zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

Distribution Mismatch, Lighting, and Background Generalization in a Low-Cost Imitation-Learning Pick-and-Place Policy

Iliya Radulov[1]

[1]Independent researcher

October 2026

Abstract

We report four quantified failure modes and their diagnoses from an extended imitation-learning deployment on a low-cost SO-101 robotic arm trained with the Action Chunking Transformer (ACT) policy. First, a 120-episode dataset collected under a high-mounted overhead camera produced a policy with zero rollout successes despite smooth, healthy training-loss convergence; frame-by-frame review of the training footage, not the loss curve, revealed that the operator's own hand was visible in a large fraction of frames during teleoperation – a camera-position artifact absent at autonomous inference time. Repositioning the camera and re-recording resolved the failure entirely. Second, a systematic 300-episode, five-background experiment shows that "background generalization" is not one phenomenon but at least three mechanistically distinct failure modes with different remedies: a pattern-distractor effect from high-contrast patterned surfaces, a glare/specular-reflection effect from glossy surfaces that is lighting-responsive, and a camera auto-exposure instability on very low-reflectance surfaces that we confirm is a platform-level hardware limitation, not a model or data problem. Third, we quantify the role of ambient lighting directly with lux measurements and show that under- and over-illumination produce two qualitatively different failures – a localization failure and a precision/glare failure respectively – separated by a measured threshold near 400 lux. Fourth, we report a final, statistically confirmed baseline (86.3% success over 80 trained-position trials, 95% Wilson CI [77.0%, 92.1%]; 53.6% over 56 novel-position trials, CI [40.7%, 66.0%]) alongside an honestly unresolved anomaly: a single training-grid position at which an untrained test object is almost never detected, while performing normally two centimeters away. We report each finding with its original diagnostic evidence and distinguish confirmed causal mechanisms from open, documented hypotheses throughout.

1 Introduction

Imitation-learning policies trained on a handful of demonstrations tend to fail in ways that look, from the outside, like a single undifferentiated "it doesn't generalize." In practice, on a physical low-cost manipulation platform, several independent and mechanistically distinct problems can produce superficially similar symptoms: a policy that never succeeds, a policy that succeeds inconsistently across physical positions, or a policy that succeeds on one visual background and fails completely on another. Distinguishing these mechanisms matters because they call for different fixes – more data, different data, a camera reposition, a lighting change, or in one case documented here, no available software fix at all.

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

Project context. The work reported here is part of a single-operator project to build a reproducible, low-cost manipulation testbed around the open-source SO-101 arm and the LeRobot framework. The project began on 5 August 2026 with a first autonomous pick-and-place success using ACT, and has since branched into a study of trigger-based attacks on robot policies [3], the policy-quality problems documented here, and preparation for moving from ACT to a language-conditioned vision-language-action model (SmolVLA) and to additional hardware platforms. ACT served throughout as the baseline policy; the findings below are the problems we had to diagnose before a comparison with that next generation of policies would be meaningful. A dated timeline is given in Section 2.

This paper documents four such findings from a roughly seven-week deployment (5 August–22 September 2026) of ACT [1] on a self-built SO-101 arm using the LeRobot framework [2]. All four findings share a common methodological thread: in each case, an aggregate success-rate number or a training-loss curve initially suggested one explanation, and a more granular analysis – frame-by-frame video review, a controlled background sweep, direct lux measurement, or a fine-grained positional probe – revealed a different and more specific one. We report the findings in the order they were diagnosed, together with the evidence that ruled out competing explanations at each step, and we are explicit about which conclusions are confirmed and which remain open.

Our contributions are:

- a documented case in which a policy trained on footage containing an unintended distribution-shifting artifact (the operator's own hand) converged normally in training-loss terms while being functionally useless at inference, diagnosed only through direct video review;
- a controlled, five-background, 300-episode experiment showing that background-dependent failure in this setting decomposes into at least three distinct mechanisms, one of which (camera auto-exposure instability) we confirm is an uncorrectable platform limitation rather than a model issue;
- a direct, lux-metered quantification of lighting sensitivity showing two distinct and oppositely-signed failure modes separated by a measured threshold; and
- a final, statistically powered baseline for the corrected system, reported alongside a genuine, reproducible, and explicitly unresolved small-scale spatial anomaly, rather than omitted or force-fit to a tidy explanation.

2 Platform and Methods

All experiments used a self-built, two-arm SO-101 rig (3D-printed, 6-DOF, STS3215 servos), two USB cameras (an overhead/main view and a wrist-mounted close-up view), and the ACT policy as implemented in LeRobot, trained with the standard LeRobot `lerobot-train` / `lerobot-record` / `lerobot-rollout` pipeline. Recording and inference occurred across several physical machines over the course of the project (a Mac, a Raspberry Pi 5, and an RTX 5060 Ti desktop for training); each finding below states which machine(s) were used in that experiment. Table 1 gives the chronology. Note that the findings are presented by topic, not by date: the background-diversity experiment (Finding 2) was carried out in August, before the Raspberry Pi 5 migration and before the camera repositioning of Finding 1, and therefore used the earlier camera configuration of the rig.

Episodes were recorded at a fixed grid of physical object positions, with objects placed by eye against a gridded cutting mat used as a placement reference in several (but not all) experiments;

Table 1: Timeline of the work reported here (all dates 2026).

Dates	Machine	Activity	Section
5 Aug	Mac (MPS)	First autonomous pick-and-place success (30 episodes, 5,000 steps)	–
8–9 Aug	Mac (MPS)	Text- and visual-trigger poisoning experiments (companion study)	–
11 Aug	Mac (MPS)	Position-balanced policy (120 episodes); precision weakness near the target first observed	–
18–19 Aug	Mac; cloud GPU	Sparse position column fixed with targeted data; white-background test ≈0% success; first cloud SmolVLA pipeline test	§4
20–23 Aug	Mac (MPS)	Background-diversity experiment: 300 episodes recorded, 80,000-step training, evaluation	§4
5 Sept	Pi 5	Migration to Raspberry Pi 5; new cameras and mounts installed	§3
7 Sept	Pi 5; Mac	Dense 120-episode dataset recorded; policy gets 0 successes on both machines; footage review finds the operator's hand in frame	§3
18–20 Sept	Pi 5	Camera stand redesigned and repositioned; pre-flight checks; 96 episodes re-recorded (20 Sept)	§3
20–21 Sept	ML desktop; Pi 5	Training (20,000 and 50,000 steps); rollout confirmation and lux measurements (21 Sept)	§5
22 Sept	Pi 5	Red-sample and novel-position passes	§6

coordinates reported below are the mat's own grid coordinates in centimeters unless stated otherwise. Beginning with the recordings described in Section 3, all recording commands used LeRobot's streaming-encoding flags (`--dataset.streaming_encoding=true`, `--dataset.encoder_threads=2`, `--dataset.num_image_writer_processes=0`, `--dataset.num_image_writer_threads_per_camera=2`), which had previously been identified as the fix for an unrelated `BrokenProcessPool` crash during episode encoding and were applied as standard practice from that point onward.

Success rates are reported as observed trial counts together with 95% Wilson score confidence intervals [4], computed wherever a specific, fixed number of trials at a specific condition is available. As in prior work on this platform, the Wilson interval is preferred over a normal approximation because several conditions here involve small trial counts or success rates near 0% or 100%, where a normal approximation is unreliable. Where a result is reported only as a percentage in a summary table (as in Table 3), the underlying fraction is given in the text or in the experiment's own logs, and we do not compute a CI for every cell; intervals are given for the headline aggregate figures.

3 Finding 1: A Camera-Position Artifact Produced a Zero-Success Policy Despite Normal Training Convergence

3.1 Initial result and the misleading signal

An early 120-episode dataset was recorded using a high-mounted overhead camera – a design inherited from an official LeRobot/XLeRobot mounting pattern, originally adopted in anticipation of future cross-hardware compatibility with a planned mobile-base platform. An ACT policy trained on this dataset for 20,000 steps converged smoothly, reaching a final loss of 0.157 (2.23 epochs), with a loss curve indistinguishable in shape from every other successful training run in this project. On autonomous rollout, however, the policy achieved **zero successes**, with a characteristic "loses the plot" behavior: the arm would begin a plausible motion, then stall mid-episode and restart from a different, unrelated motion, without ever completing the task.

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

3.2 Ruling out inference hardware

Because the Pi 5 was a relatively new addition to the pipeline, the first hypothesis was an inference-hardware problem specific to that device. The identical checkpoint was re-run on the Mac rather than the Pi 5. The Mac rollout reproduced the identical "loses the plot" behavior, which ruled out Pi 5-specific inference hardware as the cause and pointed instead at the training data itself.

3.3 Diagnosis by direct video review, not by metrics

Frame-by-frame review of the original teleoperation footage – not any further change to training hyperparameters or inference configuration – identified the actual cause: the high overhead camera mount captured the operator's own hand and forearm in frame during a substantial fraction of each teleoperated demonstration. This is a genuine train/inference distribution mismatch: no hand is present during autonomous rollout, so a policy that had partly learned to condition its behavior on the hand's presence and position had no equivalent signal available at inference time. The same high mount was also found to cause occasional physical collisions between the arm and the camera stand in certain poses, and the wrist camera's fisheye lens was separately found to add distortion to the close-up view.

This diagnosis also resolved the original motivation for the high mount: it had been adopted for hypothetical compatibility with a bimanual mobile-base platform that is planned to run a different policy family (a vision-language-action model) entirely, with its own, separate recording pass. The cross-hardware-compatibility requirement was therefore dropped, freeing the camera position to be optimized purely for this single-arm rig.

3.4 Fix and re-record

The camera was repositioned to a front-facing, purpose-built three-leg stand (an off-the-shelf design, selected after a brief custom-CAD detour was judged not worth the time investment), and the wrist lens was replaced with a non-fisheye unit. A pre-flight validation pass confirmed, via test stills, that the operator's hand was no longer visible in frame in a simulated teleoperation pose, and that no arm/stand collision occurred across a set of representative poses.

A new dataset was then recorded across the full position/orientation grid (8 positions × 4 orientations × 3 repetitions) in three batches, recovering from one partial batch failure (a `TimeoutError` during video-encoder finalization that cost a single episode, confirmed via episode-index inspection to be a shutdown-time resource-contention issue rather than mid-recording data loss, and made up by recording one additional episode in the following batch):

Table 2: Re-recorded dataset, after the camera-position fix.

Batch	Episodes	Frames
1	32	18,956
2	31 (of 32 attempted)	18,355
3	33	19,545
Total	**96**	**56,856**

Two training runs were performed on an RTX 5060 Ti desktop (roughly 8× faster than the Mac's MPS backend for this workload, approximately 8.3 steps/sec vs. approximately 1 step/sec): a 20,000-step quick-validation run on batch 1 alone (40m27s, final loss ≈ 0.128), used as a sanity

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

check on the new camera setup before committing to full training time, and a 50,000-step run on the full 96-episode merged dataset (1h40m52s, final loss ≈ 0.112–0.116). Both runs completed without error, in contrast to the original dense dataset, which had also trained without error – underscoring that the camera-framing artifact, not any property of the training pipeline, was the actual cause of the original failure.

3.5 Result

The re-recorded, camera-fixed policy produced coherent, explicable rollout behavior in place of the original incoherent stall-and-restart pattern. The statistically confirmed success rate of this corrected policy is reported in Section 6.

3.6 Why this matters methodologically

This finding is reported in part because of what it did *not* show: nothing in the training loss curve, which converged as smoothly as every other model in this project, distinguished the broken 120-episode policy from a working one. The defect was entirely a property of the recorded visual content, invisible to any training-time metric, and detectable only by watching the actual footage. We return to this point in Section 7.

4 Finding 2: Background Generalization Decomposes into (at Least) Three Distinct Failure Mechanisms

4.1 Motivation

Prior testing of a transfer-learned policy on a plain white/brown paper background – never seen during training – produced essentially 0% success despite substantial prior training, closely matching the very first clean model's total failure on novel backgrounds weeks earlier. This motivated a direct, controlled test of whether background-diverse training data could close this gap, and whether "background generalization failure" was in fact one phenomenon or several.

4.2 Methodology

A 300-episode dataset was recorded: 5 positions (P1–P5, trained) × 4 orientations (0°/45°/90°/135°) × 3 repetitions × 5 backgrounds (white, black, patterned gift-wrap, newspaper, and wood – the last chosen specifically because it had an existing 50–60% baseline from prior training, enabling a before/after comparison), for 60 episodes per background and 168,826 total merged frames. An ACT policy was trained from scratch (deliberately not transfer-learned, to avoid inheriting a grid-pattern-dependent prior from earlier checkpoints) for 80,000 steps on an Apple Silicon MPS backend (27h55m wall-clock; CPU was rejected after a 30-step speed test showed approximately 506s/step). Final loss was 0.133 (L1 0.132, KLD ≈ 0), with KLD converging to approximately zero by step 9,000.

Evaluation used 3 trials per (position, orientation) cell at baseline, escalating to 5–10 when a result was ambiguous, across the five trained positions plus two novel positions, P6 (15,10) and P7 (5,5), neither used in training.

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

Table 3: Background-diversity experiment: success by position and background, as percentage (successes/trials). P1–P5 were part of the training grid; P6–P7 were not. Where two values are shown (white), they are dim / extra light. The camera auto-exposure issue on black is described in Section 4.3.3.

Background	P1	P2	P3	P4	P5	P6 (nov.)	P7 (nov.)
Wood (trained)	85% (17/20)	100% (12/12)	∼100%[a]	**8% (1/12)**	100% (12/12)	33% (4/12)	0% (0/4)
Cutting mat (novel)	92%[b] (11/12)	25% (3/12)	100% (12/12)	0% (0/12)	8% (1/12)	0% (0/12)	0% (0/8)
Newspaper (trained)	100% (12/12)	92% (11/12)	∼100%[c]	**100% (12/12)**	100% (12/12)	33% (4/12)	0% (0/8)
Gift-wrap (trained)	100%	100%	100%	**8% (1/12)**[d]	100%	0% (0/3)	0%
White (trained)	100%	100%	100%	67% / 83%[e]	75% / 100%[e]	33% (4/12)	0% (0/8)
Black (trained)	Disrupted by confirmed camera auto-exposure instability; no clean figure (Sec. 4.3.3).						

[a]12/14 clean plus 2 partial (both at 90°). [b]9/12 clean plus 2 recovered on a second attempt within the same episode. [c]11/12 clean plus 1 partial. [d]Same (1/12) under normal and extra lighting. [e]8/12 → 10/12 (P4) and 9/12 → 12/12 (P5) with extra lighting. Success on a second attempt within the same episode counts as success.

4.3 Results

4.3.1 Mechanism 1: pattern-distractor (not lighting-responsive)

On the cutting mat – the grid-printed surface used by earlier models but deliberately excluded from this experiment's training set, and hence one of the most out-of-distribution backgrounds tested – the dominant failure mode, coded at evaluation time as "failed to find the sample," occurred at P2, P5, P6 and P7 (P4: 0/12). On the busy, non-glossy gift-wrap, which was part of the training set, P4 reached only 1/12 under both normal and extra lighting, and orientation-recognition failures at 90° at P1 likewise persisted under extra light. Because added light did not help on the gift-wrap, we interpret both as the vision backbone being distracted by high-contrast pattern features acting as competing salient cues, rather than as a brightness-related effect. The cutting mat itself was not tested under varied lighting, so for that surface the pattern-distractor reading rests on the failure signature (failure to find the sample) rather than on a lighting control.

4.3.2 Mechanism 2: glare/specular reflection (lighting-responsive)

On the glossy white-paper background, performance at the two hardest positions improved directly with additional lighting: P4 rose from 67% to 83%, and P5 – which specifically showed orientation-recognition failures at 90° under dim light – rose from 75% to 100% once those failures resolved under brighter light. This pattern is the functional opposite of Mechanism 1: it is specifically responsive to lighting, consistent with a glare/specular-reflection effect distorting the camera's read of the sample's precise edges or orientation rather than a pattern-distraction effect.

4.3.3 Mechanism 3: camera auto-exposure instability (a confirmed hardware/platform limitation)

Testing on the black background was disrupted by a directly measured camera auto-exposure artifact: ten sequential frames captured at the start of a black-background session showed a brightness ramp from 25.7 to 121.9 (in the camera driver's own brightness units) over approximately three seconds, with the auto-exposure algorithm visibly "hunting" for a stable setting against a low-reflectance scene. We attempted to disable this behavior directly via OpenCV's `CAP_PROP_AUTO_EXPOSURE` and `CAP_PROP_EXPOSURE` properties across five different parameter combinations; none produced any measurable change in the captured frames, which we take as confirming that these properties are silently ignored by this camera/macOS-AVFoundation combination,

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

rather than being a configuration error on our part. During disruption, the arm's observed behavior (dropping a grasped sample mid-episode, or aborting and retrying immediately after an apparent success) matched a general origin-fallback pattern observed elsewhere in this project whenever the policy encounters high input uncertainty, and the per-episode poses involved were inconsistent, distinguishing this cleanly from a targeted or trigger-like behavior. We explicitly flag this as a confirmed hardware/platform limitation that we were unable to work around in software, not an unresolved question.

4.4 Cross-cutting findings: P4 and P7

Position P7 (5,5) achieved 0% success across every background and lighting condition tested, roughly 40 trials in total (95% Wilson CI [0%, 8.8%]), consistently coded as "failed to find." Because this null result held regardless of background – the one variable this experiment was designed to probe – we interpret it as a camera-coverage/framing problem specific to that position (the arm's own body likely occluding the workspace at that extreme position under the camera configuration used in this experiment) rather than a background-generalization failure, and flag it as the clearest, most actionable single finding of this experiment.

Position P4 (20,10), by contrast, ranged from 0% to 100% depending specifically on background (wood 8%, cutting mat 0%, newspaper 100%, gift-wrap 8%, white 67–83%). Photographic inspection confirmed a real, partial camera-coverage limitation at this position (the object and target are not always reliably within the same field of view), but the wide variance across backgrounds shows this is not an absolute blind spot: newspaper's high-contrast print happened to provide the most robust signal at this specific position, and the position's practical reliability is genuinely background-dependent rather than fixed.

5 Finding 3: Lighting Is Quantifiable, with Two Distinct, Oppositely-Signed Failure Modes

5.1 Methodology

Reflected light intensity was measured directly with a phone lux meter (back-camera-based, chosen because it allows visual confirmation of exactly what is being measured) off six colored test samples, under three lighting conditions (no added light, one added light source, two added light sources), and at a fixed, neutral table position chosen to be roughly equidistant from both light sources.

5.2 Lux by sample color and a position-dependent confound

Table 4: Reflected lux by sample color and lighting condition (near position 5).

Sample	No added light	+1 light source	+2 light sources
Black	25–30	100–110	140–150
Blue	40–45	120–140	200–220
Red	45–50	130–150	210–230
Gray	55–60	160–170	270–290
Yellow	80–90	180–200	310–320
White	100–110	240–260	340–380

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

A table-position dependence was discovered during this measurement: the two LED light sources are physically closer to one end of the workspace, producing measurably higher reflected lux there for any sample, most dramatically for white (rising to roughly 400–420 lux at that end under two added light sources, versus 340–380 lux at the neutral reference position). This single rig property also resolves an earlier apparent discrepancy in black-sample lux readings taken at two different positions in separate sessions; the readings were correct, and the positions differed.

5.3 Two distinct failure modes, separated by direct observation

A targeted retest at the neutral position, with two grasp attempts per sample, found:

Table 5: Lux and grasp result by sample, at a neutral table position (2 trials each).

Sample	Shape	Lux	Result
Black	–	70	0/2
Blue	egg-like	130	1/2
Red	thick disk	120	2/2
Gray	egg-like	140	1/2
Yellow	roughly cubic	220	2/2
White	–	310	2/2

Direct observation distinguished two opposite-signed, mechanistically different failures rather than one "bad lighting" story:

- **Under-illumination → localization failure (black).** The arm failed to visually localize the sample at all, even for an initial approach – a perception/contrast failure, not an imprecision problem.
- **Over-illumination/glare → precision failure (white, above approximately 400 lux).** The arm localized and reached the correct general area but consistently missed narrowly, grasping just beside the sample rather than failing to perceive it. Below approximately 400 lux (e.g., 310 lux in Table 5), white performed identically to every other sample, indicating a fairly sharp threshold rather than a smooth degradation.

A third, independent confound was identified at the same time: egg-like sample shapes (blue, gray) underperformed at moderate, otherwise unremarkable lux values, more plausibly explained by grasp difficulty on a rounded shape than by lighting.

5.4 Shape, object identity, and standardization

Of the three roughly-cubic samples available (black, yellow, white), black was ruled out by its localization failure and white by its position-dependent glare failure, leaving yellow as the standardized training/testing object for the remainder of this work (Section 6).

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

6 Finding 4: A Final Confirmed Baseline, and an Open, Unresolved Spatial Anomaly

6.1 Trained-position baseline

Using the camera-fixed policy from Section 3 and the standardized yellow sample, under lighting matched to recording-time conditions, a 10-trial-per-position confirmatory pass across all 8 trained positions gave:

Table 6: Final trained-position confirmation pass (yellow sample, 10 trials per position).

Position	1	2	3	4	5	6	7	8	**Total**
Success/10	7	10	10	8	8	8	9	9	**69/80**

69/80 (86.3%), 95% Wilson CI [77.0%, 92.1%], with no position collapsing toward zero and the weakest position still clearing 70%. This is the headline confirmation that the camera-position fix (Section 3) and the lighting-matching practice (Section 5) together produce a reliable policy across the full trained grid.

6.2 Novel-position generalization

An initial single-trial pass at 8 untrained coordinates gave 4/8 (50%), explicitly flagged at the time as too small a sample to trust. A follow-up pass at 7 trials per point, same coordinates, same yellow sample, gave:

30/56 (53.6%), 95% Wilson CI [40.7%, 66.0%].

The aggregate rate is consistent between the one-trial and seven-trial passes, but several individual points flipped outcome entirely between passes (two points went from apparent failure to majority success, one went from apparent success to majority failure), which we take as a concrete, measured illustration that single-trial evaluation at a given condition is not a reliable estimate, even when the aggregate across many such points happens to look consistent.

6.3 An open anomaly: a sharp, small-scale blind spot

A parallel pass using an *untrained* object (a red, disk-shaped sample, never seen during training) at the same trained and novel positions gave a markedly different and partially inverted pattern:

Table 7: Red (untrained object) vs. yellow (trained object), at the same 8 trained positions (10 trials each).

Position	1	2	3	4	5	6	7	8	**Total**
Red/10	9	8	5	**1**	4	3	9	8	**47/80**
Yellow/10	7	10	10	8	8	8	9	9	**69/80**

Red's overall rate, **47/80 (58.75%), 95% Wilson CI [47.8%, 68.9%]**, is lower than yellow's as expected for an untrained object, but the *shape* of the pattern is what is notable: outer positions (1, 2, 7, 8) remain strong (80–90%) while inner positions (3–6) collapse, with position 4 nearly failing outright (1/10). Direct observation separated this into two independent problems overlapping in the same table: the outer-position weakness is a mechanical grasp/geometry difficulty with the disk

shape (the camera correctly localizes the sample; the arm struggles to get a stable grip), while the inner-position weakness, and position 4 in particular, is a genuine non-detection/perception failure.

At novel positions, red dropped further, to **7/40 (17.5%), 95% Wilson CI [8.7%, 32.0%]**, completing a full 2×2 (object $\times$ position-familiarity) comparison:

Table 8: The full 2×2 comparison: object familiarity $\times$ position familiarity.

	Trained positions	Novel positions
Yellow (trained object)	86.3% (69/80)	53.6% (30/56)
Red (untrained object)	58.75% (47/80)	17.5% (7/40)

The two "single-disadvantage" cells (yellow at novel positions, red at trained positions) land close to one another (53.6% vs. 58.75%), consistent with novel-object and novel-position difficulty being of roughly comparable and additive magnitude, while the "both disadvantages" cell (red at novel positions) is markedly the worst.

The single most striking and still-unexplained data point in this investigation is a direct, sharp spatial discontinuity: the red sample was essentially undetected at trained position 4 (1/10) but succeeded in 4 of 5 trials at a novel position only approximately two centimeters away. This was independently re-confirmed by repeating the direct comparison between the two points. Three candidate explanations were tested and each ruled out as a complete account: relocating the physical light source near position 4 (reflected lux stayed in a similar 110–140 range, no material change); boosting light intensity at the two weakest inner positions (helped the first attempt inconsistently, with no reliable repeat effect); and flipping the sample's orientation, exploiting a known asymmetry in its two faces (result inconclusive). Switching back to the trained yellow sample at the same positions reconfirmed normal performance, ruling out a general rig-level or policy-degradation explanation. The leading, *untested* hypothesis is a narrow camera-viewing-angle/specular-reflection interaction specific to this object's flat face, present at this exact position and not at an immediately adjacent one, and not showing up as a large change in surface-reflected lux (which measures light leaving the sample, not what the camera receives at a specific viewing angle). A second, independent and likewise untested hypothesis – that part of red's broader novel-position underperformance may be a grasp-height/timing imprecision rather than pure horizontal mislocalization – is noted as a plausible, separate, and possibly co-occurring issue. We report this anomaly in full rather than omitting it or asserting a single tidy explanation; the single most diagnostic next step we did not take would be direct inspection of a raw camera frame with the sample at the affected position, which would distinguish "not visible in the image" from "visible but not acted on."

6.4 A note on compute resources during rollout

Because occasional stalls had been observed during earlier rollout sessions on the Raspberry Pi 5, CPU load and temperature were monitored directly (via a locally connected monitor and `htop`) during a rollout under the corrected setup. Load rose to approximately 36% at the start of inference and peaked at approximately 50% during the grasp-and-transport phase; temperature remained steady at 51–54°C throughout. We found no evidence of CPU or thermal bottlenecking under the corrected configuration, and flag any future recurrence of a stall as more likely software/timing-related than a hardware resource limit.

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

7 Discussion

Training-loss convergence is not evidence of behavioral soundness. The zero-success camera-artifact policy in Section 3 converged exactly as smoothly as every working policy in this project; the defect was entirely in the recorded visual content and was detectable only by watching the footage. We recommend direct, if brief, visual review of a sample of training episodes as a standard step before attributing a rollout failure to model capacity, training duration, or hyperparameters – a loss curve alone cannot distinguish a working dataset from one containing a hidden distribution-shifting artifact.

"Background generalization failure" is not a single thing. Section 4's three mechanisms – pattern distraction, glare, and exposure instability – would each be invisible or misdiagnosed if treated as one undifferentiated phenomenon. Pattern distraction is plausibly addressable with more diverse training backgrounds; glare is addressable with lighting control at both recording and deployment time; exposure instability, in our case, was not addressable in software at all, and is better treated as a hardware/platform selection criterion than a modeling problem. We recommend that any report of "poor background generalization" be broken down by direct behavioral observation (what specifically did the arm do: fail to find the object, locate it but miss, or behave erratically) before a remedy is proposed, rather than aggregating every novel-background failure into one percentage.

Lighting and background interact, but are separable. The same glossy background that performs identically to every other surface below approximately 400 lux degrades sharply above it (Section 5), while the pattern-distractor mechanism in Section 4 was explicitly shown *not* to respond to additional lighting. Treating "background" and "lighting" as a single combined variable would have obscured this distinction; measuring lux directly, rather than relying on qualitative lighting descriptions, was what made the separation possible.

Small trial counts are not a minor caveat. The flip in outcome at several novel-position coordinates between a 1-trial pass and a 7-trial pass (Section 6) is a direct, measured demonstration that a handful of trials per condition can mislead even when the aggregate rate across many conditions happens to look consistent. We recommend treating any single-digit-trial-count success rate, including several reported elsewhere in this paper where a larger pass was not available, as a provisional estimate rather than a settled result.

Open anomalies should be reported as open. The position-4 red-sample blind spot (Section 6) remains unexplained after three targeted diagnostic attempts. We considered omitting it in favor of reporting only the clean trained/novel-position baseline, but judged that doing so would overstate how well this system is understood; a reliable system can still contain narrow, unexplained failure modes, and we think this is worth stating plainly rather than quietly excluding the inconvenient data point.

What we would do differently, and a practitioner checklist

Several of the problems above cost weeks and were avoidable. In hindsight we would have done the following from the start, and we suggest the same to others building on similar low-cost hardware.

1. **Watch the footage before training.** The zero-success policy of Section 3 cost a full recording session, a training run, and diagnosis on two machines before anyone reviewed the raw episodes. A ten-minute visual review of a sample of episodes, looking for anything present at recording time but absent at inference (an operator's hand, a collision, a distorted lens), would have caught it first.

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

2. **Choose the camera position for the rig actually in use.** The high mount was adopted for hypothetical compatibility with hardware that was not yet built and would run a different policy anyway. Optimize for the setup you are running today.

3. **Verify the capture chain on the recording machine before recording.** Shared USB bandwidth slowed one recording campaign, and auto-exposure instability very likely degraded the black-background training data (Section 4.3.3); both are checkable in minutes before a long session.

4. **Control and log lighting from day one.** We measured lux only after weeks of qualitative remarks about "bad lighting". It is also possible, though we could not confirm it, that some instability we first attributed to an earlier hardware platform was a lighting mismatch, since that hardware was retired before the question was asked. Match deployment lighting to recording lighting before attributing a regression to the model.

5. **Standardize the test object early.** Object shape and color were a confound throughout Section 5; fixing one object at the start would have removed it.

6. **Characterize failures by direct observation** (not found vs. found-but-missed vs. erratic) before selecting a fix, and log the failure code for every trial, as in the result codes used in Section 4.

7. **Budget enough trials per condition.** We suggest 5–10 at minimum, consistent with the swings observed here between 1- and 7-trial passes.

8. **Capture a raw camera frame at every unexplained failure.** The position-4 blind spot (Section 6) remains open largely because we never saved the frame that would show whether the sample was visible to the policy.

Limitations

All findings come from a single operator, a single rig, and a single pick-and-place task; we make no claim that the specific mechanisms identified here (e.g., the approximately 400 lux glare threshold, or the exact camera/OS combination responsible for the exposure-instability finding) generalize numerically to other hardware. Some cells of Table 3 (gift-wrap and white at P1–P3 and P5) are reported as percentages without a fraction in the experiment's summary, and the black background has no clean figure. The position-4 spatial anomaly in Section 6 is reported as genuinely unresolved; none of the candidate explanations tested should be read as confirmed. Finally, the auto-exposure instability finding (Section 4.3.3) is specific to the camera/macOS-AVFoundation combination used in this work and should not be assumed to transfer to other camera/OS combinations without separate verification.

Code and Data Availability

The dataset merge pattern, streaming-encoding recording flags, and training/rollout command templates used throughout this work are provided in the Appendix.

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

A Dataset Merge Pattern

All multi-batch dataset merges in this work (Sections 3 and 4) used the same underlying pattern, built on `LeRobotDataset.create()` / `add_frame()` / `save_episode()`. The most common error encountered across several independent merge scripts was supplying the policy-feature-schema format (`{"type": "STATE"/"VISUAL", ...}`) where the dataset-metadata format (`{"dtype": "float32"/"video", ...}`) is actually required, which raises `KeyError: 'dtype'` deep inside a `video_keys` property lookup. The corrected pattern:

```
"""
Merge multiple recording batches into one LeRobotDataset, preserving the
dataset-metadata feature schema (NOT the policy-feature schema).
"""

import sys
sys.path.insert(0, "src")
import numpy as np
from lerobot.datasets.lerobot_dataset import LeRobotDataset

BASE_REPO_IDS = [
    "<batch_1_repo_id>",
    "<batch_2_repo_id>",
    "<batch_3_repo_id>",
]
MERGED_REPO_ID = "<merged_repo_id>"
TASK = "<task description, exactly matching the recorded task string>"


def get_episode_ranges(dataset):
    episode_index_col = np.array(dataset.hf_dataset["episode_index"])
    ranges = {}
    for ep in np.unique(episode_index_col):
        idxs = np.where(episode_index_col == ep)[0]
        ranges[int(ep)] = (int(idxs.min()), int(idxs.max()) + 1)
    return ranges


def to_writer_image(img):
    arr = img.permute(1, 2, 0).numpy()
    if arr.dtype != np.uint8:
        arr = (arr * 255).clip(0, 255).astype(np.uint8)
    return arr


def copy_dataset_episodes(source, new_dataset, task_label):
    ranges = get_episode_ranges(source)
    for ep_idx in range(source.meta.total_episodes):
        ep_start, ep_end = ranges[ep_idx]
        for frame_idx in range(ep_start, ep_end):
            sample = source[frame_idx]
            frame = {
                "observation.state": sample["observation.state"],
                "action": sample["action"],
                "observation.images.main": to_writer_image(sample["observation.
                    images.main"]),
                "observation.images.gripper": to_writer_image(sample["observation.
                    images.gripper"]),
                "task": task_label,
```

```
            }
            new_dataset.add_frame(frame)
        new_dataset.save_episode()


def main():
    sources = [LeRobotDataset(repo_id=r) for r in BASE_REPO_IDS]
    new_dataset = LeRobotDataset.create(
        repo_id=MERGED_REPO_ID,
        fps=sources[0].meta.fps,
        features=sources[0].meta.features,  # dataset-metadata format, not policy-feature format
        robot_type=sources[0].meta.robot_type,
        use_videos=True,
    )
    for source in sources:
        copy_dataset_episodes(source, new_dataset, TASK)


if __name__ == "__main__":
    main()
```

B Recording, Training, and Rollout Commands

```
# Recording (streaming-encoding flags fix a BrokenProcessPool crash
# during episode encoding, and are used as standard practice throughout)
lerobot-record \
    --robot.type=so101_follower \
    --robot.port=<follower_port> \
    --robot.id=my_follower_arm \
    --robot.cameras='{"main": {...}, "gripper": {...}}' \
    --teleop.type=so101_leader \
    --teleop.port=<leader_port> \
    --teleop.id=my_leader_arm \
    --dataset.repo_id=<dataset_repo_id> \
    --dataset.single_task="<task description>" \
    --dataset.num_episodes=<n> \
    --dataset.episode_time_s=20 \
    --dataset.reset_time_s=10 \
    --dataset.streaming_encoding=true \
    --dataset.encoder_threads=2 \
    --dataset.num_image_writer_processes=0 \
    --dataset.num_image_writer_threads_per_camera=2 \
    --dataset.push_to_hub=false

# Training
lerobot-train \
    --dataset.repo_id=<merged_dataset_repo_id> \
    --policy.type=act \
    --policy.device=<mps or cuda> \
    --steps=<steps> \
    --batch_size=8 \
    --policy.push_to_hub=false

# Rollout / evaluation (input/output features must be passed explicitly;
# lerobot-rollout does not auto-load them from the checkpoint's config.json)
```

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.

```
lerobot-rollout \
    --robot.type=so101_follower \
    --robot.port=<follower_port> \
    --robot.cameras='{"main": {...}, "gripper": {...}}' \
    --policy.type=act \
    --policy.pretrained_path=<checkpoint_path> \
    --policy.input_features='{"observation.state": {"type": "STATE", "shape": [6]},
        "observation.images.main": {"type": "VISUAL", "shape": [3, 480, 640]},
        "observation.images.gripper": {"type": "VISUAL", "shape": [3, 240, 320]}}' \
    --policy.output_features='{"action": {"type": "ACTION", "shape": [6]}}' \
    --policy.device=<mps or cuda> \
    --strategy.type=base \
    --fps=30 \
    --duration=20 \
    --task='"<task description>"'
```

References

[1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *Robotics: Science and Systems (RSS)*, 2023.

[2] Hugging Face. *LeRobot: State-of-the-art AI for real-world robotics.* https://github.com/huggingface/lerobot

[3] I. Radulov. *When Does a Trigger Reach a Robot Policy? Architecture, Fine-Tuning, and a Correctly Targeted Visual Backdoor.* Zenodo, 2026 (all versions). https://doi.org/10.5281/zenodo.23143960

[4] E. B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

zenodo:23212210 [cs.RO] 7 Oct 2026 DOI: 10.5281/zenodo.23212209.