ROBOLEAK: CLAIM-CONDITIONED OVERLAP AUDITING FOR EMBODIED AI BENCHMARKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot-learning benchmarks often reuse objects, scenes, and skills across training and evaluation to probe transfer under controlled changes. Yet a benchmark score alone does not reveal which aspects of the evaluated behavior are novel relative to the training collection. We introduce ROBOLEAK, a claim-conditioned auditing framework that first records cross-boundary relations at five levels—stored files, decoded observations, near-duplicate views, object and scene structure, and task semantics—and then applies a declared *evaluation contract* specifying which observed relations violate the benchmark's intended novelty. Auditing LIBERO with ROBOLEAK shows that all seven Goal-task object models already appear in the training suites, while no scene composition is repeated; under a strict novelty rule a single familiar task is excluded, and this one exclusion raises SmolVLA's score from 76% to 80%, leaves X-VLA unchanged at 100%, and slightly lowers VLA-JEPA's score. The same cleaning moves different models in opposite directions because it removes a task each model finds differently hard. A complementary controlled exposure study finds no consistent average gain from exact-task demonstrations, while revealing substantial seed-level variation under a fixed training budget. ROBOLEAK turns a benchmark score into an answerable question: what was shared, what was required to be new, and which tasks does the number average over?

1 INTRODUCTION

Suppose a robot learns to put a bowl on a plate. At test time, the bowl starts in a different place. This task can test whether the robot handles a new arrangement while using familiar objects. A test of unfamiliar objects would require a different bowl. Both are useful tests, but they answer different questions. To understand a robot's score, a reader needs to know which parts of the test are new and which parts are familiar.

Robot datasets and simulation benchmarks reuse objects, environments, and skills to study how learned behavior transfers to new situations (Open X-Embodiment Collaboration et al., 2023; Liu et al., 2023; Mees et al., 2022). That reuse ranges from the same image to different tasks that both involve placing an object, and each form provides a different kind of familiarity. Finding the shared content is the first step; deciding whether it conflicts with the purpose of the test is the next. The central distinction is therefore between overlap evidence and evaluation invalidity: the former is an empirical relation between collections, whereas the latter depends on what the benchmark claims to hold out.

Concretely, RoboLeak audits a single directed boundary — one training collection versus one evaluation collection — and reports its findings at five levels: files, decoded images, near-duplicate views, objects and scenes, and task instructions. The evaluator then states what the target tasks must introduce (for example, a new object or a new action pattern); we call this rule an evaluation contract. Figure 1 shows the procedure.

Removing tasks also changes what a score averages over. A robot can gain points without improving: SmolVLA scores 76% on the ten Goal tasks, and 80% after the strict rule removes the stove task, which it fails on six of ten initializations. The remaining tasks are chosen only by shared content, not by how the model performs. To show what the new average is worth, RoboLeak compares the selected test set with every other subset of the same size, using the same recorded model outcomes:

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.

the ten possible nine-task subsets average 76%, and the selected subset lies at the extreme end of this reference distribution, with exact two-sided subset probability $p = 0.10$.

Our main study compares LIBERO's Long, Object, and Spatial collections with its ten Goal tasks. All seven digital object models used in Goal appear among the 29 source object models, while their scene compositions differ. The choice of task rule also matters: excluding familiar broad skills leaves one Goal task, whereas excluding a familiar action-and-state-change pattern leaves nine.

We also examine CALVIN, where a new test environment retains the same 34 task definitions as the training environments. This gives a second example of how a test can change one part of the problem while keeping others familiar. Separately, we fine-tune SmolVLA with different numbers of demonstrations of one exact task, keeping the total training data and training steps fixed; across three training seeds, individual-seed success changes span $+8$ to -6 percentage points around a small pooled average change. Together, these studies turn shared-content records and a declared novelty rule into a retained-task list and a score whose rank among equally sized subsets is answerable.

In summary, this work contributes: (i) a claim-conditioned *evaluation contract* that maps observed reuse to task eligibility, derived from the benchmark's stated generalization claim before audit outcomes are inspected (Section 3.2); (ii) a five-level typed relation inventory—stored files, decoded observations, near-duplicate views, object and scene structure, and task semantics—with explicit evidence and blinded review rules for each relation; (iii) equal-count reference subsets with exact finite-population probabilities that rank each cleaned score among its equally sized alternatives; and (iv) a multi-level audit of a directed LIBERO boundary, with external profiles of CALVIN and DROID-100 reported at accepted-evidence level.

2 Related Work

Data and environments for learning robot skills. Robot learning draws on collections of demonstrations and reusable test environments. Open X-Embodiment brings together robot experience, while Octo and RT-2 use robot data or web-derived knowledge to learn transferable behavior (Open X-Embodiment Collaboration et al., 2023; Octo Model Team et al., 2024; Zitkovich et al., 2023). DROID provides real-world manipulation demonstrations (Khazatsky et al., 2024). LIBERO, CALVIN, and RoboCasa provide tasks, objects, and environments for studying performance in new situations (Liu et al., 2023; Mees et al., 2022; Nasiriany et al., 2024). RoboLeak examines these reusable elements across a chosen pair of collections. It distinguishes, for example, a familiar object placed differently from an exact repeat of an observation.

Finding repeated content and checking its effect. Studies of language models inspect whether training and test data share text and examine the effects of removing repeated material (Brown et al., 2020; Lee et al., 2022). Related work also studies overlap that survives changes in wording (Yang et al., 2023), and training on the test task itself is a further form of task-relevant reuse that confounds relative evaluations (Dominguez-Olmedo et al., 2025). For images, CLIP retrieves likely duplicates, inspects them, and compares evaluation results on subsets separated by overlap (Radford et al., 2021); for vision-language models, recent work detects contaminated checkpoints through multi-modal semantic perturbations (Park et al., 2026). This sequence of finding, checking, and separating shared content is the closest connection to RoboLeak. We extend the audit to images, objects, scenes, and tasks within robot data and tie the resulting exclusions to the test's stated novelty requirement.

Testing behavior under changed conditions. LIBERO-Plus, VLA-Arena, and recent benchmark analyses examine robot performance as evaluation conditions change, including visual conditions, instructions, and tasks (Fei et al., 2025; Zhang et al., 2026; Jiang et al., 2026). Work on embodied task language examines how much instructions vary in wording and structure (Wanna et al., 2026). Prior benchmark analyses primarily ask how policies behave under modified evaluation conditions; RoboLeak instead asks whether a relation observed across a declared training-to-evaluation boundary conflicts with the benchmark's own generalization claim. It therefore reports shared content for a chosen boundary and connects it to the tasks that survive the novelty rule.

Table 1 highlights the combination studied here: cross-boundary content auditing, claim-conditioned exclusion, and equal-count score controls.

Table 1: **Comparison of reported study features.** RoboLeak connects overlap evidence to claim-conditioned exclusions and equal-count score controls.

Work	Content audit	Asset/scene reuse	Robot rollouts	Perturbed rollouts	Claim-based exclusion	Equal-count controls
Text deduplication (Lee et al., 2022)	✓	n/a	n/a	n/a	✗	✗
CLIP audit (Radford et al., 2021)	✓	n/a	n/a	n/a	✗	✗
Embodied language (Wanna et al., 2026)	✓	✗	✗	n/a	✗	n/a
LIBERO-Plus (Fei et al., 2025)	✗	✗	✓	✓	n/a	n/a
RoboLeak (ours)	✓	✓	✓	n/a	✓	✓

✓ reported; ✗ not included; n/a outside the evaluation setting. Reuse means cross-boundary asset/scene identity; claim-based exclusion follows a novelty requirement; equal-count controls remove the same number of tasks.

3 Claim-Conditioned Overlap

RoboLeak checks what the source and target collections share, then decides whether that sharing requires removing a target task. A familiar bowl illustrates the distinction: the same bowl matters for a test of new objects, while a test of new arrangements can permit it. Section 3.1 defines the shared-content records; Section 3.2 turns them into task decisions through an evaluation contract. Section 4 describes the detectors and review process that produce these records.

3.1 Comparison units and accepted relations

These records start with the boundary. We choose a source collection a and a target collection b, written as the directed boundary $B = (a, b)$; the direction records that sharing is measured from the training side toward the evaluation side. An episode is one recorded attempt at a task. It provides files, image frames, scene objects, and a task description. Each item keeps its link to its episode or task. For relation r, the sets $\mathcal{U}_a^r$ and $\mathcal{U}_b^r$ contain the items being compared: frames for image matching or digital object models for asset matching.

Across these items, the checks fall into five levels: identical files (L0), identical decoded images (L1), near-duplicate views (L2), shared objects or scenes (L3), and related tasks (L4). The relation set $\mathcal{R}$ includes separate checks for object identity, scene composition, and layout within L3, and the stated task rules within L4. Checks establish exact matches or supply candidates for review. Confirmed source-target pairs form $E_r \subseteq \mathcal{U}_a^r \times \mathcal{U}_b^r$, with their scores and supporting evidence.

For each relation, we count confirmed pairs and the target items they cover. One summary is coverage: what fraction of the target items has at least one source match:

$$R_r(a \to b) = \frac{1}{|\mathcal{U}_b^r|} \sum_{u \in \mathcal{U}_b^r} \mathbf{1}[\exists v \in \mathcal{U}_a^r : (v, u) \in E_r]\,. \tag{1}$$

Each target item contributes once to coverage, even if several source items match it. For example, seven shared object identities cover all seven Goal assets but only seven of the 29 source assets: coverage is directional, and the two numbers use different denominators.

3.2 Evaluation contracts and task eligibility

Before inspecting audit outcomes, the evaluator derives from the benchmark's stated generalization claim which relations must be novel in the target tasks. We express this rule as an evaluation contract $C = (B, \mathcal{V}_C)$, where $\mathcal{V}_C \subseteq \mathcal{R}$ lists the shared relations that trigger removal. A new-object test includes asset identity in this list. A new-arrangement test can allow the same assets while checking layout. The same confirmed match can therefore lead to different decisions under different test requirements.

These decisions apply at the task level: removal operates on whole tasks. Let $\mathcal{T}_b$ be the target tasks and $\mathcal{U}_b^r(t)$ the items associated with task t. A matched frame points to its parent task; a shared object

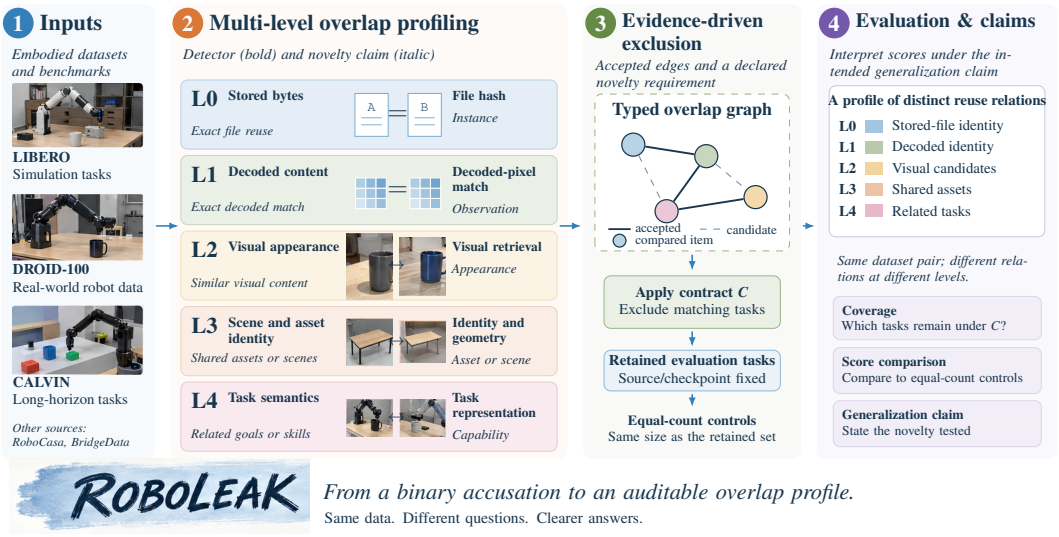


Figure 1: From reuse evidence to evaluation. Detectors establish exact relations or retrieve candidates; acceptance decisions form typed edges; a declared contract selects the edges that require exclusion. This study evaluates the resulting target-task subsets and their equal-count controls. Robot and object thumbnails are schematic illustrations.

can appear in several tasks. We retain a task when every associated confirmed match is permitted by the contract. The retained set and its fraction of the original tasks are

$$\mathcal{T}_b^C = \{t \in \mathcal{T}_b : \nexists\, r \in \mathcal{V}_C,\ u \in \mathcal{U}_b^r(t),\ v \in \mathcal{U}_a^r \text{ with } (v,u) \in E_r\}, \quad \kappa(C) = \frac{|\mathcal{T}_b^C|}{|\mathcal{T}_b|}. \tag{2}$$

A single conflicting match removes a task. Multiple matches to that task still count as one removal. The fraction $\kappa(C)$ reports how much of the test remains. A cleaned evaluation score is defined only when at least one target task remains.

Retained sets are monotone in the strictness of the match predicate: if a stricter matching relation confirms only a subset of the pairs confirmed by a broader one, using the stricter relation cannot remove more tasks. Formally, if every relation $r' \in \mathcal{V}_{C'}$ has a counterpart $r \in \mathcal{V}_C$ with $E_{r'} \subseteq E_r$, then $\mathcal{T}_b^{C'} \supseteq \mathcal{T}_b^C$ and $\kappa(C') \geq \kappa(C)$. The LIBERO views instantiate this ordering: sharing an action/state template implies sharing a broad skill, so the strict rule retains at least as many tasks as the broad rule (nine versus one), and the object-target-aware view, which adds entity-role constraints on top of the strict template, confirms no task pairs and retains all ten.

For example, suppose a target task uses a bowl model found in the source, has a new scene composition and layout, and has zero confirmed repeated observations. An unseen-object contract removes the task because the bowl identity matches. An arrangement contract checks the layout instead and can retain it. A duplicate-only contract also retains it under these conditions. The records stay the same throughout; the evaluator changes the question that the task must answer. Later sections refer to the evaluation contract as a *task rule* or *novelty rule*; the equal-count reference subsets introduced next are also called *controls*.

We construct these contracts from the benchmark's own stated claims, using only claim text and excluding audit outcomes. Three blinded passes structure six quoted claims; a per-level majority combines their decisions. LIBERO describes each suite's transfer objective; this directed boundary requires an explicit exclusion rule. We therefore declare broad-skill, strict action/state-template, and object-target-aware views. Appendix H gives the construction, agreements, and decision records.

4 ROBOLEAK AUDIT PROTOCOL

Figure 1 shows the four stages of the audit. The first three subsections describe how records are produced: exact comparison of files and decoded images (Section 4.1), visual retrieval with blinded

review (Section 4.2), and object, scene, and instruction relations (Section 4.3). The final subsection turns those records into numbers: task scoring with equal-count reference subsets and the two uncertainty measures that accompany the reported scores (Section 4.4).

4.1 Exact files and decoded observations

L0 compares files byte by byte. L1 opens the image files, converts them to RGB, and compares their dimensions and pixel values. This catches identical images stored with different encodings. If one container stores both source and target records, we compare the individual observations inside it. Each result states which files or frames were checked. Appendix D gives the procedures.

4.2 Visual retrieval and acceptance

L2 starts by finding image pairs worth checking. We rank pairs using perceptual image hashes (Zauner, 2010) and pretrained DINOv2 image features (Oquab et al., 2024), then review the union of their top-200 lists. A pair proposed by both methods is reviewed once. Combining the lists covers pairs favored by either representation.

Review asks whether a pair shows the same original rendered observation after an allowed crop, resize, compression, or photometric change. Two views of the same bowl can still be different observations. Two blinded passes review the images, and a third resolves disagreements. Detector names, distances, ranks, and the other pass's decisions are hidden. Confirmed pairs enter E_{visual}. We also test transformed copies with known originals and inspect extra pairs near the retrieval threshold and outside the candidate lists. Recovery of known copies measures robustness to those transformations. Appendix G gives the sample sizes and decisions.

Acceptance is kept separate from retrieval because a single distance cutoff does not transfer across encoders: the same threshold admits different fractions of known transformed copies under pHash and each DINOv2 variant, and the two DINOv2 variants share only 17.6% of their top-100 pairs (Appendix J). The protocol therefore fixes the decision rule and leaves the threshold to control only which pairs are inspected. Because top-k retrieval defines an enriched candidate pool rather than an exhaustive pairwise scan, zero accepted pairs means no verified duplicate was found within the reviewed pool, not an estimate of a global zero-duplicate rate.

4.3 Scene components and semantic views

L3 asks whether scenes use the same object models, contain the same types and numbers of objects, and arrange them in the same way. Official identifiers and asset content establish object identity. Scene composition counts fixtures and objects by type; layout records their regions and initial spatial relations. Moving a bowl changes layout. Adding a second bowl changes composition. We check these relations separately and use multiset Jaccard similarity to measure partial type sharing, counting repeated instances.

L4 compares what the robot is asked to do. We describe each instruction by its broad skill, detailed action/state pattern, manipulated object, target object, relation, and state change. Two blinded passes assign these fields from anonymized task text; a third resolves disagreements. The broad rule compares skill families. The strict rule compares action/state templates, and the object-target-aware rule also compares object roles. For example, "turn on the stove and put the moka pot on it" and "turn on the stove" share stove activation. A test requiring a new action/state template excludes that shared pattern.

4.4 Scores, equal-count controls, and uncertainty

We score each task on ten fixed initializations and average task success rates. Write this score as $S_m(\mathcal{T})$ for model checkpoint m and task set $\mathcal{T}$. If the contract removes k of N tasks, we score all $\binom{N}{k}$ possible removals of k tasks; these equal-count *reference subsets*, indexed by z and written $\mathcal{T}_{C,z}^{\text{match}}$, match the selected subset in task count. We report the reference mean minus the selected score:

$$\Delta_{m,C} = \mathbb{E}_z\left[S_m(\mathcal{T}_{C,z}^{\text{match}})\right] - S_m(\mathcal{T}_b^C). \tag{3}$$

A positive $\Delta_{m,C}$ means the selected tasks score below the reference mean; a negative value means they score above it. All scores use the same recorded outcomes. The comparison changes which tasks are averaged.

For example, the SmolVLA strict subset scores 80% against a 76% reference mean, giving $\Delta = 76 - 80 = -4$ percentage points. The negative sign indicates higher success on the selected tasks. It describes which task outcomes entered the average while the model stayed fixed.

For ten Goal tasks, removing one gives ten possible nine-task subsets. Comparing them shows whether the contract leaves an unusually easy or difficult subset. This reference fixes task count while allowing difficulty to vary. We add a descriptive difficulty comparison and use a separate training experiment to study exact-task demonstrations.

We report two kinds of uncertainty for these scores. The first asks how unusual the selected score is among equally sized task subsets. Let $\mathcal{Z}_C$ index these subsets, $q_z = S_m(\mathcal{T}^{\text{match}}_{C,z})$ be each reference score, $\mu = |\mathcal{Z}_C|^{-1} \sum_z q_z$ their mean, and $s_C = S_m(\mathcal{T}^C_b)$ the selected score. We count the fraction of references at least as far from the mean as the selected score:

$$p_C = \frac{1}{|\mathcal{Z}_C|} \sum_{z \in \mathcal{Z}_C} \mathbf{1}[|q_z - \mu| \geq |s_C - \mu|] \,. \tag{4}$$

The selected subset belongs to the reference set and ties count as equally extreme, so with ten tasks and one removal even the most extreme score has $p_C = 0.10$.

The second resamples retained tasks with replacement, keeping each task's ten initialization outcomes together, and recomputes mean success; the middle 95% of 10,000 replicates gives the interval. It therefore reflects task sampling for fixed models and outcomes. The training experiment instead resamples training seeds and paired rollout indices to include training variation.

5 RESULTS

The boundary fixes LIBERO Long, Object, and Spatial as the source and Goal as the target. It covers all 40 task definitions and all 300 source-target scene pairs. The image audit samples 13,544 RGB frames from 451 HDF5 episode containers, and the initial L2 retrieval draws 64 frames per suite, forming $3 \times 64 \times 64 = 12{,}288$ source-target pairs whose two top-200 lists give a 343-pair union for review.

Five novelty rules differ in what counts as shared content. Three are declared contract views (Section 4.3): the broad-skill, strict action/state-template, and object-target-aware rules. Two are baselines: duplicate-only removes confirmed repeated observations or identical scenes, and candidate-only treats DINOv2 retrieval candidates directly as matches. Three public checkpoints, SmolVLA, X-VLA, and VLA-JEPA, are evaluated on ten initializations per task (Shukor et al., 2025; Zheng et al., 2025; Sun et al., 2026), with exclusions and controls reusing identical recorded outcomes within each checkpoint. Appendix H lists the retained sets and Appendix I the checkpoint conditions.

We first report what the source and target collections share (Section 5.1), then which tasks each novelty rule retains (Section 5.2) and how removing one task moves each checkpoint (Section 5.3). The remaining subsections ask how far these conclusions extend: CALVIN keeps familiar task definitions in a new environment (Section 5.4), a controlled exposure study varies the number of exact-task demonstrations (Section 5.5), and a denser visual sample probes retrieval stability (Section 5.6). Two secondary checks, an earlier Goal-Spatial pilot and a DROID-100 cross-domain diagnostic, complete the design (Appendix F). Figure 2 summarizes the main comparisons, and Appendix K shows the associated robot behavior.

5.1 FAMILIAR OBJECTS APPEAR IN DIFFERENT SCENES

Every one of the seven Goal object models also appears in the source. In the reverse direction, these seven models account for 7/29 (24.1%) of the source inventory. We next ask whether those objects also appear in identical scenes.

Of the 300 source-target scene pairs, 240 share fixture or object types; across all 300 pairs, the composition Jaccard similarity ranges from 0 to 0.455. Every pair differs in composition. For

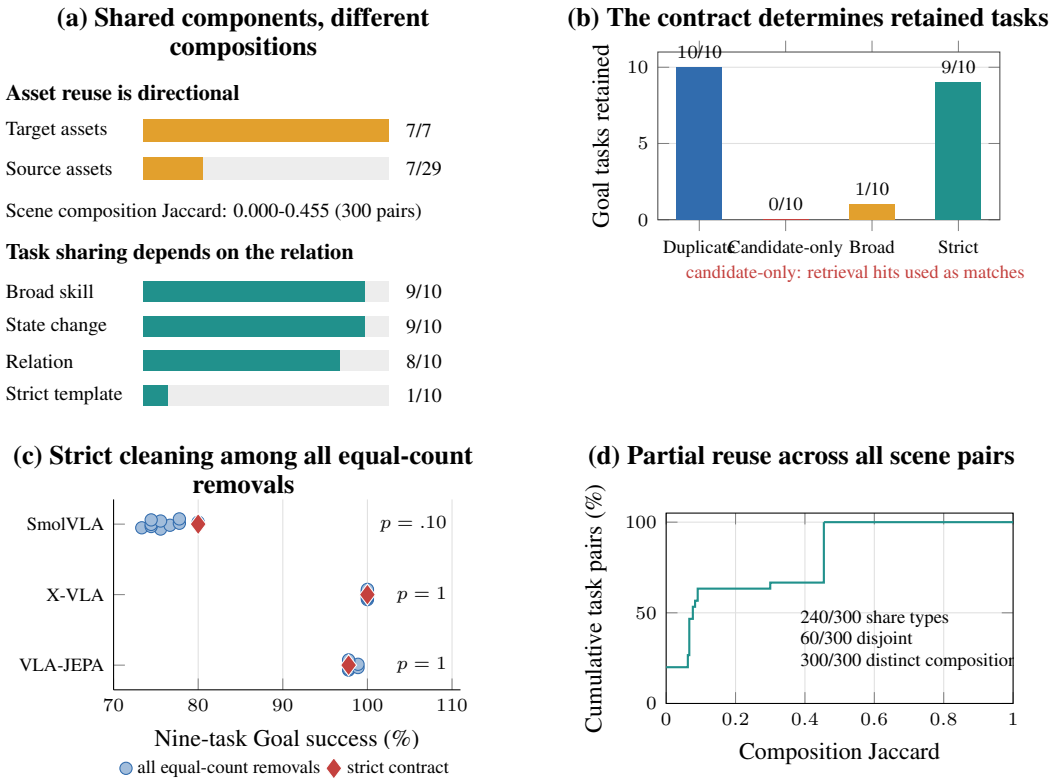


Figure 2: **From shared components to evaluation decisions.** (a) Shared assets cover every target identity but only 7/29 source identities; task coverage depends on the semantic relation. Appendix E reports all projections and exact-match counts. (b) Duplicate-only, broad-skill, and strict exclusions retain 10, 1, and 9 tasks; the candidate-only baseline retains zero. Duplicate-only protects observation and whole-scene novelty while permitting standalone asset reuse; the candidate-only baseline intentionally treats unverified retrieval candidates as matches and is not an overlap estimate. (c) Diamonds mark strict exclusion among all ten equal-count subsets. The exact reference probabilities are 0.10 for SmolVLA and 1.0 for the other checkpoints. (d) Composition similarity includes all 300 task pairs, including the 60 with disjoint type sets. Shared fixtures contribute to the measure; Goal tasks share one composition.

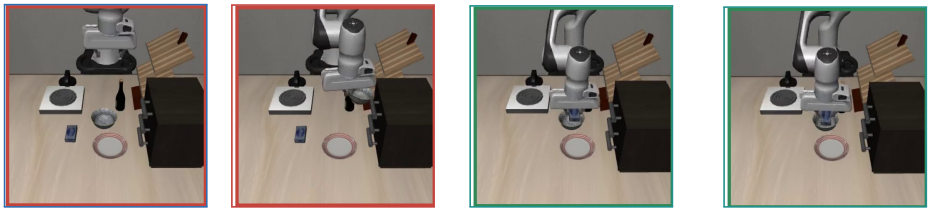
example, the bowl-placement source and stove-activation target share five type instances out of an eleven-instance multiset union. All ten Goal tasks reuse the same fixture set, so these maxima describe one composition. Appendix E gives the object lists and calculation.

5.2 The novelty rule determines which tasks remain

The same shared-content records produce different test sets. The broad-skill rule retains one of ten Goal tasks. The strict action/state-template rule retains nine, changing eight decisions between the two; its single removal is the stove task. The object-target-aware rule additionally compares manipulated and target roles, finds zero matching task pairs, and retains all ten.

The duplicate-only rule retains all ten tasks because the checked scope contains zero confirmed repeated observations or identical whole scenes. Directly using retrieval candidates as matches would remove all ten tasks, because every task occurs in the top-200 DINOv2 list. Review, by contrast, accepts zero pairs in the combined pool, so no task is removed on reviewed visual evidence. This comparison shows why candidate selection and task removal need separate decisions. Tables 11 and 12 give the retained sets.

Turn on the stove (task 6) Removed by the strict rule



Initial (ep. 0) SmolVLA: failure; 4/10 X-VLA: success; 10/10 VLA-JEPA: success; 10/10

Figure 3: **The task the strict rule removes.** Episode 0 of the excluded stove task from a shared initialization: SmolVLA fails while X-VLA and VLA-JEPA succeed, so removing this task raises SmolVLA's mean, leaves X-VLA's unchanged, and lowers VLA-JEPA's. Rates aggregate all ten initializations. Appendix K shows further cases.

5.3 The same exclusion changes scores in different directions

Strict exclusion gives SmolVLA 80.0% success on nine tasks against a 76.0% equal-count reference mean, a Δ of -4 percentage points (Eq. 3). X-VLA remains at 100%; VLA-JEPA reaches 97.8% against 98.0%. The removed task explains these directions. SmolVLA succeeds on four of its ten stove initializations, so removing that below-average task raises its score by four percentage points. X-VLA succeeds on every task, leaving its mean unchanged. VLA-JEPA succeeds on every stove initialization but has a 98% full-suite mean, so removing the stove task slightly lowers its average. Cleaning therefore does not imply score correction in a fixed direction: the score shift is algebraically determined by the removed task's performance relative to the suite mean, not by an improvement of the evaluated policy. Figure 3 shows the removed task from a shared episode-0 initialization: SmolVLA fails, while X-VLA and VLA-JEPA succeed.

For N equally weighted tasks with mean $\bar{s}$, removing task j with success s_j gives $\bar{s}_{-j} = (N\bar{s} - s_j)/(N-1)$. For the same SmolVLA outcomes, $(10 \times 76\% - 40\%)/9 = 80\%$. Averaging all one-task removals recovers the original mean, which explains the reference value of 76%. Appendix I.1 supplies the derivation and all checkpoint calculations.

SmolVLA's strict subset is the highest-scoring of the ten nine-task subsets, with exact $p = 0.10$; X-VLA and VLA-JEPA give $p = 1.0$. Removing the nearest task by observed outcome difficulty gives scores within 2.2 percentage points of strict exclusion for all three checkpoints (Table 18).

5.4 A new environment can retain familiar task definitions

CALVIN tests transfer from environments A, B, and C to environment D. The scene changes, while 2 of the 4 top-level asset files (the environment-definition files at the root of the asset bundle) and all 34 task predicates are shared with training. A task predicate is the rule used to check task completion. This test combines a new environment with familiar task definitions and some shared assets.

The visual sample has median DINOv2 nearest-neighbor distance 0.088; the visual evidence consists of retrieval distances, because the target observations arise during online evaluation. Figure 4a and Appendix D specify this evidence separately from the reviewed LIBERO candidates.

5.5 Sensitivity to exact-task exposure

We next change what the model sees during training. The target is to move a black bowl from a cookie box to a plate. We fine-tune SmolVLA with 0, 11, or 43 demonstrations of this exact task. Each condition contains 389 episodes and uses 10,000 training steps: exact-task demonstrations replace examples from the other nine tasks. The environment family and object roles stay fixed, while the initial spatial relation distinguishes the exact task from related demonstrations.

Each condition uses three training seeds, evaluated on the same 50 initial states. This gives 150 rollouts per condition and 450 in total. Success rates are 22.7%, 21.3%, and 23.3% for 0, 11, and 43 examples. Comparing 43 with 0 gives 0.7 percentage points, with a paired hierarchical 95% interval of [-10.0,11.3]. Figure 4b shows each seed; Appendix B gives the design.

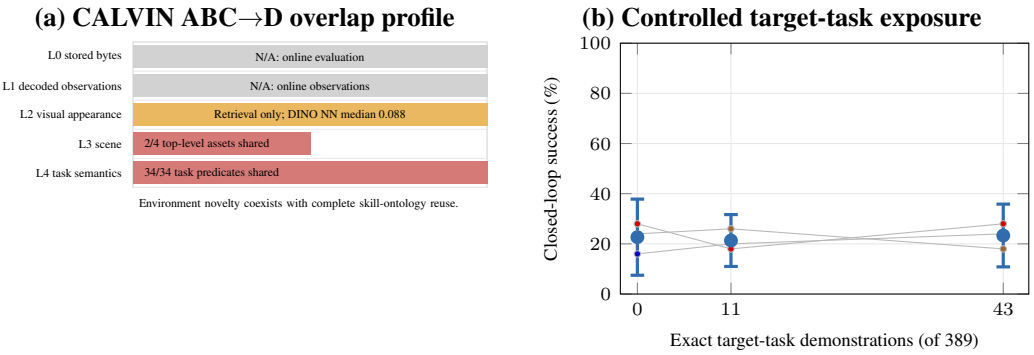


Figure 4: **Interpreting an external boundary and a controlled intervention.** (a) CALVIN D changes the scene configuration while reusing 2/4 top-level asset files and all 34 task predicates. The visual sample reports retrieval distances; cross-split L0/L1 is N/A because target observations arise during online evaluation. (b) Three SmolVLA training seeds per dose share the initialization family, 389-episode training size, optimization budget, environment family, entity roles, action family, and language scaffold. Exact-task exposure changes, including the defining initial spatial relation. Gray lines show individual seeds; blue bars show 95% Student-t intervals over seed success ($df = 2$). The paired hierarchical interval for 43 minus 0 remains the primary estimate of the fine-tuning exposure effect.

The three seeds change by $+8$, 0, and -6 percentage points when moving from 0 to 43 examples, so their different responses average to a small pooled change. These seed-level changes measure the fine-tuning-stage effect of replacing related demonstrations with exact-task examples for this pretrained initialization. With three seeds and one target task, this experiment does not establish a monotonic or average benefit of exact-task exposure; it shows that replacing related demonstrations with exact-task examples can change individual-seed outcomes under a fixed training budget.

5.6 Similar retrieval summaries can contain different image pairs

The denser sample contains 256 frames per suite, giving $3 \times 256 \times 256 = 196{,}608$ source-target pairs. DINOv2-Base and Small share 30 pairs in a top-100 union of 170, giving 17.6% Jaccard similarity. Yet nearly all top-ranked entries come from the same source suite (Figure 8a, Appendix J). Similar source distributions can contain different image pairs. Raising the distance threshold recovers more known transformed copies and admits 104 reviewed real pairs, all labeled as different observations (panel b). Figures 9 and 10 show the complete curves, and Appendix G works through one accepted and one rejected candidate from the earlier pilot.

5.7 What a benchmark release should report

The audit procedure doubles as a reporting standard: a benchmark release benefits from stating its directed boundary between training and evaluation, the declared novelty claim in claim-anchored terms, the relation inventory with accepted-evidence counts at each level, and equal-count reference statistics that turn a cleaned score into a rank. With these statements, a reported score becomes an auditable claim about what was shared and what was required to be new.

6 Conclusion

RoboLeak checks what robot-data collections share and relates those matches to the purpose of a test. In LIBERO, familiar objects appear in different scenes, different task rules retain different test sets, and a task's difficulty explains how its removal changes the score. CALVIN shows how a new environment can keep familiar task definitions. RoboLeak turns a benchmark score into four answerable questions: what was shared between the compared collections, what the test required to be new, which matches triggered removal, and where the new score ranks among equally sized alternatives. Appendix A discusses these findings and details the scope of the structural, visual, and behavioral evidence.

ETHICS STATEMENT

All data used in this work come from public datasets and simulated evaluation rollouts. Raw media are redistributed only when licenses permit; released evidence otherwise defaults to identifiers, aggregates, and fingerprints. Because overlap labels can be reputationally sensitive, we distinguish direct duplicates, expected skill reuse, and uncertain inferred similarity, and avoid claims about unverifiable proprietary corpora.

REPRODUCIBILITY STATEMENT

The supplement provides aggregate results, analysis and detector code, and software versions. Appendix D specifies experimental boundaries and sampling rules so that the reported comparisons can be reproduced.

AI USE STATEMENT

The authors used AI-based writing assistance for language editing and manuscript formatting. All research content—the audit design, detector and review protocol, contract construction, experiments, statistical analyses, and all reported numbers—was produced and verified by the authors, who take full responsibility for the content of this submission.

REFERENCES

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Ricardo Dominguez-Olmedo, Florian E. Dorner, and Moritz Hardt. Training on the test task confounds evaluation and emergence. In *International Conference on Learning Representations*, 2025.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Tianchong Jiang, Xiangshan Tan, Samuel Wheeler, Luzhe Sun, Tewodros W. Ayalew, and Matthew Walter. What are we actually benchmarking in robot manipulation? In *Conference on Robot Learning*, 2026.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2021. doi: 10.1109/TBDATA.2019.2921572.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, 2023.

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

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024.

Octo Model Team, Dibya Ghosh, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

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

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaa El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Jaden Park, Mu Cai, Feng Yao, Jingbo Shang, Soochahn Lee, and Yong Jae Lee. Contamination detection for VLMs using multi-modal semantic perturbations. In *International Conference on Learning Representations*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

Selma Liliane Wanna, Agnes Luhtaru, Jonathan Salfity, Ryan Barron, Juston Moore, Cynthia Matuszek, and Mitch Pryor. Limited linguistic diversity in embodied AI datasets. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11058–11086. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.507.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*, 2023.

Christoph Zauner. Implementation and benchmarking of perceptual image hash functions. Master's thesis, Upper Austria University of Applied Sciences, 2010.

Borong Zhang, Jiahao Li, Jiachen Shen, Yuhao Zhang, Yishuai Cai, Lu Liu, Hailu Ji, Yuanpei Chen, Juntao Dai, Jiaming Ji, and Yaodong Yang. VLA-Arena: An open-source framework for benchmarking vision-language-action models. *arXiv preprint arXiv:2512.22539v4*, 2026. Accepted at ICML 2026.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

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