How Visible Are Silent Manipulation Failures? An Observability Study of False-Success Detection in Simulated Robot Episodes

Aarav Bedi
University of California, Berkeley
aaravbedi@berkeley.edu

***Abstract*—Imitation-learning policies for robot manipulation inherit the quality of the success labels attached to their training episodes. In the autonomous setting these labels are produced by the robot's own success check, and a particularly damaging error is the false success: an episode the robot logs as a success when the task outcome was in fact wrong. We ask a narrow but practical question: once an episode is already flagged as a success by such a check, how much of the information needed to overturn that label lives in proprioception, and how much requires vision? We build a simulated testbed on two bimanual ALOHA tasks, induce failures through environment perturbations rather than label edits, label every episode by privileged simulator state the detector never sees, and keep only flagged-success episodes. We then compare proprioception-only detectors against a vision detector. In cube transfer, false successes are almost fully recoverable from joint data, but the separating signal rests on velocity differences far below any realistic sensor-noise floor, so we report it as a simulation artifact rather than a deployable result. In peg insertion, proprioception recovers only 65% of false successes while vision reaches 94%. All numbers are single-seed and should be read as optimistic upper bounds. The takeaway for production data-curation pipelines is that catching silent manipulation failures generally requires exteroception, and that clean-simulation proprioception results on this problem should be treated with care.**

***Index Terms*—robot learning, manipulation, failure detection, success detection, data quality, observability, imitation learning.**

I. Introduction

Manipulation policies are increasingly trained by imitation from large demonstration corpora and every corpus rests on the assumption that each episode carries a correct success or failure label. Labels are produced in several ways: by human demonstrators or teleoperators, by post-processing pipelines, or, in autonomous data collection, by the robot's own success detector or a scripted end-of-trajectory check. When an autonomous check fires on an episode that did not accomplish the task, the resulting false success enters the training set as a positive example, and the policy has no way to know the behavior it is imitating ended badly. We do not address labels supplied by a human or an external pipeline, where the failure mode and the appropriate audit are different.

False successes are harder to find than ordinary execution failures. A stalled joint or a violent jerk leaves an obvious trace in the sensed trajectory; a cube placed two centimeters off target, or a peg that looks seated but is not, may leave almost none. For a team trying to clean a training set before deployment, the practical question is not whether a failure occurred, but which sensing channel carries enough signal to recover it after the robot has already declared victory.

We study that question in simulation, deliberately avoiding the shortcut of injecting failures as label edits: a detector trained on such data learns to recover the injection rule rather than a real failure signature. We instead run a scripted controller under randomized environment perturbations, let genuine misses emerge from the physics, and read the true outcome from privileged simulator state the detector never receives. Real-world datasets do not contain such privileged state, which is why this outcome is hard to audit at scale in practice and why we isolate the observability question in simulation first; transferring the analysis to hardware-logged data is the open problem this work is meant to motivate. We keep only the episodes the robot flagged as successful, so the task mirrors what a production curation step faces: separate real successes from false ones using only observable signals.

Our contribution is an observability analysis, not a new detection method. We report, for two manipulation tasks, how far a proprioception-only detector and a vision detector get on the false-success problem, and we are explicit about where each signal comes from and how fragile it is. We do not claim the detectors are deployable; they are instruments for measuring how much of the false success signal each sensing channel carries.

II. Related Work

Manipulation datasets: Large demonstration corpora have driven recent progress in robot learning. Open X-Embodiment aggregates data across many robots and labs; BridgeData V2 and DROID provide large single-embodiment collections; and LeRobot standardizes the data format and supplies widely used reference datasets. These efforts target scale and coverage, not a labeled measure of annotation quality.

Failure and anomaly detection: Detecting execution failures from proprioception and force is well studied, with residual-monitoring and learned sequence-model approaches. These methods work precisely because the failures they target leave a signature in the sensed trajectory. The case where the trajectory looks nominal but the outcome is wrong has received much less attention.

TABLE I
TESTBED COMPOSITION (FLAGGED-SUCCESS EPISODES ONLY, SEED 42).

	Transfer	Insertion
Flagged-success episodes	500	500
True successes	262	341
False successes	238	159
False-success rate	47.6%	31.8%
Object planar jitter	1.5 cm	0.5 cm
Object yaw jitter	15°	5°
Object friction range	0.15–0.9	0.15–0.9
Added observation noise	none	none

Success detection: Vision-based success classifiers and learned reward models judge task outcome from images and are the natural tool for catching false successes. What has been missing is a controlled measurement of how much of the false-success signal lives in proprioception versus vision once the robot has already reported success. The simulated ALOHA tasks and scripted policies we build on come from the ACT line of work.

III. TESTBED DESIGN

Both tasks derive from the bimanual ALOHA simulator: a cube transfer task and a peg insertion task. We use the scripted end-effector controllers that ship with the simulator so that nominal behavior is competent and reproducible, and we draw the true outcome from the simulator's privileged object state.

Inducing failures: For each episode we apply randomized environment perturbations (Table I): a small planar and yaw perturbation of the manipulated object, and randomized object friction. The scripted controller then runs to completion. Some perturbed episodes still succeed and some do not; the failures are a consequence of the physics, not of an annotation.

Two labels, kept apart: The robot flag is a cheap success heuristic computed from proprioception only, standing in for the telemetry-based check a real robot would use. The ground-truth outcome is read from privileged simulator state (object pose relative to target), which the detector never observes. A false success is an episode where the flag fired but the ground-truth outcome is a failure. We retain only flagged-success episodes, so the dataset is precisely the pool a curation step would inspect.

Failure modes: Modes are assigned after the fact from ground truth, never injected. In transfer the failures are dominated by a misplaced cube; in insertion, by a peg that ends up not seated near the socket.

IV. EVALUATION PROTOCOL

Given an episode flagged as a success, predict whether it truly succeeded. The headline metric is false-success recall, the fraction of truly failed episodes the detector flags, because that is the quantity a curation step cares about and because overall accuracy is misleading when one class dominates. False-success recall is only one axis, however: a curation step also pays for the good data it discards, so the operating point matters. We therefore report macro F1 alongside recall as a coarse proxy for this tradeoff, and we note that a deployed filter would be tuned to a target precision rather than to recall alone. A full precision-at-fixed-recall characterization across operating points is left to follow-on work.

TABLE II
DETECTOR METRICS ON THE HELD-OUT TEST SET (150 EPISODES).

Task	Detector	Acc.	FS rec.	TS rec.	Macro F1
Transfer	Proprioception	0.973	0.972	0.975	0.973
Transfer	Vision	0.987	0.986	0.987	0.987
Insertion	Proprioception	0.760	0.646	0.814	0.727
Insertion	Vision	0.973	0.938	0.990	0.969

Baseline A (proprioception) Uses 39 velocity summary statistics over the full episode: root-mean-square, peak, and standard deviation of the per-step joint-velocity magnitude, globally and for each of the twelve arm joints. No privileged state or perturbation parameter is used as a feature. **Baseline C (vision)** uses three features read from the rendered final-frame image: the normalized image centroid and the pixel area of the manipulated object, obtained by color-thresholding. Object pose is never read from simulator state. Both detectors are gradient-boosted tree classifiers with class balancing, trained on a fixed split and evaluated on 150 held-out episodes (seed 42).

V. RESULTS

Recoverability differs sharply by task: Table II gives the headline single-seed numbers. In transfer, the proprioception detector flags 97% of false successes and vision adds little, because the placement failures change the load the arm carries and are written into the joint trajectory. We write "flags 97%" rather than "recovers 97%" deliberately: as the next paragraph shows, this figure reflects a separability specific to the noiseless simulator and should not be read as a claim that proprioception recovers 97% of false successes in the field. In peg insertion, proprioception flags only 65% of false successes while vision reaches 94%. Insertion is where the modality gap is real and the message is actionable: exteroception is necessary to catch silent peg-insertion failures.

The proprioceptive signal is tiny in absolute terms: Table III pairs each trajectory window's Cohen's d with the actual difference between the class means. In transfer's first window the mean feature differs by only about 0.001 in normalized units, yet $d \approx 0.99$. The large effect size comes almost entirely from near-zero within-class variance, which the deterministic simulator produces, and a real encoder would not. The implication is direct and testable: any additive sensor noise with standard deviation comparable to or larger than the roughly 0.001 mean gap would erase the separability, collapsing the transfer proprioception recall toward chance while leaving the vision detector, which reads a final-frame object centroid, largely unaffected. We state this as a prediction that follows from Table III rather than a measured curve; a controlled ablation that sweeps injected encoder noise and domain randomization to locate the exact noise level at which

TABLE III
PER-WINDOW SEPARABILITY AGAINST ABSOLUTE MEAN DIFFERENCE (TRANSFER).

Window	mean (true)	mean (false)	Cohen's d
t000–100	0.8598	0.8608	0.987
t100–200	0.2002	0.1929	0.993

proprioceptive separability disappears is the most important single follow-on experiment, and the one we prioritize next. Read in this light, the 0.97 and 0.65 numbers are best treated as optimistic upper bounds, not expected field performance.

VI. IMPLICATIONS FOR PRODUCTION PIPELINES

The insertion result carries a direct implication for data curation at scale: a pipeline relying only on joint-state telemetry will miss roughly 35% of false-success peg-insertion episodes in this testbed, while adding a lightweight final-frame vision check closes that gap to 6%. This favors even simple exteroceptive verification over proprioception-only quality checks. Because the study is single-seed and noiseless, the specific percentages should be read as indicative rather than as calibrated field estimates; the qualitative ordering, that vision materially outperforms proprioception on the misalignment-dominated task, is the robust claim.

The transfer result is a caution in the other direction: high proprioceptive recall does not imply the failures are silent. When perturbations change the arm's own dynamics, a proprioception detector performs well, but that reflects physical-failure detection, not the harder misalignment problem. Practitioners building a curation benchmark should verify that their testbed failures are genuinely invisible to joint data before claiming misalignment detection, or risk over-reporting what proprioception can recover in the field. Where labels are produced by human demonstrators or an external pipeline rather than an autonomous success check, the relevant audit differs and the observations here do not directly apply.

VII. LIMITATIONS

Episodes are generated in a MuJoCo simulator and the proprioceptive separability we report rests on velocity differences that real encoder noise would attenuate; the proprioception numbers are therefore best read as upper bounds. Validating the failure signatures against hardware-logged episodes with calibrated sensor noise is a natural next step and would directly test the noise-fragility prediction made in Section V.

Results use a single random seed (42), so confidence intervals across seeds are not yet available. This is the most significant limitation of the present version: the reported recall figures are point estimates, and a five-or-more-seed rerun with error bars is required before the specific percentages can be relied upon. We report the single-seed numbers as an observability result whose qualitative conclusion, that vision is necessary on the insertion task and that transfer proprioception separability is a simulation artifact, we expect to be stable, while treating the exact values as provisional.

The scripted controller steers toward the true object pose, so part of the insertion proprioception signal is tied to this oracle policy; a learned or open-loop policy would distribute it differently. The vision baseline reads a color-thresholder object centroid and pixel area from the rendered final frame, which is specific to this simulator; a general-purpose learned success classifier would be the appropriate exteroceptive baseline for a deployment study, and we expect it to be at least as capable as the simple detector used here. Finally, the taxonomy is compact, two tasks with one to two dominant failure modes each, leaving drops, partial grasps, external disturbances, and multi-step failures for future work. A deployable, reproducible curation system that applies these findings to filter a real dataset is beyond the scope of this observability analysis and is the intended longer-term direction.

VIII. CONCLUSION

We asked which sensing channel carries the information needed to overturn a false success once a robot has already flagged an episode as successful. The answer is task-dependent and the honest reading of our numbers is cautionary. On the misalignment-dominated insertion task, proprioception alone misses roughly a third of false successes while a lightweight vision check recovers almost all of them, so exteroception is necessary. On the transfer task, proprioception appears nearly perfect, but that separability rests on velocity differences below any realistic sensor-noise floor and is best read as a simulation artifact. For production data-curation pipelines, the practical message is that catching silent manipulation failures generally requires exteroception, and that strong clean-simulation proprioception results on this problem should be treated with care until they are validated against hardware-logged data with realistic noise.

AI USAGE STATEMENT

AI tools were used to assist with code debugging. The author takes full responsibility for all the experimental design, results, and conclusions.

REFERENCES

[1] Open X-Embodiment Collaboration, "Open X-Embodiment: Robotic learning datasets and RT-X models," arXiv:2310.08864, 2023.

[2] H. Walke et al., "BridgeData V2: A dataset for robot learning at scale," in *Proc. CoRL*, 2023.

[3] A. Khazatsky et al., "DROID: A large-scale in-the-wild robot manipulation dataset," in *Proc. RSS*, 2024.

[4] R. Cadene et al., "LeRobot: An open-source library for end-to-end robot learning," in *Proc. ICLR*, 2026.

[5] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE RA-L*, vol. 3, no. 3, pp. 1544–1551, 2018.

[6] A. Inceoglu et al., "FINO-Net: A deep multimodal sensor fusion framework for manipulation failure detection," in *Proc. IEEE/RSJ IROS*, 2021.

[7] Y. Du et al., "Vision-language models as success detectors," arXiv:2303.07280, 2023.

[8] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Proc. RSS*, 2023.