version `v3-explicit-agentview-left-right` in every accepted label and lock.

Independent review yields highly consistent visible-outcome labels. The two non-author primary reviewers lock all 105 first-pass judgments before accessing other labels or hidden metadata, with

success/failure counts 40/65 and 39/66. They agree on 102/105 binary labels (97.1%, Cohen's $\kappa = 0.939$). A distinct third reviewer first locks judgments for all 105 items, then adjudicates only the three primary disagreements while remaining blind to checker, source, and model identity. The final set contains 39 successes and 66 failures, with 101/105 unanimous first-pass judgments. Ten hidden native-only boundary controls remain included regardless of outcome; four are judged successful and six unsuccessful.

Table 63: Agreement with the locked human E1 labels. H+ and H− denote human success and failure; C+ and C− denote checker success and failure. Both endpoints include the same no-gripper-contact requirement.

Endpoint	Agree	H+C+	H+C−	H−C+	H−C−
Native contact-free	95/105	39	0	10	56
Strict 2-mm support	97/105	31	8	0	66

The human comparison distinguishes the operating characteristics of the two endpoints. Native contact-free final-50 includes every human success plus ten human failures; the 2-mm support-qualified endpoint includes no human failure but misses eight successes. Both already require release, so the difference concerns their other conditions. For this supervision study, native contact-free final-50 remains the primary policy endpoint and 2-mm geometry the conservative secondary endpoint and treatment definition. The earlier V19 measurement study retains its own primary 2-mm endpoint. The two-view judgments characterize visible completion in the sampled executions, while simulator contact and broader policy superiority require separate evidence.

An earlier 30-item model-agent sparse-view pilot is excluded from the human results; its records remain separate.

W.5 CGSR DEVELOPMENT SEQUENCE AND GENERATOR VARIANTS

CGSR development proceeds through two versions on the same nine geometry failures. V1 repairs 4/9 sources in 30 attempts and 9,255 simulator steps, exhausting its attempt ceiling before reaching two sources. Inspection identifies missing whole-object yaw control, above-floor white-cup releases, and boundary-near placements. V2 retains the resulting yaw search and tries lower in-bay release before above-bay release, obtaining 9/9 repairs in 15 attempts and 4,833 steps. Because V2 uses V1 failure analysis, this comparison documents development rather than an independent estimate of method improvement.

The development controls identify which decisions change recovery. Every source has diagnosis GEOMETRY; omitting that label preserves the 9/9 outcomes, 15 attempts, and 4,833-step sequence, although cross-host arrays differ at the bit level. Fixed pre-place misses white-12 and white-44, which CGSR recovers by falling back to transport. The corrected early-generation-plus-final-filter route verifies 3/9 but also changes the boundary. Full recollection preserves no source prefix, uses a fixed grasp transform per color, and verifies 8/9 in 5,075 steps. Its white-47 lower candidate maintains floor contact for 49/50 frames, and its above-bay candidate fails to settle; no outcome-driven parameter is added.

The baseline originally named "random repair" randomizes source order while retaining the same repair operator. A frozen permutation reorders the nine immutable repair bundles under the same planner, proposal space, ceilings, and verifier. Because all bundles fit below the 5,100-action checkpoint, it completes the same 15 attempts and verifies 9/9 sources in 4,833 steps. The paired yield difference is 0 points with zero discordant sources, and action-cost reduction is 0%. The prespecified 10% improvement criterion is therefore unmet, leaving the versioned G2 decision at STOP EXPANSION.

The final-filter control was corrected after an implementation audit found that its generation stage used the full-envelope side-margin clause to choose yaw. That 7/9 result is invalid for the declared ablation and remains excluded with its failure notice. A frozen corrective amendment precedes the valid 3/9 comparison.

Checks of ledger hashes and HDF5 alignment cover states, post-action states, actions, rewards, dones, and observation streams for every verified artifact. They establish executable generation on the development cohort. Held-out recovery, diagnosis benefits, policy learning, and cross-task transfer remain separate evaluation questions.

W.6 Native action-codec audit

We audited all 72 recorded attempts from the five caddy generator routes through the historical native Qwen eight-action encode–decode path. The VQ weights and buffers were unchanged. Direct raw-action replay reproduced every stored per-frame label for 36/72 attempts and the four final-50 endpoint counts for 60/72. The remaining 12 stored-versus-raw endpoint differences are execution uncertainty and are not attributed to the codec.

Before the 72-attempt audit finished, we froze a paired diagnostic that included every one of those 12 endpoint-mismatch attempts. For each attempt, three fresh environments restored the same XML, initial state, and deterministic seed and executed raw versus encode–decoded actions in AB/BA/AB order. Raw endpoints were stable across all three repeats for every attempt, as were decoded endpoints. Only 6/36 paired executions (two of the 12 unique attempts) preserved the complete four-count endpoint tuple. The strict 2-mm final-50 endpoint held for 12/36 raw executions (four attempts) and 6/36 decoded executions (two attempts); the direction was nonuniform, including two raw failures that became decoded successes. No-gripper-contact counts were unchanged in all 36 pairs. The maximum absolute action difference was 2.0 in the gripper dimension for all 12 selected attempts, although their mean absolute errors were small.

This development-selected diagnostic establishes that encoding and decoding can change physical endpoints, including strict completion, without a uniform direction of effect. Its selection by saved-to-fresh instability confines interpretation to these cases rather than all 72 attempts or learned policies. The full-population raw/decoded comparison remains descriptive, and stored receipts do not substitute for a matched causal counterfactual.

We subsequently closed the action-interface audit over all 112 unique caddy sources represented by 238 group memberships (95 mother-pool and 17 repaired sources). No student-policy or final-test outcome was used to select these sources. All completed processes verified that the frozen Qwen VQ and QueST SkillVAE weights and buffers were unchanged.

Table 64: Complete V27 action-interface audit. "Saved/raw" uses the available saved reference for each population; exact physical equality compares the complete recomputed endpoint object, not only a success bit.

Population	Sources	Saved/raw	Raw/Qwen exact	Raw/QueST exact
Caddy mother pool	95	72/95 strict; 95/95 native	included below	
Caddy repaired	17	17/17 endpoints	included below	
All caddy	**112**	—	**0/112**	**0/112**

Fresh caddy raw replay preserves the saved native completion boolean for all 95 mother sources but preserves the stricter support-qualified boolean for 72/95. It exactly reproduces the earlier raw physical endpoint for all 17 repaired sources. Exact raw-versus-decoded physical endpoints occur for none of the 112 caddy sources under either action interface. The mean per-source absolute action error is 0.011809 for Qwen and 0.020584 for QueST; the maximum errors are dominated by the gripper channel. These denominators show that saved-versus-fresh execution uncertainty and encode–decode differences must be tracked separately. They do not provide a directionally uniform codec effect, a held-out estimate, or a student-policy result.

W.7 Frozen E2 protocol and reporting rule

E2 fixes the Qwen2.5-OpenVLA base checkpoint, rank-16 LoRA, 600 optimizer updates, batch size 4 with four-step gradient accumulation, learning rate 2×10^{-4}, 30 warmup steps, and a single saved adapter at update 600. Each batch samples red and white with equal probability, then samples uniformly over unique demonstrations within task and over valid windows. Images are vertically

corrected, resized to 224 pixels, center-cropped to 90% area, and resized again to 224 pixels. Policy execution uses four-action chunks for 600 actions.

All five data arms use primary training seeds 0–2. The additional seeds 3–9 belong only to the exploratory Mixed-paired/Paired-repaired comparison and are reported as a complete block. Final evaluation uses 25 red and 25 white official starts excluded from the historical runs examined when freezing the pool; a disjoint ten-per-task development set is recorded separately. The protocol retains every completed registered seed and final-test case, with any best-seed statistic explicitly identified as exploratory.

W.8 E2 PRIMARY AND EXPLORATORY OUTCOMES

The complete primary outcomes in Table 8 use 50 shared starts, 25 per color, for each of the 15 runs. Paired-repaired minus Mixed-paired differences are $[-4, -8, -2]/50$ for native contact-free final-50 and $[-2, +2, +3]/50$ for calibrated final-50. In exploratory seeds 3–9, the corresponding vectors are $[+1, 0, -8, +9, -7, -4, -6]/50$ and $[+2, +1, -3, +8, -6, 0, -2]/50$. Their differing directions show endpoint- and seed-dependent learning effects rather than a consistent replacement benefit. The records retain task-level and transient-hit counts alongside these totals.

W.9 SEPARATELY SCOPED ROBOCASA CONTROL

The separate RoboCasa binding control completes all 32 registered executions with byte-identical independent recomputation. Both corrected and buggy bindings succeed on 0/8 unique paired requested units, or 0/16 repeated executions per arm, across two tasks, two policy-noise seeds, and two requested scene seeds per stratum. The study uses 9,600 policy actions; formal evaluation takes 5,210 seconds and device calibration 768 seconds, with human adaptation time unrecorded. This null comparison is reported separately from LIBERO and addresses success counts, not trajectory equality or the general effectiveness of binding correction.

X TEMPORAL AND INSTRUCTION EVIDENCE ACROSS BENCHMARKS

Temporal and instruction tests. MINERVA's released L1 policy (Sendai et al., 2026) is evaluated on 40 LIBERO tasks $\times$ 50 starts, with hard resets, official horizons and task-ID inputs. The same policy continues for ten steps after first hit. OSMesa replaces the released EGL renderer, so published scores are references rather than paired baselines. An earlier 60-case SmolVLA screen and no-action holds probe predicate losses (Appendices B, G). RoboCasa uses source-method tests and two native controlled-state tasks (Section 4.3).

SIMPLER compares base π_0, released CoVer and a fixed-reference CoVer reconstruction (Kwok et al., 2026) on seven tasks and 140 shared starts. The stricter criterion was fixed *after* inspecting positional outcomes and grasp defects but before its evaluation; these are reused, selected cases. Historical successes are rerun with matched reset images, state/proprioception hashes, policy noise, and candidate/executed actions through the original stopping point. Intermediate physical states are not hashed. Follow-up stays within 150 steps. The reconstruction scores all 40 candidates at each replan. Instruction revision uses a 20-pair development block and separate 40-pair confirmation blocks (Appendix R).

Figure 8 summarizes the suite-level temporal comparison and the paired instruction effects.

X.1 FIRST-HIT RETENTION IS NOT EVENTUAL COMPLETION

On the same 2,000 LIBERO episodes, MINERVA reaches the native goal in 94.8% and maintains it for ten further policy steps (0.50 s) in 84.4%. The drop is 10.4 points, with task-cluster bootstrap interval [6.2, 15.7], and losses occur in 25/40 tasks. Of 209 losses, 161 later recover; the last follow-up state alone succeeds in 1,838/2,000 episodes (91.9%). The collector instead permits a later successful window. These results measure continued-policy retention, not passive stability or a replacement official score. Only 54/80 selected first-hit trajectories reproduce for the no-action test, precluding an overall hold-effect estimate (Table 20; Appendix B).

Table 65: CoVer's aggregate advantage survives both strict criteria on this selected corpus. Counts use 140 paired starts per policy. No-ordering removes only displacement ranking and is post-hoc; all paired intervals and tests are retained in Table 70.

Policy	Position hit	Frozen strict@5	No-ordering@5
Base π_0	50 (35.7%)	40 (28.6%)	43 (30.7%)
Released CoVer	79 (56.4%)	61 (43.6%)	62 (44.3%)
Reconstructed CoVer	74 (52.9%)	65 (46.4%)	65 (46.4%)

X.2 Stricter SIMPLER scoring preserves CoVer's aggregate advantage

The released evaluator computes target relation, motion and grasp but returns only the relation. Our frozen strict event requires

$$q_t = \texttt{src_on_target}_t \wedge \texttt{moved_correct}_t \wedge \neg\texttt{grasped}_t \tag{4}$$

for five consecutive states at 5 Hz. Motion requires source displacement > 3 cm and larger than every distractor's displacement. The ranking condition is not demanded by the instruction, so we also remove it in an explicitly post-hoc sensitivity analysis. Absence of the native qualifying two-finger grasp is not absence of all robot contact; a native single-finger control confirms that distinction (Appendix S.3).

Across 420 rollouts, frozen strict scoring accepts 166 of 203 positional successes. The 37 exclusions span all seven tasks. Removing displacement ranking gives 43, 62 and 65 successes per 140 starts for base, released and reconstructed CoVer. Their net paired gains over base are 13.6 and 15.7 points ($p = 0.006609, 0.002602$; Table 65). Gains are positive on five tasks and negative on carrot/zucchini; tennis-ball and Red Bull contribute most of the aggregate. The conclusion is confined to this selected corpus, not superiority on every task (Appendices R, T.1).

X.3 Whole-instruction revision improves the base policy

For zucchini, policy text and the prompt bank are revised while world and weights stay fixed. The base prompt changes from "Set the green squash straight on the patterned cloth" to "Set the green squash on the blue plate," changing modifiers as well as the target noun. In the separate 40-pair confirmation, strict base-policy success rises from 11 to 26; all 15 discordant pairs favor revision ($p = 6.10 \times 10^{-5}$). The paired gain is 37.5 points (Tango 95% interval [24.2,53.0]). Reconstructed CoVer changes from 9 to 14 ($p = 0.267$), and rephrase-finetuned π_0 from 21 to 26 ($p = 0.383$). Neither latter contrast establishes a gain, and the intervention does not isolate the noun's effect (Table 40).

Y Evidence Map and Supplementary Comparisons

The following displays connect the main claims to their supporting comparisons. They retain the detailed evidence map, teacher/student stratification, and instruction-label endpoints, while distinguishing directly recorded placement from historical reconstruction. This organization makes the relevant task reference, experimental unit, and measurement provenance explicit for each result.

The primary placement evidence is the 150-run direct-recording study. Historical reconstructed geometry is documented separately because its original rotations and contacts are not fully observed; Appendix L specifies the resulting limits on comparison.

Y.1 Retained decision rules

Primary paired label comparisons use the two-sided exact McNemar test; success-rate intervals use the Wilson 95% method. The primary MINERVA effect is $Y_{\text{hit}} - Y_{\text{persist@10}}$. Its prespecified criteria are a drop of at least 2 percentage points overall or 5 within a suite, and exact $p < 0.05$. A deterministic 10,000-sample task-cluster bootstrap resamples the 40 tasks while preserving their episodes. The zucchini confirmation requires at least eight discordant pairs, net improvement in both base and reconstructed CoVer, and at least one exact $p < 0.05$.

Table 66: The present study links task declarations to policy inputs, data, rewards and runtime checks. The columns compare this evidence with the closest validity and monitoring analyses, without excluding other uses of the prior methods.

Prior work	Evidence added here	Remaining limits
Jiang et al. (2026): shortcut, significance, overfitting and data-source diagnostics	Policy-input routing and conflicts between the declaration, data, rewards and behavior	Selected cases; no detector recall
SafeManip (Huang et al., 2026): finite-trace temporal-safety monitors	Asset bias in a placement evaluator; direct native/geometric comparisons; effects of native query timing	Operational proxy; shadow cache only
MANIGUARD (Peng et al., 2026): shared physics-grounded monitoring for demonstrations and fine-tuning	Reward hits and parameter changes distinguished from matched-data local adaptation	Conditional-label effect unresolved

Table 67: Native-success selection does not ensure calibrated whole-cup placement, particularly for the white cup. The 2-mm geometric final-50 criterion includes support and absence of gripper contact. Teachers were selected by native success and scored on saved pre-action suffix states; students use directly recorded post-action states. Seeds share 25 starts per color. New executions include no original-data controls.

Task	Teacher quality	Student seed 0	Seed 1	Seed 2
Red cup	36/46 (78.3%)	11/25 (44%)	9/25 (36%)	9/25 (36%)
White cup	6/49 (12.2%)	3/25 (12%)	7/25 (28%)	2/25 (8%)

Reported p-values are unadjusted across the study. A post-hoc two-test Bonferroni correction retains the base instruction-revision result ($p = 1.22 \times 10^{-4}$), without changing the original decision criteria. Episode-level tests condition on the selected tasks; different caddy starts are not independent task replications. Because persistence requires a first hit, we interpret the drop using its magnitude and task-cluster interval.

Table 68: The main claims rely on distinct evidence units and support distinct inferences. Each row states the task reference, measured outcome, sample and evidence source; full changed/fixed stage settings are in Table 41.

Task reference and comparison	Outcome / sample	Evidence and source	What the comparison shows
Caddy declaration vs runtime; same trajectories	Native goal hit; 100 demos, 100 starts/model	Table 9; recorded labels, source routing	Declaration/runtime conflict
Caddy learning; matched within-stage	Reward hits: 240 rollouts/arm; native first-hit: 40 QueST / 50 Qwen starts/model	Figure 10A, Table 14; reward/tensor/label logs	No rewarded compartment samples; demonstrations enable local adaptation
Caddy placement; frozen models	Native/geometric final-50 at 2 mm; new 150 executions, 50 reused starts	Figure 10B, Table 45: directly saved geometry/contacts	Gap in the same executions; no new original controls or validation of historical rotations
Dual-instruction / fixed-target labels	Qualified final-50 / joint; 20 starts/seed, 20 fixed-target executions	Table 69, Appendix T.4; directly saved geometry/contacts	Limited single-target completion; instruction-label gain unresolved
MINERVA native goal; continue same policy	First hit / retention@10; 40 tasks $\times$ 50 starts	Table 20; recorded predicates, 0.50-s continuation	Policy-continuation retention, not passive stability
SIMPLER text; motion/grasp/window	Five-state completion; 7 tasks, 140 paired starts	Table 70; logs plus four motion reconstructions	CoVer retains its gain on this corpus after ranking removal
Zucchini text; fixed world/weights	Strict completion; 40 pairs/policy	Section X.3, Table 40; recorded labels	Whole-instruction base-policy gain
RoboCasa goal; vary reads	Full monitor; 12 source counters, two native cases	Appendices I, S.4; mocked source tests / native reward–Gym path	Caching at a stated boundary preserves one-read history; trajectory frequency unmeasured

Table 69: Qwen's exploratory label experiment. A uses coherent labels; C uses permuted labels. Both instructions use the same 20 starts per seed. Triples count native target hits $\rightarrow$ qualified hits $\rightarrow$ qualified success throughout the final 50 steps. Joint counts starts that succeed under both instructions. Exterior qualification requires native `On` without gripper contact; compartment qualification requires floor-calibrated geometry at 2 mm with support and no gripper contact.

Arm	Seed	Exterior triple	Compartment triple	Joint
A	0	$16 \rightarrow 16 \rightarrow 10$	$2 \rightarrow 1 \rightarrow 1$	0
A	1	$17 \rightarrow 17 \rightarrow 16$	$3 \rightarrow 3 \rightarrow 2$	2
A	2	$12 \rightarrow 12 \rightarrow 11$	$6 \rightarrow 2 \rightarrow 1$	1
C	0	$11 \rightarrow 11 \rightarrow 6$	$6 \rightarrow 5 \rightarrow 2$	1
C	1	$13 \rightarrow 13 \rightarrow 9$	$2 \rightarrow 0 \rightarrow 0$	0
C	2	$14 \rightarrow 14 \rightarrow 12$	$4 \rightarrow 1 \rightarrow 0$	0

Table 70: SIMPLER results on 140 paired starts from seven selected tasks. No ordering@5 removes displacement ranking after the original analysis, retaining source motion $>$ 3 cm, no detected qualifying two-finger grasp, and five consecutive states. The last columns compare CoVer with base in percentage points, with marginal Tango 95% intervals and unadjusted two-sided exact McNemar p-values. Appendix T.1 reports paired counts and each task's contribution.

Policy	Position hit	Frozen strict@5	No ordering@5	Frozen Δ [95% CI]; p	No-order Δ [95% CI]; p
Base π_0	50 (35.7%)	40 (28.6%)	43 (30.7%)	–	–
Released CoVer	79 (56.4%)	61 (43.6%)	62 (44.3%)	15.0 [5.4,24.6] 0.003802	13.6 [4.3,22.8] 0.006609
Reconstructed CoVer	74 (52.9%)	65 (46.4%)	65 (46.4%)	17.9 [8.2,27.4] 0.000621	15.7 [6.0,25.3] 0.002602