Auditing Demonstration-Curation Metrics: Action-Only Scorers Fail on the Structural Defects That Degrade Imitation Policies

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

***Abstract*—Imitation-learning policies inherit the quality of the demonstrations they are trained on, and a growing set of curation metrics promise to score and filter low-quality demonstrations automatically. Because each metric is validated on different data under a different protocol, it is unclear which of them identify the demonstrations that harm a policy. We build a controlled testbed in which demonstration defects are injected with known type, and audit seven curation metrics on two axes: detection, how well a metric separates defective from clean demonstrations, and downstream value, whether training a behavior-cloning policy on the metric's curated subset improves task success. We study two defect regimes. Subtle perturbations (correlated action noise, tremor, truncation) are statistical wobble layered on otherwise correct behavior; they are detectable by multivariate outlier scoring and, once removed, recover the full downstream gap. Structural defects, where the demonstration executes a wrong action at a decisive moment, are invisible to every action-only metric in our testbed, and two are inverted: they score defective demonstrations as higher quality and, used for curation, leave the policy at or below the uncurated baseline. Only metrics that read the state trajectory detect structural defects, and even the best recovers less than half of the downstream gap. High detection accuracy does not guarantee downstream improvement. We release the testbed and all curation implementations.**

***Index Terms*—imitation learning, behavior cloning, demonstration curation, data quality, robot learning.**

I. Introduction

A policy trained by imitation can only be as good as the demonstrations it copies. Large demonstration sets are rarely uniform in quality: operators differ in skill, scripted controllers occasionally fail, and autonomous rollouts get labeled successful by checks that are themselves imperfect. Whatever the source, some of the data is bad, and a behavior-cloning policy has no way to tell the good demonstrations from the bad. It imitates all of them.

Curation is the obvious fix: score each demonstration with a quality metric, drop the low-scoring ones, and train on what remains. The literature offers many such metrics, from trajectory smoothness to outlier detection in a feature space, but no basis for choosing between them, since each is reported on its own dataset under its own protocol. There is also a subtler problem. A curation metric is rewarded for identifying demonstrations that look unusual, but what a practitioner cares about is which demonstrations hurt the policy. These are different questions, and a metric good at the first need not be good at the second. We make this concrete by scoring every metric on two explicit axes, which we use throughout the paper: detection, whether the metric's quality ranking separates injected-defective demonstrations from clean ones, and downstream value, whether a policy trained on the metric's curated subset succeeds more often. The downstream-value axis is the one a practitioner cares about, and by construction it does not depend on the metric's own score.

Our findings are organized by a distinction between two kinds of defect. Some defects are statistical: they perturb how a motion is executed without changing what the demonstration accomplishes, and they average out as a policy sees more data. Others are structural: they change what the demonstration does at a decisive moment, planting a wrong action in a specific region of the state space that no amount of averaging removes. A curation metric that reasons only about action statistics is blind to the structural kind by construction in this testbed, because the information that distinguishes such a defect lives in the state trajectory, in where the arm went and what it was trying to do, not in the statistics of how it moved.

We separate detection from downstream value in a controlled testbed. We simulate a manipulation task, inject defects into demonstrations with a known type and location, and withhold those labels from every metric. Each of seven metrics is scored on the detection axis (how cleanly its ranking separates defective from clean demonstrations) and on the downstream axis (whether a behavior-cloning policy trained on the subset it keeps succeeds more often in the simulator). The downstream number is the one a practitioner cares about, and by construction it does not depend on any curation score.

Our contribution is the audit and the testbed, not a new curation method, and not a claim about imitation-learning frameworks in general. We ask a single focused question: with the trained policy as the judge, which of these existing metrics help, which do nothing, and which make matters worse.

II. Related Work

Imitation learning from demonstrations: Behavior cloning trains a policy by supervised regression from observations to demonstrated actions [7], and remains the backbone of modern manipulation learning despite its known sensitivity to distribution shift [8]. Recent bimanual systems such as ACT imitate large demonstration sets on low-cost hardware [10], and large cross-embodiment corpora and standard data formats

have scaled this further [2], [3], [6], [9]. All of these assume the demonstrations are worth imitating.

Demonstration quality and curation: The observation that demonstration quality, not just quantity, drives imitation performance is central to studies of offline human data [5], which document large differences between operators. This motivates automatic curation: scoring demonstrations and filtering low-quality ones. Proposed signals range from trajectory smoothness, for which spectral arc length is a standard movement-quality measure [1], to generic outlier detection such as isolation forests [4], and recent work scores demonstrations with mutual-information estimators over state-action data [11]. What has been missing is a controlled, head-to-head comparison that asks whether these signals identify the demonstrations that degrade a policy, rather than the ones that merely look unusual. We provide that comparison; our contribution beyond prior curation work is not a new score but a protocol that separates detection from downstream value and shows the two can disagree sharply.

III. Testbed and Protocol

Environment: We use a lightweight pick-and-place simulator written in NumPy, modeled on the single-arm ALOHA setup. The action is seven-dimensional (end-effector translation, rotation, and gripper). The behavior-cloning observation is eleven-dimensional: end-effector position and orientation, gripper state, a noisy estimate of the object position (Gaussian noise, standard deviation 0.03 m per step, standing in for imperfect perception), and a normalized time index. The four extra observation dimensions beyond the seven-dimensional action are perceptual context and do not correspond one-to-one with commanded action dimensions. The task is to grasp the object and carry it to a fixed goal region; success is the object resting in the goal at episode end. A phase-based scripted controller with access to privileged state solves the task nearly every time and supplies the clean demonstrations; the learned policy never sees the privileged state.

Two labels, kept apart: A clean demonstration is a successful scripted episode. A defective demonstration is produced by applying one of the injectors described below to a clean episode; its defect type is recorded but never made available to any curation metric. Metrics see only a view exposing the states and actions of a demonstration, enforced at the type level so a metric cannot read the label.

Defect regimes: We study two regimes, each at a 40 percent contamination rate. Subtle perturbations: (i) action noise, temporally correlated AR(1) noise added to the actions, resembling shaky teleoperation rather than random garbage; (ii) tremor, a high-frequency sinusoid added to the actions; (iii) truncation, cutting the episode to roughly half its length; (iv) detour, splicing a reversed copy of a mid-episode segment back in so the trajectory loops before completing. Structural defect: early release, in which the gripper is commanded open during the carry phase, so the demonstration drops the object partway to the goal. Early release is not noise; it is a systematic wrong action in a specific region of the state space, and it does not average out across demonstrations.

TABLE I
Detection AUROC, subtle-perturbation regime (3 seeds).

Metric	AUROC
isolation forest	0.968 ± 0.004
kNN	0.871 ± 0.003
length	0.625 ± 0.000
trajectory alignment	0.528 ± 0.044
smoothness	0.478 ± 0.005
ensemble	0.002 ± 0.001
entropy	0.000 ± 0.000

TABLE II
Detection AUROC, structural-defect regime (3 seeds).

Metric	AUROC
kNN	0.863 ± 0.025
trajectory alignment	0.762 ± 0.025
isolation forest	0.543 ± 0.031
length	0.500 ± 0.000
entropy	0.469 ± 0.028
ensemble	0.291 ± 0.028
smoothness	0.244 ± 0.020

Curation metrics: We audit seven metrics, each producing a per-demonstration quality score from states and actions only. Five are action-only: smoothness (spectral arc length of the speed profile), entropy (standard deviation of the action sequence), length (trajectory length), isolation forest (outlier scoring on action-derived summary features), and ensemble (a fixed average of the other action-only scores into a single composite score). Two are state-trajectory-aware: kNN, scoring a demonstration by its distance to nearest neighbors in a trajectory-level feature space that includes state-trajectory summaries, so demonstrations leaving the local data manifold score low; and trajectory alignment, scoring how well a demonstration's state trajectory agrees with the dataset's aggregate behavior. An oracle that filters using the hidden labels upper-bounds what perfect curation could achieve. We fix the metric ordering (isolation forest, kNN, trajectory alignment, length, smoothness, ensemble, entropy) and use it consistently in every figure and table.

Evaluation: On the detection axis we report the area under the ROC curve (AUROC) of each metric's quality score against the hidden defective/clean label. On the downstream axis we curate the contaminated set by keeping the top-scoring fraction under each metric, the curation cutoff, then train a three-layer behavior-cloning MLP on the kept subset and measure task success over 50 fresh simulator rollouts. The cutoff is fixed in advance so that every metric keeps the same number of demonstrations: we retain $N = 150$ demonstrations per condition, matched to the clean-data count, so that differences reflect quality rather than volume. This removes the confound between data quantity and data quality. We report mean and standard deviation over three seeds (42, 0, 7).

TABLE III
DOWNSTREAM SUCCESS, SUBTLE-PERTURBATION REGIME (N=150, 50 ROLLOUTS, 3 SEEDS).

Training set	Success (%)
oracle clean only	72.0 ± 4.9
isolation forest	71.3 ± 10.9
smoothness	70.0 ± 9.1
length	56.7 ± 15.5
ensemble	56.0 ± 16.1
full contaminated	55.3 ± 0.9
entropy	41.3 ± 20.2

TABLE IV
DOWNSTREAM SUCCESS, STRUCTURAL-DEFECT REGIME (N=150, 50 ROLLOUTS, 3 SEEDS).

Training set	Success (%)
oracle clean only	62.7 ± 4.7
kNN	48.0 ± 4.3
length	38.7 ± 5.0
trajectory alignment	37.3 ± 5.0
full contaminated	36.0 ± 4.3
isolation forest	35.3 ± 2.5
ensemble	32.0 ± 5.7
entropy	30.7 ± 3.4
smoothness	27.3 ± 6.8

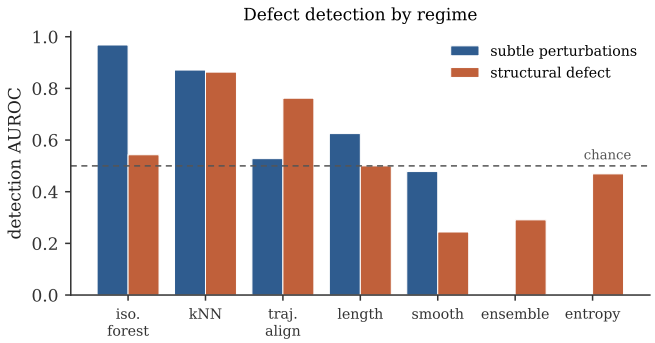

Fig. 1. Defect-detection AUROC by metric and regime. No metric is reliable in both regimes: the isolation forest leads on subtle perturbations but falls to chance on the structural defect, while the state-trajectory metrics (kNN, trajectory alignment) are the only ones that detect the structural defect. Entropy and ensemble fall below chance, indicating inverted rankings. Dashed line marks chance (AUROC 0.5).

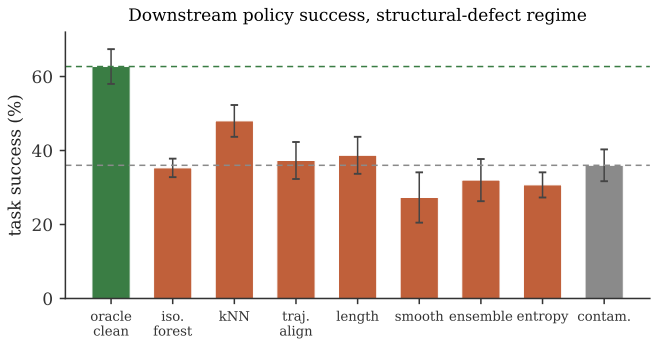

Fig. 2. Downstream policy success in the structural-defect regime (N=150, 50 rollouts, 3 seeds; error bars are one standard deviation). Dashed lines mark the oracle (green) and no-curation contaminated baseline (gray). Only kNN improves meaningfully on the baseline, recovering nearly half the oracle gap; action-only metrics match the baseline or, for smoothness, fall below it.

IV. RESULTS

Detection differs sharply between regimes: Fig. 1 and Tables I and II show that no metric is strong on both regimes. On subtle perturbations the multivariate isolation forest is strongest (AUROC 0.97) and the state-trajectory kNN is close behind; on the structural defect the isolation forest falls to chance (0.54) while kNN (0.86) and trajectory alignment (0.76), which read the state trajectory, are the only metrics that detect it. The smoothness and entropy metrics fall below chance on the structural defect, and entropy is inverted on the subtle regime: correlated noise and tremor both inflate action variance, so entropy scores the defective demonstrations as more exploratory, and therefore higher quality, than the clean ones. The ensemble inherits this inversion because the entropy component pulls its composite score the wrong way.

Regarding kNN specifically: it is the only metric that is respectable on both regimes (0.87 subtle, 0.86 structural), which is why it is the single best default if the defect type is unknown. We nonetheless avoid calling any metric good in both, because kNN's downstream recovery on the structural defect is only partial (below), and detection quality alone does not establish curation value.

Subtle defects are recoverable: In the subtle-perturbation regime (Table III) the contaminated set trains a policy at 55.3 percent against an oracle ceiling of 72.0 percent, and the isolation forest recovers most of that gap (71.3 percent). Downstream variance in this regime is wide across seeds: per-method standard deviations run as high as 20 points, so we read the subtle-regime downstream magnitudes as suggestive and lean on the detection axis, where the separation is sharp and stable, for firmer statements. The clearest of those is entropy's inversion (AUROC 0.000 across seeds): it ranks noisier demonstrations as higher quality, the opposite of what curation needs.

Structural defects resist curation: The structural regime is where curation matters most and helps least (Table IV, Fig. 2). The contaminated policy succeeds only 36.0 percent of the time against a 62.7 percent oracle, a 27-point gap. The one clear improvement is kNN at 48.0 ± 4.3 percent, recovering nearly half of the gap (about 45 percent), roughly two standard deviations clear of the baseline across seeds. No other metric improves on the contaminated set. The inverted action-only metrics (smoothness, ensemble, entropy) all sit below the baseline in point estimate, smoothness lowest at 27.3 percent; with only three seeds these below-baseline gaps fall within seed variance, so we report them as a consistent downward trend rather than a sharp effect. The robust statement is that no action-only metric helps here, while the one state-trajectory metric that helps does so only partially.

Detection does not guarantee recovery: Fig. 3 plots structural-defect detection against downstream success. Both

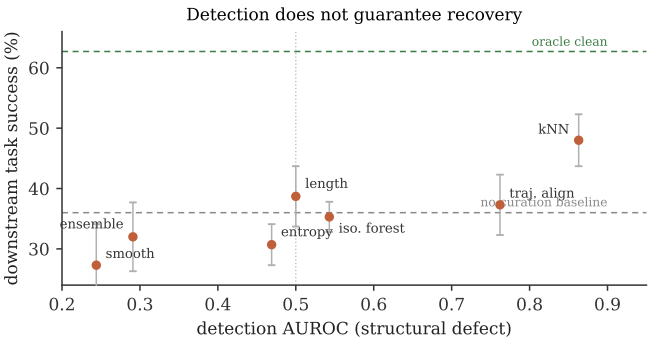

Fig. 3. Detection AUROC versus downstream success in the structural-defect regime. Comparable detection does not imply comparable recovery: kNN and trajectory alignment both detect the defect (AUROC 0.86 and 0.76) yet land about eleven points apart downstream (48.0 percent versus 37.3 percent). Reference lines mark the oracle and no-curation baseline. We show the structural regime only, because in the subtle regime every retained metric closes the downstream gap and the scatter is uninformative; subtle-regime detection is reported in Table I.

state-aware metrics detect the defect well, though not equally (AUROC 0.86 and 0.76), yet kNN recovers nearly half the downstream gap while trajectory alignment barely clears the no-curation baseline. Detection accuracy is necessary but not sufficient: which demonstrations a metric removes, and which clean ones it discards as collateral, matters as much as its ranking quality. A plausible reason is that trajectory alignment scores against the dataset's aggregate behavior, which the 60 percent clean majority pulls toward clean trajectories, diluting the signal, whereas kNN's local-neighborhood comparison is less diluted.

V. Discussion

Two things come out of this audit. The first is practical: match the curation metric to the kind of defect you expect. Multivariate outlier scoring handles subtle perturbations well, but only metrics that read the state trajectory have a chance on structural defects, and the action-only smoothness and entropy scores can do real harm, discarding good demonstrations and leaving a policy worse than if it had not been curated at all. The second is methodological: a metric's detection score says little about its curation value. Two state-trajectory metrics that both detect the structural defect (AUROC 0.86 and 0.76) produced policies about eleven points apart downstream (48.0 percent versus 37.3 percent), and isolation forest, which dominates the subtle regime, is no better than chance on the structural one. Reporting a curation method by its detection accuracy alone would therefore be an incomplete and potentially misleading characterization.

Why do the regimes behave so differently? Behavior cloning averages over its demonstrations, and the two defect families sit on opposite sides of that average. Subtle perturbations are roughly zero-mean wobble on otherwise correct behavior, so they wash out as demonstrations accumulate and even rough curation suffices. Early release is not zero-mean: it plants a specific wrong action in a region of the state space, and averaging never removes it. The defects worth catching are those whose signature lives in where the arm went, not in the statistics of how it moved, which is the information the action-only metrics discard. The action statistics capture the style of a motion but not its intent, and it is the violation of intent, the wrong action at the decisive moment, that a curation metric must see to protect the policy.

A note on using the policy as judge: Judging curation by the policy it produces is not the same as overfitting to the outcome. The downstream metric trains a fresh policy on the curated subset and evaluates it on held-out rollouts; it never selects demonstrations by their downstream effect. The point is diagnostic: existing quality scores are validated by detection alone, and we show detection can be satisfied while downstream value is not. A deployable curation rule would still use an observable score, and our results simply argue that such a score must read the state trajectory to have any chance on structural defects.

VI. Limitations

The scope here is narrow by design. The simulator is a lightweight NumPy proxy rather than a contact-rich physics engine, and the task is a single single-arm pick-and-place. The behavior-cloning observation includes a noisy object-position estimate that stands in for perception instead of raw pixels. Defects are injected with a known type rather than harvested from real operators, and we test a single policy class. These choices bought a clean, label-controlled comparison, and they cost realism.

Two caveats bear on how firmly the numbers should be read. First, we run only three seeds, and several downstream estimates carry large seed-to-seed variance, most of all in the subtle regime where per-method standard deviations reach twenty points; we therefore treat the downstream magnitudes as preliminary and the detection results, whose variance is small, as the more stable signal. A run with substantially more seeds and reported confidence intervals is the most important single follow-on, and is required before the specific downstream percentages can be relied upon. Second, because the defects are injected, the two families were chosen to straddle the action/state divide, one visible in action statistics and one only in the state trajectory, so the action-only metrics' blindness to the structural defect is partly by construction. The substantive results are the sizes of the downstream gaps and the finding that detecting a defect did not let any metric recover from it, not the bare existence of a state-only failure.

The remaining next steps are a sweep over contamination rates and additional tasks, a contact-rich benchmark such as robosuite or LIBERO, multiple policy architectures beyond behavior cloning, and real multi-operator quality labels of the kind found in offline human demonstration datasets. We expect the qualitative story to hold, with structural defects slipping past action-only curation and detection failing to imply recovery, but the specific numbers will move.

VII. Conclusion

We audited seven demonstration-curation metrics in a testbed that separates two things usually conflated: whether a metric flags a defect, and whether removing what it flags helps the policy. Subtle perturbations are both detectable and recoverable. Structural defects are neither: they slip past every action-only metric, and those metrics, used for curation, do not help and in point estimate trend below the no-curation baseline. The metric that wins one regime sits at chance in the other, and two metrics with matching detection scores can land far apart downstream. The practical takeaway is to judge a curation method by the policy it produces rather than the defects it flags, and to prefer state-trajectory-aware metrics when the defects that matter are structural.

Data and Code Availability

The simulator, defect injectors, all seven curation metrics, and the evaluation pipeline that produces every table and figure are released publicly at https://github.com/aaravbedi/structural-defect-curation.

References

[1] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet, "A robust and sensitive metric for quantifying movement smoothness," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 8, pp. 2126–2136, 2012.

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

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

[4] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. IEEE ICDM*, 2008.

[5] A. Mandlekar et al., "What matters in learning from offline human demonstrations for robot manipulation," in *Proc. CoRL*, 2021.

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

[7] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," in *Proc. NeurIPS*, 1989.

[8] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. AISTATS*, 2011.

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

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

[11] J. Hejna, S. Mirchandani, A. Balakrishna, A. Xie, A. Wahid, J. Tompson, P. Sanketi, D. Shah, C. Devin, and D. Sadigh, "Robot data curation with mutual information estimators," in *Proc. RSS*, 2025. arXiv:2502.08623.