BEYOND REPEATABLE SCORES: EXPERIMENTAL FIDELITY AUDITING IN ROBOT BENCHMARKS

Anonymous authors
Paper under double-blind review

ABSTRACT

A repeatable benchmark score is meaningful when its scenes, attempts, and observations preserve the declared experiment. Yet aggregation can conceal changes to these units, masking a different evaluation. We study this gap through *experimental-unit fidelity* and trace how execution and reporting can alter the declared units. This perspective leads to EFA-REPLAY, which aligns records with declarations, tests evaluator changes with matched controls, and determines which results the evidence supports. Paired-label accounting exposes changes hidden by aggregation, while sharp compatible-record bounds identify what reduced records preserve. In an ACT evaluation, scene count and 80% success remain unchanged despite 20% request-label disagreement. For SimplerEnv, exact pooled success leaves condition variance unidentified in 64/78 cells. A separate LIBERO audit flags a stale first observation before scoring; correcting it changes ACT task-8 success from 11/30 to 27/30 on frozen follow-up states. These results motivate preserving the unit-level evidence that gives benchmark scores their meaning.

1 INTRODUCTION

Robot benchmarks guide checkpoint selection and claims about generalization and simulation fidelity (Li et al., 2025; Zhou et al., 2025; Choi et al., 2026; Chen et al., 2026). Their meaning depends on how an evaluator assigns scenes, creates attempts, and groups observations. Shared datasets and generalist policies extend these choices across downstream comparisons (Open X-Embodiment Collaboration et al., 2023; Khazatsky et al., 2024; Octo Model Team et al., 2024). Score reproducibility is therefore important, but it leaves a complementary question: **does the repeated computation preserve the experiment and the units its score represents?**

Consider an ACT evaluation with five workers: both original and repaired runs contain twenty distinct scenes and report 16/20 successes, yet four request labels change. Episode count, scene uniqueness, and aggregate success agree while request-to-scene correspondence changes. After execution, SimplerEnv similarly rebuilds binary pseudo-trials from aggregate success, discarding initial-condition grouping. Figure 1 connects these losses of scene identity and condition structure.

We call this property *experimental-unit fidelity*. The mechanisms act at different stages, but both weaken the link between a summary and its underlying units. Opposing label changes cancel when averaged; many-to-one transformations erase grouping. This suggests a common analysis: pair outcomes behind each score, then determine which statistics the retained evidence identifies. Sharp ranges expose the ambiguity when different grouped records remain compatible with that summary.

EFA-REPLAY turns this perspective into a claim-specific audit, binding declarations to observed units, assessing evaluator changes with matched controls, and selecting evidence-supported recovery. The requirements depend on the conclusion: a record may identify pooled success while leaving condition variation unknown. Accordingly, the audit separates whether an operation violates the declaration, whether its effect is supported, and which results can be recomputed, reused, or replayed.

Across controlled SmolVLA and ACT executions and native checkpoint comparisons, unit alignment reveals changes hidden by aggregate agreement; the 78-cell SimplerEnv analysis shows which statistics survive reporting. The LIBERO audit detects an observation-timing error without success labels and tests its impact; ablations and archived replay test audit and recovery decisions.

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.

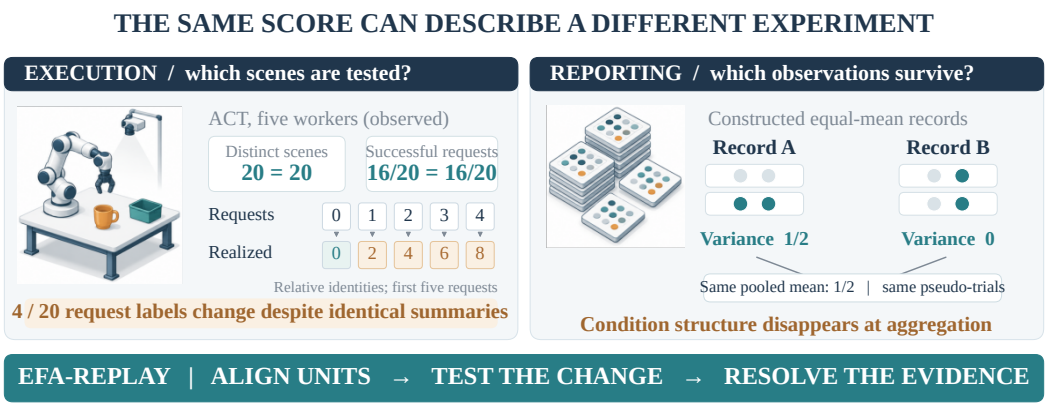


Figure 1: The same score can describe a different experiment. Left: observed five-worker ACT results preserve scene count and total success while four request labels change; relative identities illustrate the first five original requests. Right: constructed grouped records share a mean and pseudo-trials but differ in condition variance. Teal and gray denote success and failure. The two mechanisms act during execution and reporting, connecting the declared experiment to the records checked by EFA-REPLAY throughout the chain.

Our contributions.

- **Mechanisms:** we establish scene-allocation and observation-reconstruction failures that change experimental units while passing checks of counts, uniqueness, or aggregate success.
- **Quantification:** paired-label accounting and sharp per-cell identification ranges reveal changes hidden by aggregate scores and characterize the statistics that retained records determine.
- **Executable methodology:** EFA-REPLAY connects declarations to records, independently flags a LIBERO timing mismatch, and checks the evidence supporting recomputation, reuse, and replay.

2 RELATED WORK

Reproducibility and experimental units. Reproducibility programs expose implementations and variation (Pineau et al., 2021; Dodge et al., 2019), while RL evaluation methods quantify run-to-run uncertainty (Henderson et al., 2018; Agarwal et al., 2021). Interpreting either requires distinguishing experimental units from repeated measurements, the central concern of pseudoreplication (Hurlbert, 1984). Robot platforms standardize the tasks and interfaces producing these measurements (Mandlekar et al., 2022; Liu et al., 2023; James et al., 2019; Zhu et al., 2020; Nasiriany et al., 2024; Tao et al., 2024; Chen et al., 2025). We examine how evaluator operations preserve those declared units.

Benchmark validity and retained evidence. Estimands specify populations, outcomes, and aggregation (Binette & Reiter, 2024); datasheets, model cards, BetterBench, and BenchRisk document evaluation assets, uses, and risks (Gebru et al., 2021; Mitchell et al., 2019; Reuel et al., 2024; McGregor et al., 2025). Rollout Cards retain evidence and reporting views (Masters et al., 2026), while ABE-Ralph checks execution against reference-method contracts (Yu et al., 2026). Together with robot audits of shortcuts, significance, and redundancy (Jiang et al., 2026; Navasardyan & Davtyan, 2026), these efforts motivate operation-level checks. We trace request-to-scene and outcome-to-group mappings, quantify their information loss, and test evidence-supported recovery.

Identification and intervention-based checking. Partial identification separates information supplied by observations from assumptions selecting among compatible explanations (Manski, 2003; Imbens & Manski, 2004). We apply this perspective to finite records, bounding statistics over grouped outcomes consistent with the retained representation. Provenance and dependency analysis supply the lineage machinery (Buneman et al., 2001; Sculley et al., 2015), while metamorphic and causal testing supply intervention-checking principles (Eniser et al., 2022; Valle et al., 2026; Xu et al., 2026; Foster et al., 2024; 2025). EFA-REPLAY connects these foundations through robot evidence and claim-specific recovery; Appendix E.3 compares their inputs and outputs for the studied evidence.

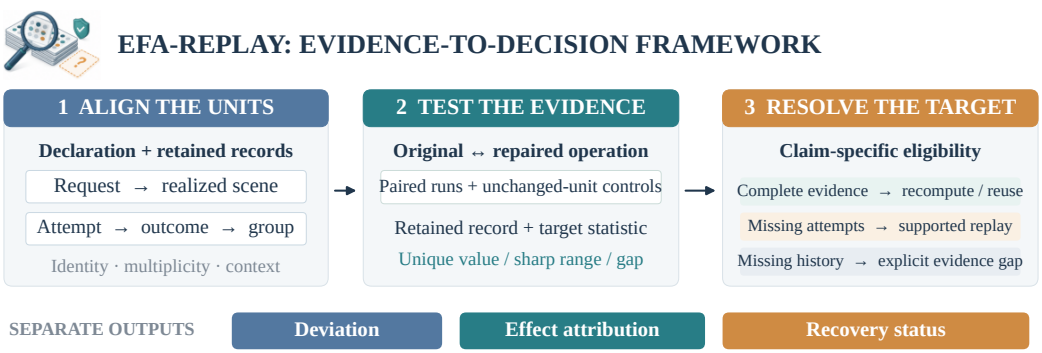


Figure 2: A claim-specific audit connects identity, evidence, and recovery. Alignment binds declarations to execution and reporting records. Matched controls assess evaluator changes, and compatible records characterize retained information; eligibility and registered boundaries then determine the action. Outputs distinguish the deviation, supported effect, and recovery decision, linking each to its own evidence requirements.

3 METHOD

3.1 DECLARED UNITS AND TWO AGGREGATE BLIND SPOTS

The declaration-to-report chain. Experimental fidelity auditing first specifies what a score summarizes. Requested unit i, denoted u_i, fixes task, scene, policy, attempt rule, and exogenous conditions. Execution realizes scene $s_i^\star$, attempt/randomness instance a_i, outcome Y_i, observation group g_i, and report unit r_i:

$$u_i \longrightarrow s_i^\star \longrightarrow a_i \longrightarrow Y_i \longrightarrow g_i \longrightarrow r_i. \tag{1}$$

A declaration D fixes target units, permitted replacements or retries, groups, weights, denominator, and context. Checking operations against D accommodates legitimate replacement, best-of-k sampling, and variant averaging while testing their correspondence to the declared population.

Cancellation hides changed outcomes. To distinguish net change from disagreement, pair original/repaired labels $Y_i^O, Y_i^R \in \{0,1\}$ by requested identity. Let n_{01}, n_{10} count each direction of change among N requests. The net success-rate change Δ and changed-label fraction D_{labels} satisfy

$$\Delta = \frac{n_{01} - n_{10}}{N}, \qquad D_{\text{labels}} = \frac{n_{01} + n_{10}}{N}, \qquad D_{\text{labels}} - |\Delta| = \frac{2\min(n_{01}, n_{10})}{N}. \tag{2}$$

The final term is exactly the disagreement canceled by averaging: even a zero score change can conceal substantial movement in which requests succeed. Pairing the labels recovers this distinction.

Reduction hides observation structure. Reporting can also replace grouped outcomes by a binary reconstruction. For $N > 1$ condition means $x_i \in [0,1]$, let $p = N^{-1}\sum_i x_i$; a reconstruction of length M has mean $q = \text{round}(Mp)/M$. When $M = N$ and $q = p$, its variance s_z^2 and the condition-mean variance s_x^2 obey

$$s_z^2 - s_x^2 = \frac{1}{N-1}\sum_{i=1}^{N} x_i(1 - x_i) \geq 0. \tag{3}$$

The gap persists under exact mean and count preservation: within-condition disagreement becomes a different empirical distribution. Appendix B.2 adds the effects of rounding and unequal sample sizes.

3.2 IDENTIFYING WHAT A REDUCED RECORD DETERMINES

Compatible records. When full records survive, a distorted statistic can be recomputed. Otherwise, recovery depends on what the reduced representation z determines under its allowed count and grouping constraints H. For a target statistic T, define

$$\mathcal{L}(z, H) = \{L' : L' \text{ satisfies } H,\ \text{Rep}_H(L') = z\}, \qquad \mathcal{I}_T(z, H) = \{T(L') : L' \in \mathcal{L}(z, H)\}. \tag{4}$$

A singleton $\mathcal{I}_T$ identifies the target; two compatible records with distinct target values establish non-identifiability. When neither conclusion has been established, the audit retains an unresolved status. For a scalar statistic, the sharp identification range is

$$[\min \mathcal{I}_T(z, H),\ \max \mathcal{I}_T(z, H)]. \tag{5}$$

Compatible records attain the endpoints, giving the exact extremes permitted by the evidence. Bounds use each target's own cell and constraints H; Appendix B.7 provides the endpoint constructions.

Attainable variance bounds. To bound the variation hidden by reduction, consider N conditions with k binary outcomes each. If condition i has s_i successes and $\sum_i s_i = S$, the condition-mean variance is

$$V(S, s) = \frac{\sum_i s_i^2 - S^2/N}{(N-1)k^2}, \qquad 0 \le s_i \le k. \tag{6}$$

For fixed S, balanced counts minimize the squared sum, while concentrating successes into full groups maximizes it. For pseudo-trials, we enumerate every integer total S reproduced by the pinned helper and take extrema over feasible totals. The allocations witness both bounds and information loss; Appendix B.7 gives the proof and exact rounding treatment for the released reconstruction.

3.3 From unit alignment to supported recovery

Align and test. Each adapter emits a `UnitRecord` binding the chain in Eq. (1) to the recorded context and provenance. Shared checks detect substitutions, collisions, omissions, and altered accounting. The contracts cover initial-state validity, attempt isolation, scorer non-interference, and result-population accounting; the latter and attempt isolation govern our main cases. Matched repair comparisons $b \to b'$ fix population, policy, unrelated operations, and randomness. Trajectory repeats and unchanged-scene controls support attribution; complete accounting supports record analysis.

Reuse follows the generating evidence. Repairing an operation also requires deciding which earlier results remain eligible for the target. The recovery inputs are D, retained records L, supported dependency edges E, changed records A_0, and an adapter-supplied hierarchy of registered replay boundaries. Starting from A_0, the shared core propagates invalidation to $A_\star$ and selects eligible records:

$$\begin{aligned} A_{t+1} &= A_t \cup \{w : \exists v \in A_t, (v, w) \in E\}, \qquad A_\star = \bigcup_{t \ge 0} A_t, \\ \mathcal{R} &= \{r \in L : P_D(r) \text{ passes},\ r \notin A_\star\}. \end{aligned} \tag{7}$$

Here P_D checks applicable identity, attempt, source, context, grouping, and provenance, followed by population-level coverage and multiplicity. Complete evidence permits recomputation with the original grouping and denominator. For partial records, eligibility is checked before missing outcomes are regenerated; a separately executed reference then validates the completed target. Missing or invalid attempts expand to the first supported boundary on the registered hierarchy, specifying regeneration under the required initialization and source conditions. The outputs link each permitted recovery to its supporting evidence and identify any remaining gap in the declared target.

Requirements follow the target: labels can identify pooled success even when missing context prevents trajectory reuse. A missing attempt may instead be regenerated at a registered fresh-process boundary. When historical identities are absent, new collection defines a fresh evaluation target. Figure 2 links these decisions to the declarations and boundaries in Appendices C.2 and D.2.

4 Experiments

4.1 Experimental setup

Complementary evidence sources. We combine fixed-policy execution, saved-record analysis, and diagnostic checks (Appendix A). Controlled SmolVLA uses two tasks, sixteen scene IDs per task, two policy-noise seeds, and paired original/repaired execution under calibrated rendering (Shukor et al., 2025). ACT extends the test to two task-specific checkpoints with native cameras, normalization, and action queues, at widths four and five (Zhao et al., 2023). The native pool covers five task/checkpoint combinations over two fifty-request cohorts. Eight of ten planned comparisons pass the controls; Appendix A.3 retains the full ledger and both descriptive exclusions with their controls.

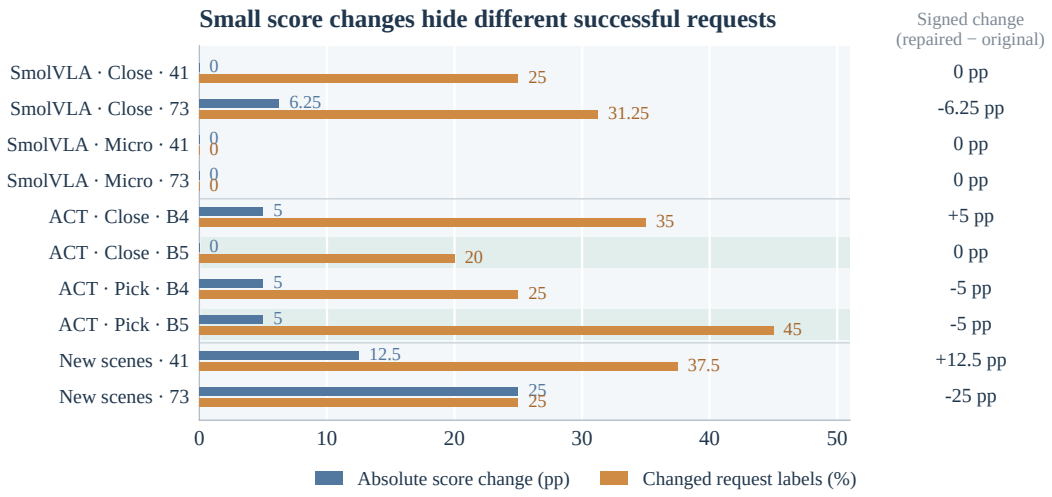


Figure 3: Small score changes can hide substantial request-level disagreement. Blue shows $|\Delta|$ in pp; amber shows changed requests in %, with signed Δ (repaired minus original) at right. All controlled, ACT-transfer, and additional-scene configurations are included, with every zero-change result retained. Close and Micro mean CloseFridge and OpenMicrowave. Each row pairs requested identities; widths and noise strata share scenes.

Pairing and analysis units. Labels are paired by requested identity within task, checkpoint, cohort, width, and noise stratum. The first formal execution supplies scores; the second verifies labels and changed-request identities. We run the pinned SimplerEnv helper on all 78 cells, preserving both resolutions of one conflicting record (Li et al., 2025; xuanlinli17, 2024), and analyze reductions of these saved outcomes post hoc. The summaries describe fixed, paired records with shared requests, repeated widths, and overlapping cells across the retained configurations (Appendices A–B).

4.2 Aggregate agreement conceals changed experimental units

The execution mechanism. At the audited LeRobot revision, per-slot seeds receive a second worker-index offset, so request $s + i$ executes scene $s + i + (i \bmod B)$ at width B. Fifty requests yield 26 distinct scenes with four workers; nine workers preserve fifty scenes but substitute 44 identities. Five-worker ACT exposes the subtler case: both arms contain twenty distinct scenes and 16/20 successes, yet four labels change. The mapping differs beneath identical counts and aggregate outcomes.

Paired outcomes reveal the cancellation. Paired outcomes expose cancellation in these comparisons (Figure 3): controlled SmolVLA changes 9/64 labels (14.06%) while successes move from 18/64 to 17/64 (-1.56 pp); all 128 same-arm and 32 unchanged-scene controls pass. Five-worker ACT pick/place changes 9/20 labels as success moves from 7/20 to 6/20. Its 45% disagreement and -5 pp net change describe the same records through the terms in Eq. (2). A separately frozen cohort reproduces five changes in sixteen scene/noise blocks across eight new scenes with the same checkpoint.

The reporting mechanism. In SimplerEnv, the released helper averages variant success and rebuilds a rounded binary vector. The resulting pseudo-trial variance exceeds condition-mean variance in 64 of the 78 cells, is smaller in one, and agrees in thirteen (Figure 5a). The exception is informative: Drawer Open/Octo-Base contains one success in 108 outcomes, which becomes zero successes in 27 pseudo-trials. Figure 5b displays this loss. Together with Eq. (3), it separates a structural change that persists under exact mean preservation from the additional distortion introduced by rounding; Appendix B.2 retains both conflict resolutions and every comparison cell from the public records. The pattern remains after removing aggregate rows: 52 of 66 component comparisons increase, one decreases, and thirteen are unchanged (Appendix B.2).



Figure 4: Unit alignment connects fixed-policy score shifts to execution evidence. (a) Original and repaired scores in eight control-admitted comparisons, with fifty requests per arm and signed shifts at right; Appendix A.3 retains both exclusions. (b) Frozen forecasts use one corrected scene ledger, queried by reported or realized identity, across eighty comparisons sharing forty task/scene IDs. Lower panel: the post-hoc decomposition separates scene reweighting from the remaining native-outcome residual in the admitted cohort-A comparisons.

4.3 Unit alignment connects mechanisms to policy outcomes

Fixed-policy score impact. Native comparisons quantify how much the evaluator changes a fixed checkpoint's reported performance. The eight control-admitted contrasts range from -12 to $+16$ pp (Figure 4a). Although the first-cohort task and macro winners are preserved, the ACT–multitask macro margin grows from 21 to 26 pp. The evaluator thus changes the margins used to interpret relative performance, even when it preserves the identity of the winning policy checkpoint.

Forecasting from the same scene ledger. To connect the source mechanism to subsequent outcomes, we freeze ACT label forecasts from a corrected four-slot ledger before the original executions. Both forecasts query that same ledger; changing the lookup from reported seed to source-derived effective identity reduces errors from 27/80 to 3/80 (Figure 4b). The remaining errors show why a scene mapping alone is insufficient for trajectory reuse across batch shapes. Consistently, the reference decomposition leaves a six-success residual in two CloseFridge SmolVLA comparisons after scene reweighting. This post-hoc diagnostic motivates checking the generating context in addition to identity; the matched execution controls determine which full-pipeline effects support attribution.

Independent observation-timing audit. The context requirement also applies to observation timing. In LIBERO's standalone evaluator (Liu et al., 2023), five zero-action settling steps leave the first

Table 1: A LIBERO timing violation is detectable before scoring and changes task-8 success. Stale/current denote pre-/post-settling inputs; success counts (↑) pair the same physical initial states. Discordance lists stale failure/current success and the reverse. The 20 exploratory states and 30 frozen follow-up states are separate stages. These single-checkpoint ACT adapter results are not official BC-CLI scores.

Task / ACT checkpoint	Stage	Stale	Current	Discordant
Task 8 / public ACT	Exploratory	6/20	**17/20**	11 / 0
Task 8 / public ACT	Frozen follow-up	11/30	**27/30**	17 / 1

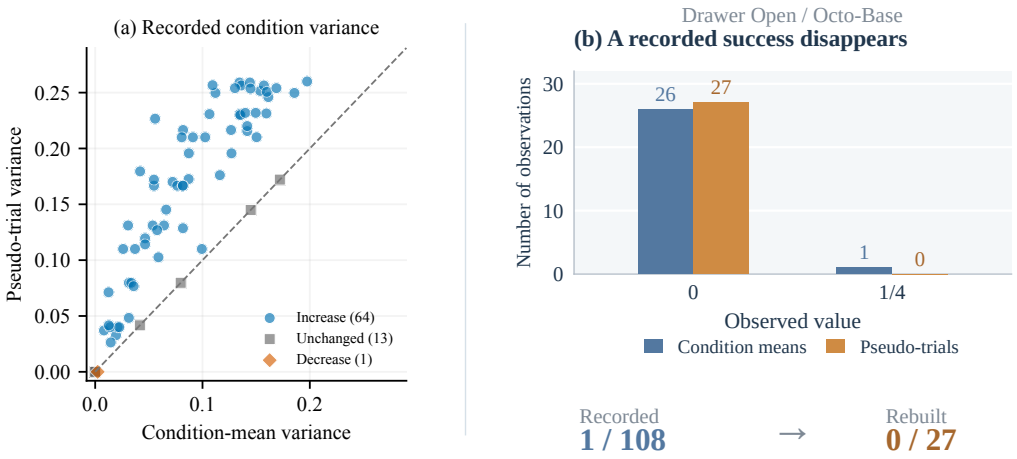


Figure 5: Record reduction changes the distribution even when its mean barely moves. (a) All 78 variance comparisons: 64 increases, one decrease, and thirteen equal values, including overlap. (b) Drawer Open/Octo-Base has 26 zero condition means and one mean of $1/4$, from one success in 108 outcomes; its reconstruction contains 27 zeros. Both resolutions of the separate Move Near duplicate conflict preserve these comparisons.

policy input at the pre-settling observation. EFA's context check flags every stale trace and no corrected trace without using success labels. Changing only that input while pairing post-settling physical states raises task-8 ACT success from 11/30 to 27/30 on a frozen follow-up (Table 1). The native-environment adapter reproduces the standalone entry's timing for one public checkpoint; its results are not official BC-CLI scores (Appendix A.5).

4.4 Reduced records identify averages, but not their structure

Four representations, three targets. To determine what survives reporting, we retain four representations for each SimplerEnv cell: full grouped outcomes (R0), ungrouped outcomes (R1), exact mean and counts (R2), and released pseudo-trials (R3). The common header fixes outcome count, condition count, equal group size, and pseudo-trial length. Targets are original success count and mean (T1), the condition-indexed table (T2), and condition-mean variance (T3). Isolated processes read only their allowed input; a separately implemented internal verifier checks target values and witnesses.

Exact averages can coexist with missing structure. R1 and R2 preserve pooled success in all 78 cells, yet identify only 9 condition tables and 14 condition variances (Figure 6a). Thus grouping loss alone leaves variance non-identifiable in 64/78 cells, even before rounding enters the analysis. R3 further reduces the identifiable original success counts to eight cells. The distinction is one of information, not simply numerical accuracy: a proxy may match the original statistic while another admissible grouped record with the same visible representation yields a different target value.

The different counts for T2 and T3 clarify why recovery is target-specific. A condition-indexed table specifies which initial state carries each success count, whereas its variance depends on the counts' squared sum. Permuting counts can change the table while preserving variance. Thus 14 variances remain identified from ungrouped outcomes even though only 9 tables do. The audit recovers values fixed by the retained representation while preserving ambiguity in condition assignments; the same evidence can support a coarser statistic more strongly than its condition-indexed table.

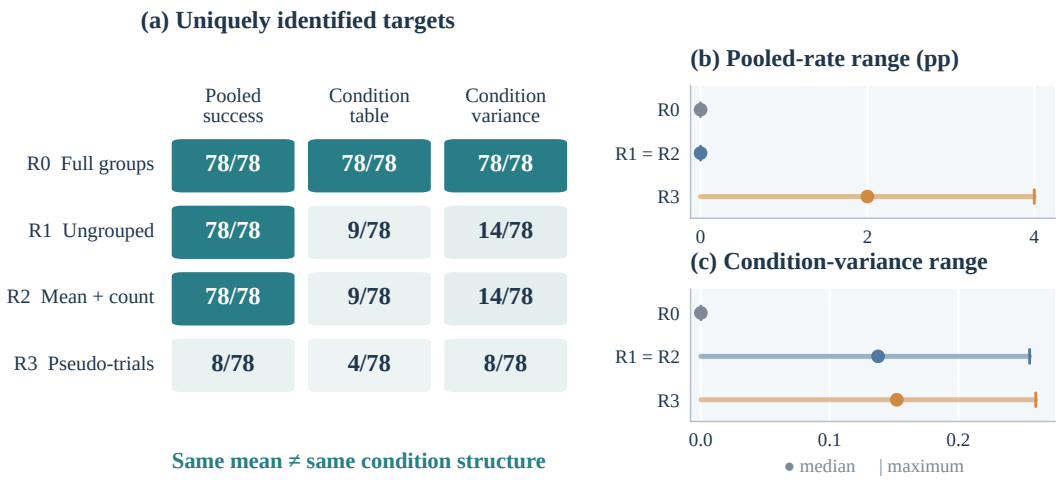


Figure 6: Preserving pooled success can leave condition-level evidence undetermined. (a) Uniquely identified targets among all 78 cells; the remaining targets have compatible counterexamples. (b,c) Median (dot) and maximum (end mark) of sharp identification-range widths. Both conflict resolutions agree. Ranges use each cell's retained representation and count constraints, with feasible records attaining the endpoints (Section 3.2).

Table 2: Unit-level evidence changes audit decisions, and each component has a distinct role. Top: fixed diagnostic counts, with B3 as the internal paired oracle; bold marks tied best values. Bottom: natural-record removals test scene/group checks; constructed diagnostics test context and propagation in their respective units. Preset flags define diagnostic controls; measured execution controls support policy-effect attribution in the paired policy studies.

Audit method	**Detected / 6 ↑**	**False flags / 5 ↓**	**Deferred / 1 ↑**
B0: repeat / total	0	**0**	0
B1: nominal IDs	0	**0**	0
B2: full-log assertions	5	**0**	**1**
B3: paired oracle	**6**	**0**	**1**
EFA-REPLAY	**6**	**0**	**1**
Removed component	**Observed consequence**		**Count**
Actual-scene binding	Missed scene-allocation cases		2
Observation grouping	Missed record-structure case		1
Execution context	Unsafe reuse decisions		2
Dependency propagation	Stale downstream records retained		2

Sharp ranges quantify what remains unknown. R1/R2 condition-variance widths have median 0.138 and maximum 0.255. R3 pooled-rate widths have median 2 pp and maximum 4 pp, alongside variance widths of 0.152 and 0.260 (Figure 6b,c). The targets use different scales, but both expose the distinction between a preserved scalar average and uncertain condition structure. Zero-width cases and the six Eggplant cells with unequal condition/pseudo-trial counts remain included. Appendix B.7 supplies every endpoint witness, providing concrete grouped records that attain the reported extremes.

4.5 AUDIT DECISIONS AND COMPONENT ABLATIONS

Which evidence changes the decision? The diagnostic suite tests unit-level information through six deviations, five declared legal configurations, and one insufficient input. B0 checks repeats and totals; B1 adds nominal IDs; B2 checks scene, group, and context records; B3 receives paired candidate records with the same controls as EFA-Replay. Full-log assertions detect five deviations, while the paired oracle and EFA-Replay detect all six and preserve the insufficient-evidence verdict (Table 2). Declared replacement, best-of-two, and variant averaging serve as explicit legal controls.

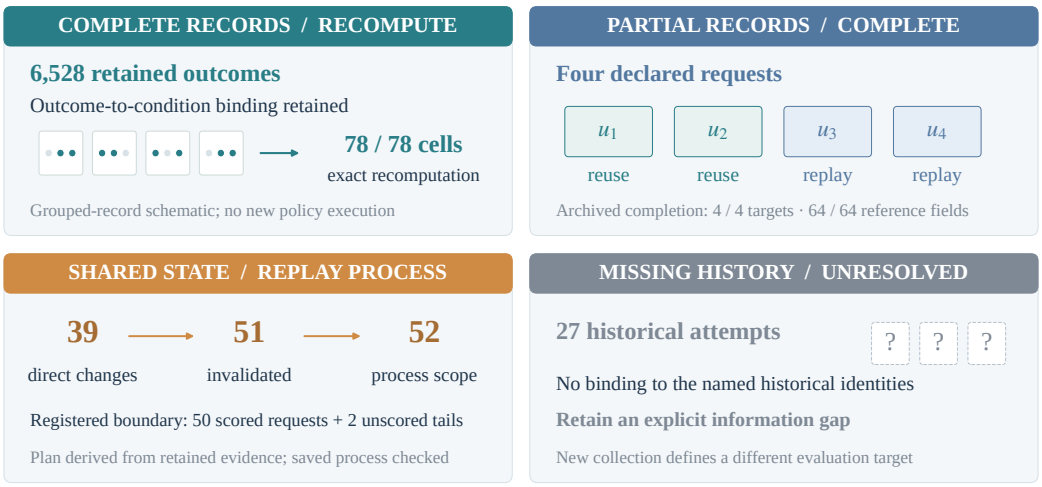


Figure 7: Recovery scope follows the evidence. Complete records support recomputation, and the archived partial case combines two saved episodes with two selective executions. For shared state, the conservative dependency closure expands to a 52-record process, with the plan checked against saved execution. Missing identities leave 27 historical attempts unresolved. Four separately executed reference episodes validate the partial result; their cost is recorded separately from the two selective executions that complete the target.

From detection to a justified action. The removals clarify each component's role: scene binding and grouping expose natural-record deviations, while context and propagation prevent unsafe reuse. Together, these checks avoid the registered errors. Recovery then depends on the retained evidence: twelve inputs yield two reuses, three supported replays, and seven evidence gaps. Appendix C.3 gives every proposition and decision, complemented below by richer records for validating recovery.

4.6 Evidence-supported recovery in practice

Complete records and selective completion. Recovery actions depend on the retained evidence (Figure 7). Full SimplerEnv records support recomputation of 78 cells from 6,528 outcomes. In the archived four-request case, EFA-Replay retains two eligible episodes and executes the missing pair in registered fresh processes. The plan precedes selective execution; its receipt precedes separate references. All 64 field comparisons match, including labels $(0, 0, 0, 1)$. The target needs two additional episodes versus four for a whole-target plan; Appendix D.2 details eligibility, execution, and validation.

The generating context determines the remedy. Shared state can require process-wide repair. A conservative chain expands 39 changed records to 51, and the supported boundary covers all 52, including two unscored tails. The remaining candidate is regenerated with the process. Missing historical identity calls for a different response: an aggregate may determine a statistic, but named attempts require their original binding. New collection instead defines a fresh evaluation population.

Evidence requirements for reporting. Appendix Table 11 summarizes target-specific retention requirements; Appendices D.3–E.2 provide independent checks and supplementary zero-effect cases.

5 Conclusion

Scene allocation and record reduction can change the experiment behind a benchmark score even when aggregate checks agree. Pairing requested outcomes reveals changes that cancel in a score, while compatible-record bounds expose condition structure that an average leaves undetermined. The independent LIBERO audit detects an observation-timing mismatch before scoring and validates its outcome impact in the reported task-8 setting. EFA-Replay connects these checks to evidence-supported recomputation, reuse, and replay. Reliable evaluation requires preserving and validating the scenes, attempts, and observations behind each score.

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

R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *NeurIPS*, 2021.

O. Binette and J. P. Reiter. Improving the validity and practical usefulness of AI/ML evaluations using an estimands framework. *arXiv preprint arXiv:2406.10366*, 2024.

P. Buneman, S. Khanna, and W.-C. Tan. Why and where: A characterization of data provenance. In *ICDT*, 2001.

R. Cadene et al. LeRobot: An open-source library for end-to-end robot learning. *arXiv preprint arXiv:2602.22818*, 2026.

T. Chen et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

T. Chen et al. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026.

S. Choi et al. vla-eval: A unified evaluation harness for vision-language-action models. *arXiv preprint arXiv:2603.13966*, 2026.

J. Dodge, S. Gururangan, D. Card, R. Schwartz, and N. A. Smith. Show your work: Improved reporting of experimental results. In *EMNLP*, 2019.

H. F. Eniser, T. P. Gros, V. Wuestholz, J. Hoffmann, and M. Christakis. Metamorphic relations via relaxations: An approach to obtain oracles for action-policy testing. In *ISSTA*, 2022.

M. Foster, C. Wild, R. M. Hierons, and N. Walkinshaw. Causal test adequacy. In *ICST*, 2024.

M. Foster et al. The causal testing framework. *Journal of Open Source Software*, 10(107):7739, 2025.

T. Gebru, J. Morgenstern, B. Vecchione, J. Wortman Vaughan, H. Wallach, H. Daumé III, and K. Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. In *AAAI*, 2018.

Stuart H. Hurlbert. Pseudoreplication and the design of ecological field experiments. *Ecological Monographs*, 54(2):187–211, 1984. doi: 10.2307/1942661.

Guido W. Imbens and Charles F. Manski. Confidence intervals for partially identified parameters. *Econometrica*, 72(6):1845–1857, 2004. doi: 10.1111/j.1468-0262.2004.00555.x.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark and learning environment. *arXiv preprint arXiv:1909.12271*, 2019.

T. Jiang et al. What are we actually benchmarking in robot manipulation? *arXiv preprint arXiv:2606.04233*, 2026.

A. Khazatsky et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *RSS*, 2024.

X. Li, K. Hsu, J. Gu, O. Mees, K. Pertsch, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, S. Levine, J. Wu, C. Finn, H. Su, Q. Vuong, and T. Xiao. Evaluating real-world robot manipulation policies in simulation. In *CoRL*, 2025.

B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, 2023.

A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *CoRL*, 2022.

Charles F. Manski. *Partial Identification of Probability Distributions*. Springer Series in Statistics. Springer, 2003. doi: 10.1007/b97478.

C. Masters, Z. Liu, and S. V. Albrecht. Rollout Cards: A reproducibility standard for agent research. *arXiv preprint arXiv:2605.12131*, 2026.

S. McGregor et al. Risk management for mitigating benchmark failure modes: BenchRisk. In *NeurIPS*, 2025.

M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru. Model cards for model reporting. In *FAT**, 2019.

S. Nasiriany et al. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

Z. Navasardyan and H. Davtyan. A statistical audit of physical AI benchmark redundancy. *arXiv preprint arXiv:2608.25940*, 2026.

Octo Model Team et al. Octo: An open-source generalist robot policy. In *RSS*, 2024.

Open X-Embodiment Collaboration et al. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023.

J. Pineau, P. Vincent-Lamarre, K. Sinha, V. Larivière, A. Beygelzimer, F. d'Alché Buc, E. Fox, and H. Larochelle. Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program). *Journal of Machine Learning Research*, 22(164):1–20, 2021.

A. Reuel, A. Hardy, C. Smith, M. Lamparth, M. Hardy, and M. J. Kochenderfer. BetterBench: Assessing AI benchmarks, uncovering issues, and establishing best practices. In *NeurIPS*, 2024.

D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison. Hidden technical debt in machine learning systems. In *NeurIPS*, 2015.

M. Shukor et al. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

S. Tao et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied AI. *arXiv preprint arXiv:2410.00425*, 2024.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IROS*, 2012.

P. Valle, S. Segura, S. Ali, and A. Arrieta. Metamorphic testing of vision-language action-enabled robots. *arXiv preprint arXiv:2602.22579*, 2026.

G. Xu, D. Xiao, Y. Peng, and S. Wang. MetaSpace: Metamorphic testing for spatial cognition in embodied agents. *Proceedings of the ACM on Programming Languages*, 10(OOPSLA1):343–372, 2026.

xuanlinli17. simpler-env-eval-example-videos: Dataset repository, 2024. URL `https://huggingface.co/datasets/xuanlinli17/simpler-env-eval-example-videos/tree/610a6204d46cbc6fd9feb2c79d9a98117ce5bc22`. Revision 610a6204d46cbc6fd9feb2c79d9a98117ce5bc22.

L. Yu, X. Xu, Y. Zhou, S. He, and A. Pan. Beyond execution: Auditing experimental fidelity in LLM-driven scientific research. *arXiv preprint arXiv:2608.26753*, 2026.

T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Z. Zhou, P. Atreya, Y. L. Tan, K. Pertsch, and S. Levine. Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world. *arXiv preprint arXiv:2503.24278*, 2025.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

Table 3: Execution, record analysis, and diagnostics address complementary questions. The table identifies each registered evidence population and its primary role. Scene IDs, repeated execution, and record comparisons are counted separately within each study; O/R denotes the original and repaired evaluator arms.

Evidence source	Fixed analysis units	Primary question
Controlled SmolVLA	32 scene IDs; 64 scene/noise pairs	Operation-specific effects
ACT transfer	40 task/scene IDs; 80 width/request pairs	Identity versus duplication
Native model pool	10 planned cells; 8 control-admitted	Fixed-checkpoint score impact
SimplerEnv records	6,528 outcomes; 78 overlapping cells	Retained information
Diagnostic suite	6 deviations, 5 legal, 1 insufficient case	Decision requirements
Prospective recovery	2 saved + 2 selective + 4 reference runs	Supported completion

A Policy experiments, controls, and complete outcomes

This appendix develops the execution evidence behind the main results, from runtime calibration to paired-request analysis. Table 3 identifies the units used in each study; the protocols establish scene identity, policy behavior, and outcome agreement for admitting each score contrast.

A.1 Controlled-policy calibration

The controlled study holds the policy fixed so that the original and repaired evaluator paths can be compared under the same model and runtime. We use `lerobot/smolvla_robocasa`, revision `38d38bf5`, with base VLM and tokenizer snapshot `7b375e1b`, RoboCasa `56e355cc`, and robosuite `aaa8b9b2`. These dependency revisions identify this controlled study separately from the RoboCasa subject audit. The artifact binds the configuration to full commits and hashes for nine checkpoint files, 12 base-model files, 24,765 RoboCasa assets, and 942 robosuite assets.

Pre-scoring calibration uses CloseFridge seeds 2900–2903, policy-noise seed 41, both evaluator arms, and two fresh-process repeats. To make the supplied policy noise reproducible, each scene/noise pair uses key $10^6 p + s$, where $0 \le p < 4096$ and $0 \le s < 10^6$; generated tensors are checked for all 32 formal keys. Deterministic Torch algorithms, disabled TF32 and cuDNN benchmarking, and batch-boundary RNG control maintain the same randomness protocol in both arms.

Rendering is calibrated under the same correction in both paths: a backport of robosuite PR 790. OSMesa/llvmpipe uses one software-rendering thread per native forkserver worker and one BLAS/OpenMP thread, with GPU allocation confined to policy inference. The final calibration reproduces all eight same-arm and two unchanged-seed recorded trajectories. This establishes the runtime used for the formal intervention, while retaining calibration episodes as a separate population.

The controlled intervention removes the extra offset only when an explicit seed is supplied, preserving the same missing-seed fallback in both arms. Its trace records initial and post-step observations, supplied noise and action hashes, rewards, termination, and the $(\mathrm{time}, qpos, qvel)$ vector in the MuJoCo-based stack (Todorov et al., 2012; Zhu et al., 2020). These are the channels compared for recorded-trajectory agreement; hidden solver, controller, and RNG states remain outside that physical-state vector. State reads follow the adapter step, including internal autoreset when triggered, whereas observations and labels retain their upstream timing. Appendix A.2 examines this timing.

Across the 32 registered task/scene IDs, label differences averaged over the two noise seeds are positive for three IDs, negative for three, and zero for 26, giving mean $-1/64$. Every initial observation/state pair agrees across noise seeds in the 128 matched checks. These controls connect the paired-label result to a shared initialization using requested identity to pair the outcomes.

We also reproduce the public launcher from fresh LeRobot worktrees in an isolated Python 3.12.10 environment. All four calibration processes pass their controls, and their 16 recorded trajectories and labels match V5 exactly. This check varies source and dependency installation while sharing the simulator/model assets, rendering libraries, and GPU. These remain calibration executions.

A.1.1 SEPARATELY FROZEN NEW-SCENE CHECK

To test the mechanism on additional scenes, we fix CloseFridge seeds 50000–50007, noise seeds 41/73, both arms, and two process repeats before execution. The task follows the original cohort, while the checkpoint, prompt, cameras, 600-step horizon, and calibrated runtime remain unchanged. Each noise stratum uses a fixed GPU; repeat 1 runs original then repaired, and repeat 2 reverses that order. The eight new scene IDs yield sixteen scene/noise blocks and 64 executions, with their own protocol and execution records, separate from both the original cohort and calibration.

Noise seed	Original / 8	Repaired / 8	Discordant	Original/repaired scenes
41	2	3	3	6 / 8
73	4	2	2	6 / 8

Every label and discordance in the table recurs in the second repeat. All 32 within-arm trajectory comparisons, eight unchanged-effective-seed controls, and supplied-noise checks pass. The five discordant blocks span four IDs, with three original-only and two repaired-only successes, so total success changes from 6/16 to 5/16. Each original process contains six distinct initial observation/state pairs, compared with eight after repair; all same-effective-seed pairs and 32 cross-noise initial comparisons agree. The resulting evidence extends the mechanism to eight additional scenes under the fixed runtime. The protocol, five unchanged source hashes, per-process revisions, metric arrays, and traces accompany `summarize_new_scenes.py`, which regenerates the comparison.

A.2 PROSPECTIVE ACT TRANSFER PROTOCOL

Selection and execution freeze. The ACT study tests scene allocation with task-specific checkpoints selected from public integration PR 3217. The registry `data/act_transfer/models.json` pins CloseFridge revision `bda4cd6c` and PickPlaceCounterToCabinet revision `bd5fe44e`. Strict loading checks all 51,567,500 parameters in each checkpoint before simulator actions begin. Both policies retain their native three-camera inputs, normalizers, ten-action queue, and evaluation-time zero latent. The protocol, source references, and runners are hashed before any ACT rollout; `model_preflight.json` records model loading. Execution shares the calibrated SmolVLA host, simulator assets, and rendering correction used for the controlled policy comparisons.

Population and execution order. Each task first completes an excluded calibration at seeds 79–82 and width four, with both arms repeated in fresh processes. Exact within-arm trajectories and unchanged-effective-seed controls govern admission. Formal evaluation then uses requested seeds 87–106 at both widths. A corrected four-slot ledger covers seeds 87–110: its first twenty labels provide the registered comparator, and four additional scenes cover the original effective identities. The original width-four runs repeat twice; the width-five order is original 1, repaired 1, repaired 2, original 2, with CloseFridge preceding PickPlaceCounterToCabinet. In total, 336 formal and 32 calibration executions yield eighty task/width/request comparisons over forty task/scene IDs.

Forecasts fixed before observation. Once a task's two corrected ledgers agree, the runner writes and hashes its predictions before the first original formal execution. Both forecasts query the same ledger: request i uses $87 + i + (i \bmod B)$ for the audited identity and $87 + i$ for the reported identity; the repaired forecast likewise uses the requested identity. Individual labels, totals, and predicted discordant IDs are all fixed at this stage, including changes in both directions. Timestamps, ledger hashes, and prediction checks establish the ordering, so the later comparison measures how recovering effective identity changes forecast agreement on these fixed task/width/request comparisons.

Resource allocation. Continuation amendments permit availability-based assignment to matching memory-expanded RTX 4090 boards while preserving scientific order, native invocation, model/source versions, requested populations, forecasts, and controls. At most two same-stage processes run concurrently under recorded memory headroom; each is restricted to its assigned GPU, while renderer workers use CPU rendering. Launch timestamps, device UUIDs, capacity, and co-resident contexts are retained in `GPU_ALLOCATION_EVENTS.jsonl`. Together with `RESOURCE_CONTINUATION.md`, these records link shared allocation to the protocol.

Separating outcome agreement from trajectory agreement. The verifier checks labels against rewards, requested denominators, source/configuration identities, and renderer isolation. Same-arm

and unchanged-effective-seed comparisons within each width govern effect admission. Additional comparisons match initial states, labels, and trajectories within and across widths to examine scene-level transfer. This stronger check is useful because native batch shape can change action hashes, consistent with batched-versus-sliced behavior in PyTorch 2.7's numerical-accuracy documentation.

The adapter's timing explains a separate class of state differences: it returns the completed task's observation and reward, then internally resets before the trace reads simulator state. A final state hash can consequently belong to the next scene and depend on its worker index. A channel-timing diagnostic, added after the execution freeze but before original formal rollouts, distinguishes this terminal-only case from action or active-episode differences. Its self-check covers all three patterns, while the original controls and full comparisons remain intact. The frozen protocol, predictions, complete traces, and executable verifier are packaged together in `data/act_transfer/`.

Complete native results. All twenty-four native processes finish with exit zero. Both calibration gates pass, as do 168 formal within-arm repeat pairs and 36 unchanged-effective-seed controls. Every success total and discordant-ID set in Figure 3 recurs in the second execution. On the forty task/scene IDs reused across widths, repaired-label forecast errors are 0/20 and 2/20 for CloseFridge at widths four and five, and 0/20 at both widths for pick/place. The tables below pair these label results with the stronger trajectory checks under both the cross-width and same-width matching conventions.

Task	Slots	Initial	Strict	Native channels	Terminal only
CloseFridge	4	20	11	20	9
CloseFridge	5	20	0	0	0
Pick/place	4	20	15	20	5
Pick/place	5	20	0	0	0

The canonical comparison pairs each original effective scene with the corrected four-slot ledger, giving twenty pairs per row. "Strict" includes the full recorded trajectory and simulation-state channel. "Native channels" includes applied actions, returned observations, rewards, termination, step counts, labels, and matched initial observations/state. "Terminal only" identifies agreement on those native channels with the sole state difference occurring after the adapter performs its internal autoreset.

Task	Slots	Covered pairs	Strict	Native channels	Terminal only
CloseFridge	4	18	11	18	7
CloseFridge	5	18	6	18	12
Pick/place	4	18	14	18	4
Pick/place	5	18	13	18	5

The same-width comparison instead uses every effective identity covered by a corrected run at that width. It isolates scene/action correspondence within the existing records, complementing the requested-ID intervention. All individual checks, first divergent channels, and unrounded metrics for both conventions are retained in `result.json` and the ten-file-per-process trace records.

A.3 Complete native model-score comparison

The native protocol crosses five task/checkpoint combinations with two cohorts, yielding ten planned comparisons over four checkpoints, two architectures, and two tasks. Eight satisfy their own complete recorded-trajectory, unchanged-effective-scene, mediator, and cross-model initialization controls. Table 4 retains the full planned ledger, with the remaining two cohort-B comparisons reported descriptively. The corresponding requested outcomes, trajectory comparisons, and admission decisions are indexed together in `data/native_rank_4090/result.json`.

Each original arm maps fifty requests to 26 effective identities, whereas the repaired arm preserves fifty. The two cohorts therefore contribute 100 registered task/request identities per task, shared across checkpoints and evaluator arms. All score summaries use these registered requests; repeated processes and native tails retain their distinct roles as controls and execution context.

For the five admitted first-cohort cells, a post-hoc diagnostic uses repaired scene labels to decompose the score contrast. Table 6 reports repaired-minus-original success counts, reversing both reference-term signs from the preserved source record. All 250 covered effective-scene initialization checks

Table 4: The complete native ledger separates eight admitted effects from two descriptive contrasts. O/R gives successes per fifty requested episodes, and Δ is repaired minus original in pp. SA and US denote same-arm and unchanged-effective-scene trajectory checks; admitted cells also pass mediator and cross-model initialization checks. All ten planned cells are retained with their complete scores and respective trajectory-control outcomes.

Cohort	Task / checkpoint	O	R	Δ pp	SA	US	Status
A	Close / ACT	32	33	+2	Pass	Pass	Admitted
A	Close / multitask SmolVLA	24	19	−10	Pass	Pass	Admitted
A	Close / specialist SmolVLA	6	0	−12	Pass	Pass	Admitted
A	Pick/place / ACT	20	17	−6	Pass	Pass	Admitted
A	Pick/place / multitask SmolVLA	7	5	−4	Pass	Pass	Admitted
B	Close / ACT	31	39	+16	Pass	Pass	Admitted
B	Close / multitask SmolVLA	15	21	+12	Pass	Pass	Admitted
B	Close / specialist SmolVLA	3	4	+2	Fail	Pass	Descriptive
B	Pick/place / ACT	14	21	+14	Pass	Pass	Admitted
B	Pick/place / multitask SmolVLA	4	7	+6	Fail	Fail	Descriptive

Table 5: Both repair paths reproduce the paired repaired scores. Errors are absolute pp differences from the full repaired references over eight fixed checkpoint/cohort cells. Bold marks tied minima. New counts execution added by this saved-record comparison; the comparison measures agreement with the repaired reference, with the measured input-generation, selective-execution, and reference-validation costs separated in Appendix D.2.

Representation	Eight native checkpoint/cohort cells			Cost
	Exact / 8 ↑	Mean error ↓	Max. error ↓	New
Original evaluator	0/8	9.5 pp	16.0 pp	0
Simple repair	**8/8**	**0.0 pp**	**0.0 pp**	0
EFA-Replay	**8/8**	**0.0 pp**	**0.0 pp**	0
Paired repaired reference	–	–	–	recorded

pass. The residuals are nevertheless nonzero for SmolVLA: equal effective scenes can occur at different positions in the native sampler. This motivates checking execution context in addition to scene identity, while attributing the admitted contrast to the full evaluator intervention.

Control status of the two descriptive comparisons. B Close/specialist fails the same-arm trajectory repeat; B Pick/place/multitask fails both same-arm and unchanged-effective-scene checks. Although their labels repeat, the trajectories differ at the locations in Table 7. These cells therefore contribute observed score records, while the attributed-effect distribution uses the eight cells satisfying the full protocol. The separate CloseFridge exploration in Appendix E.2 contains four further failed cells, recorded under its own study rather than merged with these planned comparisons.

A.4 Paired requests and aggregate cancellation

The reanalysis behind Figure 3 pairs the existing formal labels by study, task, checkpoint, cohort, width, noise stratum, and requested identity. This key preserves the intervention's target even when the realized scenes differ. Repeat 1 supplies each scored label, and repeat 2 validates it and the complete changed-request set. The analysis retains four controlled SmolVLA configurations, four ACT configurations, two new-scene configurations, and eight admitted native comparisons, using their registered scored prefixes with calibration and unscored tails accounted separately.

Each row is reconstructed from the four paired-label counts and checked against $n_{00} + n_{01} + n_{10} + n_{11} = N$, original successes $= n_{10} + n_{11}$, repaired successes $= n_{01} + n_{11}$, and their difference $n_{01} - n_{10}$. Equation (2) follows immediately from $a + b - |a - b| = 2\min(a, b)$ for nonnegative counts. The table therefore provides an exact accounting of cancellation on the retained records, with any missing or duplicated requested key causing the analysis to fail its input check.

Table 6: Scene reweighting leaves a native-outcome residual. The five admitted cohort-A contrasts use repaired-minus-original success counts. Each reweighting term is computed retrospectively from repaired scene labels and sums with its residual to the full contrast, describing scene composition in the matched execution.

Task	Checkpoint	R−O	Reweighting	Residual
Close	ACT	1	1	0
Close	Multitask SmolVLA	-5	1	-6
Close	Specialized SmolVLA	-6	0	-6
Pick/place	ACT	-3	-3	0
Pick/place	Multitask SmolVLA	-2	-3	1

Table 7: First recorded differences locate the two control failures. Step indices identify the initial discrepancy in observation, action, or state channels. Scored labels agree within the listed pairs, while the trajectory comparison determines their descriptive status. Original execution labels identify their traces.

Task / checkpoint	Comparison	Seed	First obs.	Action/state
B Close / specialist	original repeat	61007	0	0/0
B Close / specialist	repaired repeat	61051	319	350/350
B Pick / multitask	original repeat	61036	331	350/350
B Pick / multitask	repaired repeat	61036	331	350/350
B Pick / multitask	repaired repeat	61049	323	350/350
B Pick / multitask	O/R repeat 1	61036	331	350/350
B Pick / multitask	O/R repeat 2	61036	331	350/350

A.5 INDEPENDENT LIBERO OBSERVATION-TIMING AUDIT

Source-selected deviation and score-free detection. At pinned LIBERO revision `8f1084e`, the standalone `libero/lifelong/evaluate.py` saves the observation returned by `set_init_state`, executes five zero-action settling steps, and discards their returned observations before the first policy call. The training-time evaluator instead refreshes this observation; the finding concerns the standalone entry only. We declare that the first policy input must correspond to the physical state after settling. A policy-free probe records native event steps, observation hashes, and the post-settling MuJoCo state. The existing EFA context checker reads those fields without success labels: it flags all 20/20 exploratory and 30/30 follow-up stale inputs, and none of their corrected counterparts. The one-line intervention replaces only the first input with the observation returned by step five; subsequent inputs remain live in both arms.

Paired outcome test. For `libero_goal` task 8 (put the bowl on the plate), we use the public `Harrysunshine/act-libero-goal` ACT checkpoint, official initial-state indices 0–49, seeds $7 + i$, and a native `OffScreenRenderEnv` adapter reproducing the standalone entry's five zero-action steps, 600-step limit, and native success predicate. Each original/corrected pair uses the same checkpoint and initial state in fresh environments. All 50 pairs have identical post-settling MuJoCo state hashes before the first policy action; all 50 first action chunks differ. The first 20 indices are exploratory because their protocol file was created after the run had started: success is 6/20 with the stale input and 17/20 with the current input (11 stale failures corrected, no reversals). After seeing that result, the remaining indices 20–49 were frozen before execution. Their success counts are 11/30 and 27/30, with 17 corrections and one reversal (two-sided exact paired $p = 0.000145$). These are separate evidence stages, so we make no pooled significance claim.

Sensitivity replay. Four task-8 indices were rerun three times in fresh processes, including one arm-order reversal. Static model parameters and bowl/plate/table positions matched across arms; same-step image differences were negligible relative to the step-0-to-step-5 image change. Three stale-failure/current-success indices retained that direction. The sole reversed index in the 30-state run became two successes in its three replays, so individual boundary outcomes are not perfectly stable.

Scope and reproducibility. The ACT checkpoint was trained under LeRobot timing (ten open-gripper settling actions and a 300-step limit), whereas the standalone entry uses five zero actions and 600 policy steps. Both arms of each pair use the latter protocol; the measured contrast is therefore

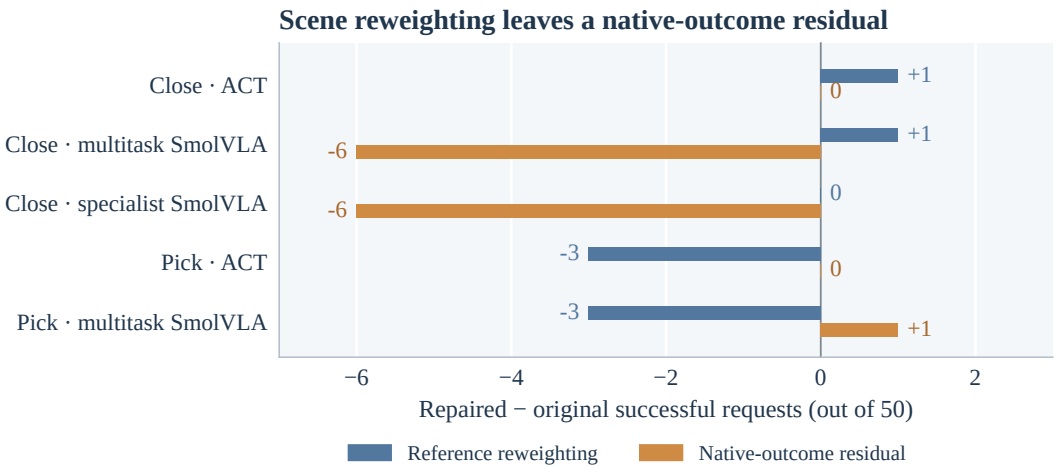


Figure 8: Native score shifts extend beyond scene reweighting. The plotted components sum to repaired-minus-original successes for the five admitted cohort-A comparisons. Reference reweighting uses repaired scene labels, and the residual retains the full-pipeline contrast after accounting for the reference scene composition.

Table 8: Paired counts retain both directions of outcome change. Rows give fixed configurations; Δ is repaired minus original in pp, and D is the percentage of changed labels. Repeat 1 supplies the scores and repeat 2 reproduces each label and changed-request set. Widths and noise strata share the underlying scene identities.

Configuration	N	n_{00}	n_{01}	n_{10}	n_{11}	O/R	Δ pp	D (%)
SmolVLA / Close / p41	16	4	2	2	8	10/10	+0.00	25.00
SmolVLA / Close / p73	16	6	2	3	5	8/7	-6.25	31.25
SmolVLA / Micro / p41	16	16	0	0	0	0/0	+0.00	0.00
SmolVLA / Micro / p73	16	16	0	0	0	0/0	+0.00	0.00
ACT / Close / B4	20	3	4	3	10	13/14	+5.00	35.00
ACT / Close / B5	20	2	2	2	14	16/16	+0.00	20.00
ACT / Pick / B4	20	11	2	3	4	7/6	-5.00	25.00
ACT / Pick / B5	20	9	4	5	2	7/6	-5.00	45.00
New SmolVLA / Close / p41	8	4	2	1	1	2/3	+12.50	37.50
New SmolVLA / Close / p73	8	4	0	2	2	4/2	-25.00	25.00
Native A / Close / ACT	50	7	11	10	22	32/33	+2.00	42.00
Native A / Close / Spec. Smol	50	44	0	6	0	6/0	-12.00	12.00
Native A / Close / Multi. Smol	50	19	7	12	12	24/19	-10.00	38.00
Native A / Pick / ACT	50	24	6	9	11	20/17	-6.00	30.00
Native A / Pick / Multi. Smol	50	42	1	3	4	7/5	-4.00	8.00
Native B / Close / ACT	50	5	14	6	25	31/39	+16.00	40.00
Native B / Close / Multi. Smol	50	23	12	6	9	15/21	+12.00	36.00
Native B / Pick / ACT	50	23	13	6	8	14/21	+14.00	38.00

attributable to first-input timing within this adapter, but neither arm is a standard score for the checkpoint. The original standalone CLI expects official BC checkpoint files, which were unavailable for these tests. We do not infer changes to official BC results, training-time evaluation, all LIBERO tasks, or historical leaderboards. The pinned source, patch, frozen protocols, paired traces, native probes, EFA checks, and render audit are indexed in the supplementary LIBERO artifact.

B Retained information: provenance, definitions, and complete bounds

The information analysis asks which original statistics are uniquely determined after a record is reduced. This appendix first establishes the public outcome population and reconstruction rule, then gives the representation-specific checks, constructive witnesses, and attainable bounds.

B.1 PUBLIC RECORDS AND OBSERVATION UNITS

The inputs bind SimplerEnv revision `06accaca`, the official paper PDF with SHA-256 prefix `7414c22d...`, and public video revision `610a6204` (xuanlinli17, 2024). The archive contains 22,009 MP4s: 6,664 success, 15,342 failure, and three task-level merged videos. Following the pinned metric, labels are read from filenames, with selected paths and condition membership retained in the trace. The analysis uses 6,528 outcomes across all 78 Visual Matching cells, including aggregate cells sharing component records. Pick Coke Can had already been inspected; the remaining task/policy/variant population was frozen before inferential outputs, and reduction and range analyses subsequently used these saved records under specified retained representations.

The recorded counts are 25 per Pick orientation/variant, 60 per Move Near variant, 9 per Drawer variant (27 per endpoint), 27 per Apple variant, and 24 per Bridge endpoint/seed. One Move Near directory provides conflicting success and failure records for RT-1 Converged episode 0, despite covering every ID 0–59. Because the authoritative rerun is unspecified, we retain both resolutions, with rates .4375 and .4417. They yield the same headline representation and identification counts, while the accounting check requires resolving the extra record before accepting 60 episodes. The six Eggplant cells retain $N = 24$ conditions alongside the helper's $M = 30$ pseudo-trials.

B.2 EXACT EMPIRICAL DECOMPOSITION OF REPRESENTATION LOSS

To separate grouping loss from rounding and count changes, let $x_i \in [0, 1]$ denote the N observed condition means, with $p = N^{-1} \sum_i x_i$ and $w = N^{-1} \sum_i x_i(1 - x_i)$. The released reconstruction creates M binary values with success fraction $q = \mathrm{round}(Mp)/M$. Their variance s_z^2 and the condition-mean variance s_x^2 satisfy the exact finite-ledger identity

$$s_z^2 - s_x^2 = \frac{N}{N-1} w + \left[\frac{M}{M-1} q(1-q) - \frac{N}{N-1} p(1-p) \right]. \tag{8}$$

The identity follows from $N^{-1} \sum_i x_i^2 = p - w$. Its first term isolates within-condition disagreement, while the bracket captures rounding and any change in sample count. For the six Eggplant comparisons, the bracket explicitly retains $M = 30$ and $N = 24$; for the remaining cells, $M = N$. This keeps the empirical representation comparison tied to the condition and pseudo-trial counts.

Under either duplicate resolution, the full ledger contains 65 cells with mixed conditions, 64 variance increases, one decrease, and 13 unchanged variances. Nine condition variances are zero, leaving 69 cells in the ratio summary while all 78 remain in the variance-gap analysis. The ratio has median 1.92665 and range 0–5.71429. The endpoints illustrate the identity: Drawer Open/Octo-Base changes condition means $(1/4, 0, \ldots, 0)$ to all-zero pseudo-trials, reducing the mean from $1/108$ to zero and variance from .00231481 to zero. Apple/RT-1 Converged instead has twenty zeros and seven $1/4$ means, reconstructed as two successes among 27 pseudo-trials—the largest ratio.

The same pattern persists when aggregate rows are removed. Among the remaining 66 cells, variances increase in 52, decrease in one, and agree in thirteen; the median ratio is 1.95 across 57 nonzero condition variances. The existing 51-cell terminal/no-Eggplant family gives 42 increases, one decrease, and eight unchanged cells, with median ratio 2.05 across 44 nonzero variances. These subsets retain the same finite-record pattern. The script `analysis/representation_loss.py` regenerates the full results and checks both the identity and the equal-mean counterexample.

B.3 ALLOWED REPRESENTATIONS AND TARGET DEFINITIONS

The retention study starts from the same five frozen public trace files. Before reducing any record, it resolves the Move Near conflict by min/failure and max/success separately and reproduces every reported pooled count, condition variance, pseudo-trial mean, and pseudo-trial variance. Each representation then receives a header containing the task/policy cell, resolution, original count K, number of conditions N, common group size k, and pseudo-trial length M. Equal group size is checked against source membership, establishing equal pooled and condition-weighted means.

R0 retains the complete grouped binary arrays. R1 retains their K values in canonical sorted order, removing filenames, original order, and membership; R2 retains the exact pooled mean as an integer numerator/denominator pair. R3 retains only the header and the vector returned by

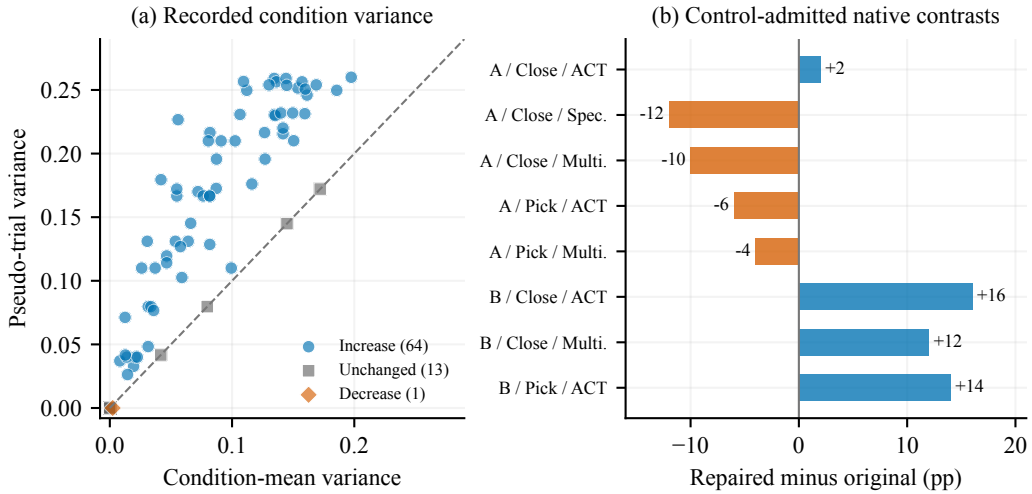


Figure 9: Complete empirical distributions and fixed-policy score contrasts. Left: all 78 condition-mean versus pseudo-trial variance comparisons, including overlapping points and the single decrease. Right: the eight control-admitted native contrasts. Each panel preserves its original comparison population.

`construct_unordered_trial_results` at revision `06accaca`. Extracting that function unchanged preserves its NumPy rounding and integer conversion. Every resulting vector matches the frozen reconstruction, including the six Eggplant cases with $N = 24$ and $M = 30$.

The targets refer to the original record: T1 is its success count and exact pooled mean; T2 is its condition-indexed table of counts, successes, and means; T3 is the sample variance of those condition means. Each target is evaluated using the specified header and retained representation. This access rule makes the test operational: the recovery process must determine the target from precisely the information delivered to it, while the full grouped record is reserved for subsequent verification.

B.4 Uniqueness and compatible-record witnesses

The three statuses encode distinct conclusions. `EXACT` requires a unique target value under the allowed information. `NOT_IDENTIFIABLE` requires two admissible grouped records with the same header and representation but different target values. `UNRESOLVED` records that neither has been established. Thus numerical agreement with the original reference becomes evidence of exact recovery only when accompanied by a uniqueness argument under the same allowed representation.

For R1 and R2, the total success count S is fixed; for R3, the script enumerates every integer $S \in \{0, \dots, K\}$ reproducing the visible vector under the released helper. At each feasible total, balanced and concentrated allocations produce counts $s_1, \dots, s_N$ satisfying $0 \le s_i \le k$ and $\sum_i s_i = S$. Reversing a concentrated allocation additionally checks ambiguity of the condition-indexed table. Every pair is passed through the representation constructor and accepted as a witness only when both visible representations agree exactly and their target values differ under the same header.

Appendix B.7 proves that the balanced and concentrated allocations attain the variance extrema. If the extrema coincide at every feasible total and share the same value across totals, T3 is uniquely determined; otherwise the endpoint allocations provide a counterexample. For T2, all-zero or all-one records can force a unique numerical table even when the historical assignment of individual trials is unavailable. These cases explain why missing grouping affects some targets but leaves others fixed.

Algebraic verification covers the two-condition example $((0,0),(1,1))$ versus $((0,1),(0,1))$, all-zero/all-one labels, pooled-total preservation, loss of a success through rounding, and unequal N/M. Exhaustive enumeration for $N = 2, 3, 4$, $k = 1, 2, 3$, and $M = N, N + 1$ compares each status with the complete feasible set. These checks validate the uniqueness and witness logic.

Table 9: Every studied target is resolved by uniqueness or a compatible counterexample. Entries give exact / non-identifiable / unresolved counts over all 78 cells under each conflict resolution. Both resolutions yield the displayed results; the third entry retains the audit's explicit status for insufficient evidence.

Retained representation	T1: pooled	T2: condition table	T3: variance
R0 Full grouped outcomes	78 / 0 / 0	78 / 0 / 0	78 / 0 / 0
R1 Ungrouped outcomes	78 / 0 / 0	9 / 69 / 0	14 / 64 / 0
R2 Exact mean and counts	78 / 0 / 0	9 / 69 / 0	14 / 64 / 0
R3 Released pseudo-trials	8 / 70 / 0	4 / 74 / 0	8 / 70 / 0

Table 10: Numerical proxy agreement and identifiability measure different properties. Maxima cover all 78 cells under each resolution, which give the same values. R3 uses its own sample variance as a proxy for T3; a dash indicates that no condition-variance proxy is produced. Exact recovered values have zero target error.

Representation	Maximum absolute mean error (pp)	Maximum absolute variance error
R0 Full grouped outcomes	0.00	0.00000
R1 Ungrouped outcomes	0.00	–
R2 Exact mean and counts	0.00	–
R3 Released pseudo-trials	2.00	0.17083

B.5 Complete identification states

Table 9 expands the main-text display into exact, non-identifiable, and unresolved counts. Both conflict resolutions yield identical totals, and every studied target receives either a uniqueness argument or a compatible-record witness. The table preserves all three output categories so that these completed judgments remain distinct from cases where the audit lacks a decision.

B.6 Isolation, numerical error, and complete output

Access isolation separates construction of the retained input from verification against the original record. The reference process computes T1–T3 from complete grouped outcomes and serializes each permitted representation. A fresh worker reads only that input; an audit hook blocks the reference and unrelated files, and an explicit attempted reference read checks the restriction before analysis. All four denial checks pass. The evaluator checks exact results and witnesses against the separate reference, retaining all 156 resolution–cell rows per representation and 960 input–target witnesses.

These post-hoc reanalyses retain the original input provenance alongside their outputs. The paired configurations and four-count tables are stored in `aggregate_blind_spots.csv`; the 624 representation–resolution–cell rows, numerical errors, and target statuses are in `evidence_retention_cells.csv`; and `evidence_retention_witnesses.json` provides the compatible records and unequal target values. The source hashes, input files, and verification receipts accompany `analysis/plan_b/`, allowing direct checks of the reported information loss.

B.7 Sharp identification ranges

For a cell with $N > 1$ equal-sized conditions and k binary outcomes per condition, set $K = Nk$ and let S denote the original success count. R0 fixes the grouped record; R1/R2 fix its total; and R3 admits every integer $0 \le S \le K$ that produces the observed pseudo-trial vector through the pinned helper, including floating conversion and NumPy rounding. With $\mathcal{S}$ denoting these compatible totals, the success-rate range is $[\min \mathcal{S}/K, \max \mathcal{S}/K]$. The header and vector determine this set.

For a fixed total, let $s_i \in \{0, \dots, k\}$ count successes in condition i. The target variance is

$$V(S, \mathbf{s}) = \frac{\sum_{i=1}^{N} s_i^2 - S^2/N}{(N-1)k^2}.$$

Writing $S = Na + r$ with $0 \le r < N$, a balanced allocation has squared sum $(N-r)a^2 + r(a+1)^2$. It attains the minimum: whenever two counts differ by at least two, moving one success from the larger to the smaller strictly reduces the squared sum, and repeated transfers reach this allocation.

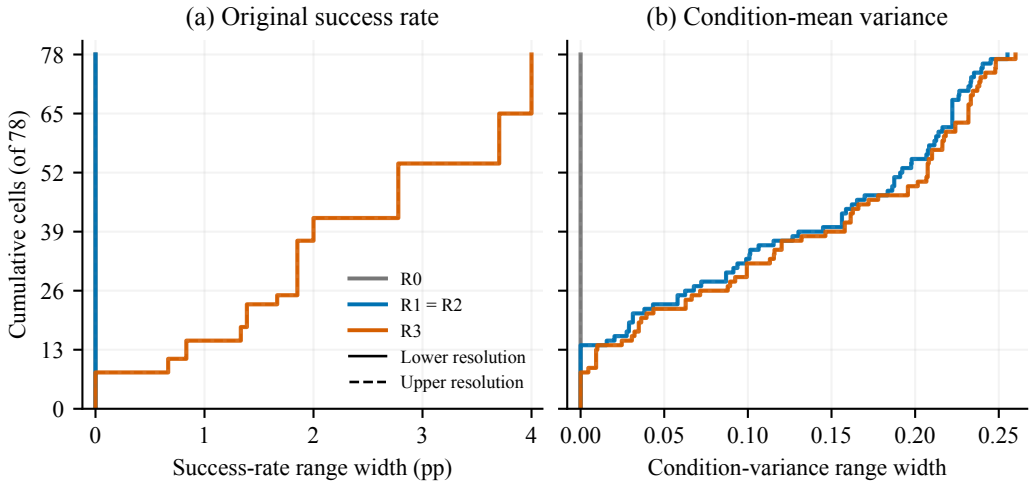


Figure 10: Complete range-width distributions reveal the information retained by each representation. Each curve contains all 78 cells under one duplicate resolution; solid and dashed curves denote the two resolutions and mostly overlap. R1 and R2 coincide. Zero-width cases are retained under the stated per-cell constraints.

Conversely, writing $S = qk + b$ with $0 \le b < k$ gives maximum $qk^2 + b^2$. Moving successes toward a larger non-full group increases the sum until all but at most one nonzero count equal k. Both constructions correspond to feasible binary records, proving attainability for the equal-group setting.

For R3, we take the minimum and maximum over every total in $\mathcal{S}$; for R0, both endpoints equal the observed variance. The intervals therefore report the sharp extremes of each target's compatible set. They are finite-record identification ranges, distinct from confidence intervals based on sampling uncertainty. Discrete target values can leave gaps between endpoints, and different targets can attain their endpoints in different witnesses. Each cell is solved under its own header and representation; using additional constraints from overlapping cells could narrow these per-cell ranges.

All R0 ranges have zero width under either conflict resolution. R1/R2 identify every pooled rate, while 14/78 variance ranges have zero width; their variance-width median and maximum are 0.1375679 and 0.2552083. R3 has eight zero-width rate ranges and eight zero-width variance ranges. Its rate-width median and maximum are 0.02 and 0.04, with corresponding variance-width summaries 0.1521266 and 0.26. Figure 10 retains the complete distributions, including zero widths and both resolutions.

The 624 cell–representation rows each carry four endpoint witnesses, for the lower and upper mean and variance, totaling 2,496 compatible records. The CSV reports every endpoint and width, and the JSON files retain each witness and feasible total. These constructions document what each representation permits under the stated constraints; the original observations remain the separate reference for checking the representations and numerical reconstructions.

C Audit declarations and diagnostic decisions

The diagnostics examine how the audit's decision changes with the declaration and available record fields. This appendix specifies their input populations, the two worked contracts, and the connection between an observed deviation, its supporting evidence, and the available recovery decision.

C.1 Case populations and reference custody

The twelve-case suite combines six deviations, five legal configurations, and one information-insufficient input. The deviations include two LeRobot configurations, a record-level reconstruction of the SimplerEnv mechanism, two registered injections targeting context and propagation, and an OpenDrawer case. The legal inputs cover identity preservation, declared replacement, sampling with

Table 11: The target determines what evidence must be retained. Sufficient obligations for the studied targets distinguish exact recomputation, supported execution, and evidence gaps. A missing field affects the conclusions that depend on it; Appendix C.3 details the required fields and executable checks for each target.

Target	Evidence that supports it	Permitted conclusion
Request outcomes	Requested identity paired to each arm's label	Net change and changed-request set
Pooled success	Actual outcomes, or exact mean and original count	Exact total and success rate
Condition variation	Outcome-to-condition binding or a uniqueness proof	Condition table or sharp compatible range
Trajectory reuse	Identity, source/context, and supported dependencies	Eligible records outside invalidation closure
Missing attempts	Supported callback and registered replay boundary	Replay and validate the declared target

Table 12: Complete decisions on the registered diagnostic suite. Entries count defect / no detected defect / unresolved outputs for the six deviations, five legal configurations, and one insufficient input. Table 14 associates each decision with its proposition, supporting record fields, and the corresponding separate recovery status.

Method	Six deviations	Five legal cases	One insufficient
B0: repeat / total	0 / 5 / 1	0 / 5 / 0	0 / 1 / 0
B1: nominal IDs	0 / 6 / 0	0 / 5 / 0	0 / 1 / 0
B2: full-log assertions	5 / 1 / 0	0 / 5 / 0	0 / 0 / 1
B3: paired oracle	6 / 0 / 0	0 / 5 / 0	0 / 0 / 1
EFA-Replay	6 / 0 / 0	0 / 5 / 0	0 / 0 / 1

replacement, best-of-two, and variant averaging. The final case removes evidence needed to decide reuse. All inputs remain counted, including known-label cases receiving no verdict.

Source and record verifiers supply the natural labels, and diagnostic labels are frozen before method scoring. The new-path label comes from a standard-library parser run after predictions are saved. Subsequently, an isolated agent independently adjudicates all twelve inputs using only the permitted declarations, source, and records; access to predictions, scored outputs, manuscript, handoff state, and frozen labels is denied. Its 12/12 agreement provides a computational cross-check. The artifact retains the reads, pair comparisons, permission matrix, and recorded adaptation and verification time.

C.2 Contract instantiation and practical configurations

From a declaration to a check. Each audit record identifies the evaluation claim and strongest applicable declaration, including its source and any conflict. It then specifies a reachable precondition, original and comparison operations, mediator, outcome, controls, and repair replay. Revisions, denominators, and trace comparisons follow shared recording rules. Domain judgment determines which declaration governs the path and which mediator is meaningful; mechanical checks then evaluate the specified correspondence. Where the semantics admit both operations or leave the target ambiguous, the record reports design sensitivity or ambiguity. The full procedure and supporting source excerpts are retained in `docs/CONTRACT_INSTANTIATION.md`.

Configuration evidence. The configurations in Table 13 explain how the audited path is reached. Pinned automatic sizing can select multiple slots, while PR 3217 (March 25, 2026) records an earlier asynchronous ACT invocation using five slots, twenty episodes, and seed 87 on CloseFridge and PickPlaceCounterToCabinet. That invocation supplies the selection rationale for our tasks, checkpoints, seeds, and five-slot arm; the outcome evidence comes from execution at the audited revision, including the added four-slot comparator. Controlled SmolVLA similarly uses explicit seeded four-slot execution on CloseFridge and OpenMicrowave. Configuration sources and the unaffected single-worker records are indexed in `evidence/efa_review_20260917/`.

LeRobot: caller and adapter compose a seed mapping. At revision `b6ec0060`, the caller supplies and reports a sequential explicit seed vector, defining the requested scene identities. The adapter

Table 13: Exposure follows the evaluator configuration at the audited revision. Source-level reachability, executed reset checks, and measured policy outcomes are shown separately. The earlier public invocation provides the selection rationale, while the admitted outcome comparisons use the pinned experimental revision.

Configuration	Existing evidence	Supported scope
Guide: one worker	No worker-index offset beyond zero	Unaffected by the double-offset mechanism
Explicit four workers	Full-stack reset and policy checks at the pinned revision	Scene replacement and repeated coverage; effects in admitted comparisons
Explicit five workers	Native ACT study, twenty requests	Changed scene identity without duplicate effective scenes
Automatic sizing	Source rule; mocked CPU counts select widths five and eight	Source-defined path exposure
Historical PR 3217	Five-slot ACT invocation on an earlier implementation	Historical configuration informing test selection

separately applies worker spreading, an operation intended to separate workers initialized from a common seed. Composing it with already-distinct caller seeds changes the declared population. The contract therefore compares the actual mapping with the caller's explicit registry, while allowing a derived-seed design when the declaration explicitly specifies that evaluation population.

Contract field	Instantiation
Precondition	Explicit scene seeds, multiple vector slots and reset waves.
Original operation	Adapter reset adds the factory's worker index to the supplied seed.
Comparison	Forward the assigned explicit seed; retain the missing-seed fallback in the controlled intervention.
Mediator	Requested/effective seeds, initial observations/state hashes and cross-wave scene collisions.
Outcome	Native labels for every scene/noise block, discordances and complete success totals.
Controls	Coupled noise, fixed checkpoint/context, exact within-arm repeats and unchanged-effective-seed comparisons.
Repair replay	The same registered cohort with only the explicit-seed offset removed.
Admission	Controlled conditional outcomes supported by matched mediator and outcome records.

With start seed s, width B, wave w, and slot j, the caller seed is $s + wB + j$ and the effective seed is $s + wB + 2j$. This source composition predicts the scene mediator before the labels are interpreted. Even widths permit adjacent-wave collisions; odd widths instead alter identities while avoiding that collision pattern. Source excerpts locate seed accounting in `lerobot_eval.py`, worker indexing and reset in `envs/robocasa.py`, and automatic sizing in `configs/default.py`.

SimplerEnv: reporting changes the observation representation. At revision `06accaca`, the Visual Matching script averages variant success fractions and reconstructs rounded binary trials before inference. The paper's observed-trial declaration governs record retention, while Section 4.4 separately defines condition means as targets of the information analysis. Keeping these two roles explicit links the accounting repair to original trials and the variance analysis to condition groups.

Contract field	C4 instantiation
Declaration	Retain the paper's observed binary trials; separately analyze mean success over fixed variants with initial condition as the simulation unit.
Comparison	Replace aggregate-based rounded pseudo-trials with actual outcomes; group variants by registered condition for the stated condition-statistic targets.
Mediator	Trial counts, condition membership, empirical distributions, exact representation loss and compatible-record witnesses.
Checks	Frozen public inputs, all 78 cells, both duplicate resolutions, complete denominators, loss decomposition, allowed inputs and endpoint witnesses.
Repair	Actual outcome vectors and validated counts repair accounting; condition targets additionally require retained grouping or a valid uniqueness argument.

The contract uses source and statistical judgment to select the applicable declaration and observation unit. Once specified, counts, membership, representation changes, and compatible-record witnesses are mechanically checked. Actual outcome vectors and validated denominators repair trial accounting; condition statistics additionally use retained grouping or a uniqueness argument. This separates the information required by each target while applying the same declaration-to-record audit.

C.3 Separating deviation and recovery decisions

The stored suite includes implementation, context-gap, dependency-invalidation, attribution, and recovery fields. We read these jointly to distinguish their conclusions in Table 12. For the three legal sampling/aggregation inputs, "not observed" in the implementation field and zero context gaps yield no detected defect, while their recovery field remains unresolved because the required boundary is unavailable. The missing-context case retains unresolved detection. For dependency spill, downstream invalidation supports the verdict even with unchanged unit binding.

The constructor also clarifies what the supplied controls mean. In the first eleven cases, both flags are preset true and repeat records are copies: the seed-profile candidates correct identity fields while retaining labels, and the grouping candidate restores saved groups. These inputs test structural decisions; measured policy-effect evidence comes from the separately controlled executions. Table 14 therefore names the proposition checked in each case and the evidence supporting it. Original inputs and generic labels accompany the proposition-specific interpretation and supporting fields.

Across the suite, two inputs support record reuse, three support replay, and seven retain an evidence gap. In particular, a legal operation and a supported recovery are distinct outcomes: correctness of the declared sampling or aggregation rule does not supply a missing execution boundary. The richer records in Appendix D address that next step with the evidence needed to validate recovery.

D Recovery validation and execution boundaries

Recovery connects a record-level decision to the result of restoring its declared target. Following Figure 7, the cases below separate complete-record recomputation, selective completion, shared-state replay, and missing historical identity, with execution and reference validation accounted separately by role.

D.1 Prospective native recovery and source repair

The protocol registers four regimes: R1, complete SimplerEnv outcomes with a reporting error; R2, missing episodes at a native reset boundary; R3, state shared across a saved native process; and R4, absent historical identity and context. Plans precede inspection of the separate reference program or execution. R1, R3, and R4 use saved-record CPU checks. R2 executes eight roles with the pinned evaluator and checkpoint on one GPU: two existing inputs, two current-core callbacks, and four reference runs for seeds 99000–99003. The core seals a four-unit receipt before reference execution, and all 28 gates pass with 64 exact matches across the recorded fields against the reference.

Developing the registered boundary revealed a source-level reset problem. The retained pre-repair inventory contains eleven strict pairs, seven passing and four failing from initial reset, with 152/176 matching fields. In `Counter.get_reset_regions`, a `set` changes XML-candidate order across fresh processes before the seeded choice; the earlier source identity also omitted this file. Preserving insertion order and registering the actual import path, file hash, revision, and patch hash addresses both gaps. Separate canaries 98993 and 98994 then match 16/16 fields each before the disjoint four-seed matrix is run. These checks establish the registered source and initialization conditions; the protocol requires new collection after any subsequent source-identity or exactness failure. Appendix E.4 reports development, selective-execution, and reference-validation costs.

D.2 Offline recovery protocol and archived costs

The offline comparison fixes two strategies before inspecting their outputs. Both receive the same declaration, visible records, and supported boundaries, and both must satisfy identity, context, coverage, and statistical requirements using decision-time information. The conservative strategy

Table 14: Each diagnostic supports a specific proposition. P denotes preset true flags in constructed controls. The new-path records provide a mapping comparison with unchanged labels and an unavailable same-arm repeat. The table keeps detection, proposition support, and recovery distinct; Appendix D evaluates richer inputs.

Case	Detection	Proposition being checked	Evidence / control	Proposition support	Recovery
Four-worker collision	defect	Candidate removes identity violations	37 mismatches; constructed corrected IDs, unchanged labels; P	identity mapping	unresolved
Nine-worker substitution	defect	Candidate removes identity violations	44 mismatches; constructed corrected IDs, unchanged labels; P	identity mapping	unresolved
SimplerEnv grouping	defect	Omitting groups loses condition binding	6,528 saved labels; constructed field removal/restoration; P	representation effect	unresolved
Identity no-op	none detected	No-op preserves declared records	two constructed records; P	conformance; defect effect N/A	reuse
Declared replacement	none detected	Replacements satisfy permitted identities	two constructed replacements; P	conformance; defect effect N/A	reuse
Sampling with replacement	none detected	Repeated actual units are permitted	declared multiplicity; P	conformance; defect effect N/A	unresolved
Best-of-two	none detected	Two attempts per request are permitted	four constructed attempts; P	conformance; defect effect N/A	unresolved
Variant averaging	none detected	Variant grouping is permitted	two constructed variants; P	conformance; defect effect N/A	unresolved
Context mismatch	defect	Declared context rejects injected mismatch	one constructed context pair; P	diagnostic only	replay
Dependency spill	defect	Invalidating d_0 excludes d_1, d_2	declared three-node chain; constructed labels; P	declared graph consequence	replay
Missing context	unresolved	Available evidence supports outcome attribution	one required field absent; no candidate; P	not assessed	unresolved
New-path OpenDrawer	defect	Repair has an attributable outcome effect	paired native records; no changed labels; same-arm repeat unknown	outcome effect unresolved; mapping observed	replay

recomputes complete records and otherwise rebuilds the whole declared target; EFA-Replay first checks eligibility and dependencies to retain usable records. A manual implementation of those same eligibility checks can select the same repair. The comparison reports whole-target requirements alongside selective recovery validated through archived execution at the registered process boundaries.

Partial records and eligibility. The target is `TurnOnElectricKettle`, seeds 99000–99003, with one scored attempt each, a fixed checkpoint, batch size one, bootstrap seed 271828, and policy-noise seed 109. At decision time, the two `partial_existing` directories are the available outcome-bearing inputs. Their configuration, identity digest, runtime source, construction RNG, model XML, traces, and labels establish the context to be checked. Required fields include initial observation/state hashes, active-transition digests, source/configuration bindings, and explicit non-scoring treatment of terminal autoreset. Channel coverage follows Appendix A.1.

Eligibility requires requested/effective identity agreement, the declared attempt, group and report population, randomness instance, checkpoint, protocol, mandatory context, and source/configuration bindings. The frozen declaration implements source checking through `registered_static_inputs`; its generic `require_source_digest` flag remains false. Initial and trajectory fields

Table 15: Retained evidence determines which conclusion can be checked. These sufficient requirements connect the studied targets to their corresponding unit, context, or boundary evidence. The final column gives the resulting check and its output, including exact statistics, compatible alternatives, and recovery.

Conclusion	Retained evidence	Check and bounded output
Request-level outcome change	Requested identity linked to each arm's outcome	Pair requests, check repetitions, report net and changed-label counts.
Realized scene matches request	Request-to-effective-scene binding and initial-state evidence	Test substitutions/collisions against the declared seed rule.
Original pooled success	Original binary outcomes, or exact mean and original count	Recompute total; verify uniqueness if only rounded pseudo-trials remain.
Condition means and variance	Outcome-to-condition binding, or a valid uniqueness argument under public constraints	Recompute, exhibit distinct compatible records, or retain unresolved status.
Old trajectory can be reused	Unit identity, context, source, and supported dependencies	Exclude invalidated descendants and check the applicable population rules.
Missing episode can be recovered	Supported replay callback and verified source/process boundary	Execute and compare the registered receipt; distinguish new collection from historical recovery.

Table 16: Archived execution costs are separated by role. Evaluator and role timing files provide measured input-generation, selective-completion, and reference-validation costs. Role wall time is summed across roles. Whole-target rebuilding is a planned comparison, and its newly scheduled runtime remains unmeasured.

Component	Episodes / processes	Evaluator seconds	Role wall seconds
Existing input generation	2 / 2	330.53	393.02
Selective completion	2 / 2	315.22	374.29
Independent reference validation	4 / 4	642.37	770.77
Total archived role matrix	8 / 8	1288.13	1538.09

must be present before reuse, with equality to the reference checked afterward. Both saved inputs qualify; the declared graph has no invalidation root or cross-episode edges. Missing seeds 99002 and 99003 consequently map to `fresh-evaluator-process/99002` and `fresh-evaluator-process/99003`, with registered RNG initialization and the order-preserving region repair.

The independently reconstructed plan matches the archived sequence: plan, two selective callbacks, four-record receipt, then four reference runs. A separate standard-library reader recomputes sixteen fields per seed from raw role files and obtains 64/64 exact matches; labels $(0, 0, 0, 1)$ yield success rate $1/4$ with complete coverage and no invalid reuse. A whole-target strategy requires four fresh episode/process executions, recorded here as a plan. The archived reference matrix instead serves to validate the selective result. If eligibility, boundary, or coverage checks fail, the protocol seeks a supported wider boundary or reports an evidence gap; authorized new collection defines a new target.

The archived GPU monitor records 1308.26 active-context seconds (0.3634 hours) across the eight roles, including generation of the two input records. Table 16 separates input generation, selective completion, and reference validation. Together these records support the statement that, after supplying valid records and boundaries, the partial target needs two fewer episode/process executions than whole-target rebuilding. Total comparative runtime remains undetermined: a newly scheduled whole-target baseline was not run, and historical adaptation, logging design, boundary establishment, and debugging time were not prospectively measured. The ledger therefore reports the measured components individually, separating selective completion from reference validation.

Complete and shared-state records. For complete SimplerEnv inputs, either strategy recomputes all 78 cells from 6,528 outcomes on CPU. The shared-state case instead concerns an ACT CloseFridge process with requests 60000–60051, of which 60000–60049 are scored and the last two provide execution tails. Thirty-nine mappings change. A conservative declared chain links records in process order, extending invalidation to 60001–60051, or 51 records; the supported fresh-process boundary then covers all 52 and replaces the remaining candidate. Both strategies therefore plan one process.

The saved repaired execution contains fifty scored outcomes and 33 successes, verified from raw rewards and labels, and supplies the retrospective check of that whole-process recovery boundary.

For missing identity, the Apple/RT-1-15% aggregate supplies a vector of length 27 but lacks the binding to the named historical attempts. Both strategies therefore preserve the unresolved historical target. New collection can provide a fresh evaluation population, whereas reconstructing the earlier named attempts requires additional identity evidence. This case completes the distinction between recovering a statistic, replaying an identified attempt, and collecting new observations.

D.3 Separately implemented internal verification and runnable checks

The portable pipeline proceeds from hash-bound inputs and fixed declarations to decisions or compatible witnesses, then to verification and reported results. `ranges.py` uses the pinned helper and equal-size allocator inside workers restricted to their retained inputs. The recovery planner similarly blocks selective and reference outcomes while reconstructing the partial-record plan. These scripts operate on saved files; execution callbacks belong to the archived runtime protocol.

The internal checker `verify.py` is implemented separately from EFA and the range generator. It combines an independent role reader and reference parser with rational moment calculations and closed-form extrema. This checks every feasible success count, every sharp endpoint, all 2,496 endpoint witnesses, and the earlier 960 non-identifiability witnesses. Exhaustive enumeration for $N = 2, 3, 4$ and $k = 1, 2, 3$ validates both extrema across 63 success-count buckets. For recovery, identity, context/source, coverage, closure, and reference fields are rechecked. Deliberate corruption of identity, context, coverage, stale descendants, and witness representations fails these predicates.

These separately authored checks provide internal implementation verification under shared specifications and parsing conventions. Their common design assumptions remain a possible source of error. The companion `diagnostics.py` additionally reparses the twelve input bundles to count identity violations, missing fields, and paired changes, giving the proposition-specific interpretation in Table 14. The full commands, portable inputs, expected outputs, and verification entry points for all three scripts are documented in `analysis/recovery_revision/README.md`.

E Reproducibility, scope, and related work

This appendix connects the paper's claims to its reproducibility paths and consolidates the scope of the evidence. It also documents related work, compute accounting, and asset licenses.

E.1 Claim-to-check map

The separate portable recheck package supports recomputation from saved evidence, while the full companion artifact supplies raw-trajectory verification and fresh execution under pinned checkpoints, assets, and runtime. Table 17 separates these paths so that each result can be checked at the appropriate input depth. Saved-record checks establish record properties; execution evidence documents their generation.

The compact recheck reads saved control comparisons. With full inputs, the ACT verifier instead reconstructs all 168 same-arm pairs and 36 unchanged-seed controls and verifies scene mappings and frozen forecasts. The native verifier checks the eight admitted cells and their associated controls, whereas portable native reconstruction uses bounded saved inputs. SimplerEnv reconstruction covers both conflict resolutions through exact record calculations. Appendix D.3 specifies the internal range and recovery checks; the README and full artifact provide all commands, exact input hashes, runtime versions, complete per-result outputs, and source-patch applicability checks.

E.2 Selection and supplementary scope

The study's principal conclusions concern the two mechanisms at the pinned paths and the finite populations specified in Appendices A–B. The original source-guided corpus spans eight public stacks selected for operation diversity, with prior knowledge of some defects and development witnesses. This purposive design supports mechanism analysis; estimating prevalence would require a separate

Table 17: Reproduction paths connect the three core claims to their evidence. Portable checks recompute saved inputs. Full-artifact paths additionally use raw trajectories or fresh execution under the documented runtime, with the input depth and corresponding evidence specified for each of the three claims.

Claim	Portable inputs and check	Additional full-artifact evidence
Unit changes	`analysis/native/`, `analysis/simpler/`, and `analysis/plan_b/`: identities, paired labels, denominators, representation changes	Raw ACT/native trajectory controls and frozen forecast receipts
Information retained	Recovery-analysis inputs: fixed allowed representations, unique values, compatible witnesses and sharp endpoints	Public-record manifests and pinned released helper
Supported recovery	Same directory: eligibility, closure, boundary selection, archived callback/reference fields and corruption checks	Registered runtime, repaired source and native execution-role records

sampling design. Historical public configurations inform test selection, while the present execution records establish effects at the audited revisions. Corpus reports and search provenance are archived.

Several distinctions govern the evidence throughout the paper. Paired requests, repeated widths, and overlapping statistical cells reuse scenes or outcomes, so their exact summaries describe the registered records. Population uncertainty requires a corresponding sampling model. Record-identification bounds instead quantify values permitted by each cell's retained information, as formalized in Appendix B.7. Recovery is conditional on recorded context, supplied dependencies, and the registered execution boundary; the physical-state trace covers the channels specified in Appendix A.1. These conditions define the scope of each supported result without conflating its distinct forms of evidence.

Supplementary studies retain the same distinction between observed behavior and admitted effects. A separate CloseFridge exploration preserves four failed trajectory gates and withholds outcome attribution. OpenDrawer exhibits a mapping change with unchanged labels and an unavailable same-arm repeat (Table 14). The post-freeze RoboMME check restores binding from $[0, 1, 0, 1]$ to $[0, 1, 2, 3]$ in four fresh-process simulator resets, ending at reset before policy rollout. A frozen known-path policy extension yields an admitted zero contrast, with both arms at 0/8 and no changed labels. These records document the tested application boundaries alongside the primary findings.

E.3 Closest work

The comparisons in Table 18 identify the foundations supplied by prior work and the robot-specific evidence added here. They compare declared objects, available inputs, and demonstrated outputs. B3 is the study's internal paired oracle, which isolates the information available to that comparison; the cited systems retain their own methods, interfaces, and broader design goals in the comparison.

E.4 Compute disclosure

Saved-record analyses use CPU processing; the original verifier ran on an Apple M4 Pro with 24 GB unified memory. Policy execution uses fixed checkpoints on memory-expanded RTX 4090 boards with 46,068 MiB visible memory, an Intel Xeon Gold 6462C host, single-threaded rendering, OSMesa 23.2.1, and LLVM 15. Inference disables mixed precision, and the study performs no model training. The following ledger retains calibration, main execution, and supplementary stages separately.

Table 18: Related approaches supply complementary foundations for the robot-specific audit. Columns compare objects, available evidence, and supported outputs. The final column identifies the checks and findings developed here; B3 is the internal paired oracle used for the registered diagnostic comparisons in this study.

Work	Existing objects, evidence and outputs	Robot-specific question and evidence here
Estimands; BetterBench	Populations, outcomes and aggregation; design assumptions and benchmark life-cycle assessment.	Bind declared scenes and attempts to execution; check the resulting population and denominator.
Rollout Cards	Tasks, rollouts and reporting views; worker/dependency records, state and drops manifests support reanalysis and reference-bundle checks.	Test actual scene/attempt eligibility, invalidation and supported process boundaries; separate historical recovery from new collection.
Jiang et al.	Manipulation validity diagnostics for shortcuts, overfitting, data sources and significance; discussion of rollout nondeterminism.	Trace seed composition and observation reconstruction to changed units and quantify per-cell information loss.
ABE-Ralph	Reference-method, data and resource contracts; staged verification and logged-output fallback support contract-aligned reproduction.	Check robot record reuse against generating context and dependency closure, then validate the selected execution boundary.
Causal and metamorphic testing	Declared interventions or transformations and expected relations; causal assumptions or metamorphic preconditions; CTF can reuse runtime data.	B3 tests paired operations; EFA additionally checks reuse, invalidation, replay boundaries and post-recovery coverage.

Study stage	Processes	Executions	GPU-hours	Wall minutes
Calibration V1–V5	18	72	0.948	68.28
Fixed controlled cohort	16	256	3.695	119.13
Fresh-environment replay	4	16	0.272	16.35
New-scene check	8	64	0.955	28.93
Native ACT transfer	24	368	4.933	378.78
Native model-score comparison	60	2160	31.164	393.63
Separate CloseFridge (all stages)	40	1216	17.809	457.22
EFA-Replay saved-record checks	1	recorded inputs	0	< 0.1
Current-core E4 repaired terminal gate	8	8	0.363	29.89
Post-freeze OpenDrawer	2	4	0.083	5.37
Held-out TurnOnElectricKettle E5	8	160	1.304	39.50
Post-freeze RoboMME identity checks	2	8 source resets	0	< 0.1
Actual RoboMME environment check	4	4 simulator resets	–	0.53
Frozen known-path E5 extension	32	32	–	86.83

GPU-hours sum allocated inference-device time per process, including initialization and rendering waits. They measure process allocation in the shared environment; wall minutes span the first start to last finish and can include pauses. Setup and fixed-action diagnostics are accounted separately. The native comparison's overlap-deduplicated allocation is 25.176 hours, complementing its summed process allocation. Detailed launch intervals and device occupancy remain in the resource ledger.

The separate CloseFridge total includes screening and calibration, with 800 scored attempts and 32 tails in its formal stage. ACT includes 336 formal and 32 calibration executions. E4's eight terminal roles comprise two existing inputs, two selective completions, and four references; its wider repair history includes 22 pre-repair episodes and four canaries. Appendix D.2 separates these measured components from unavailable human/setup effort. OpenDrawer episode counts exclude two preserved pre-reset renderer failures. RoboMME's source checks use controlled stand-ins, while the environment check ends after four real simulator resets. Dashes for its GPU active time and that of the known-path extension indicate unavailable instrumentation, with recorded wall times retained.

E.5 Existing assets and licenses

Robot platforms supply the evaluation interfaces and task settings used here (Mandlekar et al., 2022; Liu et al., 2023; Nasiriany et al., 2024; Tao et al., 2024; Chen et al., 2025; Cadene et al., 2026);

complementary audits examine shortcuts and redundancy (Jiang et al., 2026; Navasardyan & Davtyan, 2026). The study pins public AutoEval, RoboCasa, SimplerEnv, RoboTwin, and RoboDojo revisions under MIT, and vla-eval, LeRobot, and ManiSkill revisions under Apache-2.0. Upstream copyright notices are retained, with our patches, manifests, and analysis code distributed under the README's stated license. The public SimplerEnv video archive is referenced by its hosted immutable revision.