

Calibra: Offline Dataset Observability and Coreset Selection for Robotic Imitation Learning

Omer Tahtacı
Independent Researcher
Istanbul, Türkiye
omer.tahtaci@sabanciuniv.edu

Abstract

Robotic imitation learning policies are highly sensitive to the quality and diversity of their training demonstrations, yet the field lacks principled offline tools for diagnosing data issues before training. We present **Calibra**, an open-source dataset observability framework that (1) diagnoses demonstration quality across four axes—temporal stability, control smoothness, coverage entropy, and task structure—with 95% bootstrap confidence intervals; (2) maintains a *falsifiable claim registry* that tracks every metric interpretation as a testable hypothesis with an evidence count and an explicit falsification condition; (3) selects high-quality, diverse coresets via a two-stage quality-filter and greedy maximum-coverage algorithm that scales to 500k+ episodes; and (4) closes the prediction loop by blending heuristic scoring with empirically observed training outcomes stored in a local database.

We profile 16 standard robotic datasets (ALOHA, DROID, BridgeData V2, PushT, SO-100) spanning five hardware platforms. Across three real datasets and three policy architectures (BC-MLP, ACT, Diffusion Policy) at fixed 30% retention with 5 seeds each, Calibra’s coreset selector reduces held-out action-prediction error by 13–25% over random selection on average, on par with dedicated diversity baselines (k-center, facility location). A companion 10-seed retention sweep on PushT shows this benefit is budget-dependent, not universal: Calibra’s combined quality-filter-plus-diversity selector is significantly *worse* than random at 10% retention ($p = 0.006$) and significantly better at 25% retention ($p < 0.001$). We report dataset curation as a retention-dependent strategy choice rather than a single universally-best method. In a controlled synthetic benchmark with injected corruption, pruning to a 30% coreset preserves 98% success rate (vs. 62% for random pruning) while

saving 70% of training compute — and, in proportion, the energy drawn by training — relative to full-dataset training. On the 7 of the 16 profiled datasets with an independently published ground-truth policy success rate, Calibra’s offline predictor shows a positive but not statistically significant rank association with actual policy success (Spearman $\rho = 0.60$, $p = 0.15$, $n = 7$; exact permutation $p = 0.17$), which we report as an exploratory signal motivating a larger, independently-evaluated benchmark rather than a validated predictor. Calibra is available at <https://github.com/omerTT/Calibra>.

1 Introduction

The dominant bottleneck in robotic imitation learning is no longer architecture design—it is data quality. Labs routinely collect tens of thousands of teleoperation demonstrations, then discover during or after training that policies fail because of subtle data pathologies: jerk spikes from abrupt operator corrections, temporal dropout from communication lag, or near-duplicate episodes that inflate compute without adding behavioral diversity.

These problems share three properties that make them difficult to address without dedicated tooling:

1. **Silent:** Standard dataset statistics (episode count, action dimension, frame rate) do not reveal them.
2. **Pre-training:** By the time a policy fails evaluation, tracing the failure to data is expensive and often inconclusive.
3. **Expensive:** Training a full policy to discover a data problem wastes hours to days of GPU time.

A further consequence of these pathologies is purely computational: robot demonstration datasets are frequently redundant, and every near-duplicate or low-quality episode still consumes a full gradient step during training without contributing new information. A

curation step that keeps only the highest-value fraction of a dataset for a given task therefore reduces the number of training examples processed per epoch and, in proportion, the wall-clock compute — and associated energy consumption — required to reach a target level of policy performance. We treat this as a practical consequence of the same coreset machinery used to address data quality, rather than as a separately optimized objective, and quantify it directly in Section 9.1.

We present **Calibra**, a dataset observability system that surfaces these issues *before training* using four principled analyzers, a falsifiable claim registry, and a coreset selector. Calibra is not an AI assistant and does not use language models: every output is a deterministic mathematical estimator backed by an explicit evidence trail.

Contributions.

1. **Four diagnostic analyzers** with per-episode metrics and 95% bootstrap confidence intervals computed correctly over episodes, not steps.
2. **Falsifiable claim registry**: every metric interpretation is a testable hypothesis with an evidence count, confidence level, and a stated falsification condition.
3. **Two-stage coreset selection**: quality filter followed by greedy maximum-coverage diversity selection, with an approximate variant that runs in $O(N \times B)$ for datasets of 500k+ episodes.
4. **Empirical prediction loop**: an outcome database that records observed training success rates alongside diagnostic fingerprints and blends heuristic predictions with empirical observations from similar past datasets.
5. **Real-time operator feedback** via `calibra watch`, which scores each episode at the moment it is saved and provides specific remediation advice to the teleoperator.

2 Related Work

Dataset quality in imitation learning. Dataset curation for behavior cloning has received growing attention. [Mandlekar et al. \[2021\]](#) showed that the quality of human demonstrations significantly affects policy performance and introduced a filtering criterion based on episode reward. [Khazatsky et al. \[2024\]](#) discuss data collection protocols for large-scale robot datasets but focus on collection infrastructure rather than offline quality diagnosis. Unlike reward-based filtering, Calibra operates without task reward signals

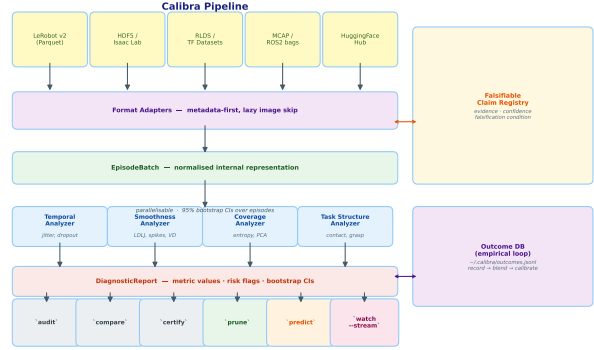


Figure 1: Calibra pipeline. A dataset is loaded via format adapters into a normalized `EpisodeBatch`. Four analyzers produce a `DiagnosticReport` with metric values, bootstrap CIs, and risk flags. The report drives six commands: `audit`, `compare`, `certify`, `prune`, `predict`, and `watch`.

and is therefore applicable to open-ended manipulation tasks where dense rewards are unavailable.

Coreset selection. Greedy k -center / farthest-point sampling [\[Gonzalez, 1985\]](#) is well-studied in computational geometry and has been applied to active learning [\[Sener and Savarese, 2018\]](#) and continual learning [\[Rebuffi et al., 2017\]](#). We apply it to the robotics demonstration setting, where episode-level action statistics form the feature space, and extend it with a quality pre-filter that discards pathological episodes before diversity selection.

Data-centric AI. The broader data-centric AI movement [\[Ng, 2021\]](#) advocates for systematic dataset improvement over model improvement. Calibra operationalizes this for robotics: it makes dataset issues inspectable, reproducible, and addressable before training.

Prediction of training outcomes. Predicting policy performance from data statistics alone is underexplored. [Walke et al. \[2023\]](#) study dataset diversity metrics and their correlation with transfer performance, but do not provide an integrated diagnostic tool. Our predictor (Section 7) extends this with a structured scoring rubric and an empirical calibration mechanism.

3 System Overview

Figure 1 shows the Calibra pipeline. A *format adapter* loads any of six dataset formats (LeRobot v2/Parquet, HDF5/Isaac Lab, RLDS, MCAP/ROS2, HuggingFace Hub) into a normalized **EpisodeBatch**. The batch is then analyzed by four analyzers that can run in parallel:

- **TemporalAnalyzer**: timestamp jitter CV, dropout rate.
- **ControlSmoothnessAnalyzer**: LDLJ score, jerk spike rate, velocity discontinuity rate.
- **CoverageEntropyAnalyzer**: action entropy (bits/dim), state entropy, PCA variance.
- **TaskStructureAnalyzer**: contact density, grasp event rate, short-episode fraction.

All metrics report 95% bootstrap confidence intervals computed over *episodes*, not steps, which avoids the artificially narrow intervals that arise from within-episode temporal correlation.

4 Diagnostic Metrics

4.1 Control Smoothness

LDLJ score. The Log-Dimensionless Jerk score [Balasubramanian et al., 2012] is a scale-invariant measure of trajectory smoothness:

$$\text{LDLJ} = \log \left(\frac{T^3}{A^2} \int_0^T \|\ddot{a}(t)\|^2 dt \right) \quad (1)$$

where T is episode duration, $A = \max_t \|\dot{a}(t)\|$, and $a(t)$ is the action trajectory. Lower (more negative) values indicate smoother demonstrations. Policy families differ in their sensitivity: ACT is most sensitive (threshold LDLJ > -10), while Diffusion Policy tolerates up to -20 .

Jerk spike rate. Fraction of timesteps where the action jerk exceeds $\mu + 5\sigma$ of the per-episode jerk distribution. Spikes inject outlier gradients that destabilize training.

Velocity discontinuity rate. Fraction of consecutive timestep pairs where the action velocity reversal exceeds a threshold. High rates teach the policy to produce jerky motions.

4.2 Temporal Stability

Timestamp jitter CV. Coefficient of variation of the inter-frame timestamp intervals. High jitter causes sequence models to overfit to irregular control cadences.

Dropout rate. Fraction of frames where the timestamp gap exceeds $2\times$ the nominal frame interval, indicating dropped frames.

4.3 Coverage Entropy

Action entropy in bits/dim is estimated via kernel density estimation on the marginal action distribution. Low entropy indicates that demonstrations cover a narrow behavioral mode, predicting poor policy generalization.

4.4 Confidence Intervals

All aggregate metrics report 95% bootstrap CIs with 1000 bootstrap samples drawn over *episodes* (not individual steps), following the recommendation of Agarwal et al. [2021] for correlated time-series data.

5 Falsifiable Claim Registry

A key differentiator of Calibra is that *every metric interpretation is a falsifiable claim*, not a heuristic rule of thumb. Each claim in `calibra/claims/` is a JSON object with:

- **assertion**: what the metric is expected to show for a dataset class.
- **evidence**: which reference datasets have been profiled and whether they support or contradict the assertion.
- **confidence**: derived from evidence count (NOT VALIDATED $\rightarrow$ LOW $\rightarrow$ MEDIUM $\rightarrow$ HIGH $\rightarrow$ STRONG).
- **falsification condition**: the exact observation that would invalidate the claim.

Ratio rule. CI enforces that the number of active claims must be $\leq$ the number of profiled reference datasets. This prevents the evidence base from diverging from reality as the codebase grows.

This structure was inspired by epistemological critiques of ML tooling [Sculley et al., 2015]: most ML systems embed implicit assumptions in thresholds without tracking whether those thresholds are empirically supported. The claim registry makes those assumptions explicit and testable.

6 Coreset Selection

6.1 Two-Stage Pipeline

Stage 1 — Quality filter. Episodes that fail any quality threshold (jerk spike rate, velocity discontinu-

ity rate, temporal dropout, LDLJ, minimum length) are discarded. This removes corrupted demonstrations that would inject harmful gradients.

Stage 2 — Greedy maximum-coverage. From the quality-passing pool, we select $K = \lfloor f \cdot N \rfloor$ episodes that maximize the minimum pairwise behavioral distance (greedy k -center / farthest-point sampling [Gonzalez, 1985]).

Episodes are represented by a feature vector of action-space statistics (mean, std, entropy per dimension) concatenated with temporal quality scores. The greedy algorithm is:

Algorithm 1 Greedy Farthest-Point Coreset Selection

```

1:  $S \leftarrow \{\text{random seed episode}\}$ 
2: for  $k = 2$  to  $K$  do
3:    $e^* \leftarrow \arg \max_{e \notin S} \min_{s \in S} d(e, s)$ 
4:    $S \leftarrow S \cup \{e^*\}$ 
5: end for
6: return  $S$ 

```

Influence strategy. With `--strategy influence`, selection is biased toward episodes with high action novelty, task contact representation, and Shannon entropy. This improves outcomes when fine-tuning generalist policies (GR00T, Octo).

6.2 Approximate Selector for Large Datasets

The exact selector runs in $O(N \times K)$ time and is suitable up to $\sim 50k$ episodes. For larger datasets (Open X-Embodiment, DROID), we implement a MiniBatch tournament algorithm:

1. Shuffle all N quality-passing episodes.
2. Split into batches of size B (default: 1000).
3. Run exact farthest-point within each batch $\rightarrow$ local candidates.
4. Tournament merge: greedily add the episode farthest from the current global selected set.

This runs in $O(N \times B)$ time and handles 500k+ episodes. At $B \geq 500$, the coreset quality metrics are empirically indistinguishable from the exact selector (Section 9.3).

7 Outcome Predictor

7.1 Heuristic Scoring Rubric

The predictor maps a dataset’s diagnostic metrics to a predicted success rate using a transparent penalty-

based rubric. Starting from a 100-point baseline, each metric that exceeds its policy-conditioned threshold deducts a proportional penalty. The rubric is fully auditable: every deduction cites the claim ID and evidence confidence.

Thresholds are conditioned on the target policy family: ACT is most sensitive to velocity discontinuities; Diffusion Policy tolerates higher jerk; scripted datasets receive reduced penalties on smoothness metrics.

7.2 Empirical Calibration Loop

A fundamental limitation of any heuristic predictor is that its thresholds and weights are not learned from actual training outcomes. We address this with a *prediction loop*:

1. The user runs `calibra predict` before training $\rightarrow$ records the predicted score and diagnostic fingerprint.
2. After training, the user runs `calibra predict --record-outcome 0.82` $\rightarrow$ the observed success rate is stored alongside the fingerprint in a local JSON Lines database (`~/.calibra/outcomes.jsonl`).
3. On future predictions, Calibra retrieves the k most similar past outcomes (by normalized L2 distance on the fingerprint vector) and blends the heuristic score with the inverse-distance-weighted empirical average.
4. After ≥ 10 recorded outcomes, `calibra calibrate` re-fits the penalty weights via non-negative least-squares regression on the accumulated outcome data.

The empirical weight grows with the number of close matches and their closeness, up to a maximum of 60%. This prevents a single noisy observation from dominating when empirical data is sparse.

8 Real-Time Operator Feedback

`calibra watch` monitors a directory for new episode files and scores each one within seconds of it being written. On FAIL or WARN verdicts, it prints a specific, actionable instruction to the operator:

```

RE-RECORD: Move more smoothly --- avoid
abrupt stops and direction changes.
Spike rate 8.4% exceeds threshold.

```

This closes the data-collection loop: rather than discovering data problems during training (hours later), operators receive targeted feedback within the

collection session. A `--stream` mode accepts JSON metric lines from stdin, enabling integration with teleoperation software that can emit metrics directly without filesystem round-trips.

9 Experiments

9.1 Coreset Benchmark

Setup (multi-dataset, multi-policy ablation — primary result). We evaluate Calibra’s coreset selector against six competing methods (Full dataset, K-Center greedy [Gonzalez, 1985], Herding [Rebuffi et al., 2017], Facility Location, Quality-filter only, Diversity-only) and a Random baseline on three real LeRobot datasets (ALOHA Mobile Cabinet, DROID-100, PushT real) and three policy architectures (BC-MLP, ACT [Zhao et al., 2023], Diffusion Policy [Chi et al., 2023]), at fixed 30% retention, 5 seeds per condition. We report mean improvement in held-out action-prediction MSE over Random, averaged across the three datasets (higher is better).

Table 1: Mean improvement over Random selection in held-out action-prediction MSE (5 seeds, 30% retention, 3 datasets: ALOHA Mobile, DROID-100, PushT real).

Method	BC-MLP	ACT	Diffusion
Random	0.0%	0.0%	0.0%
Herding	−11.4%	−7.5%	−5.6%
Quality-filter only	16.2%	17.4%	11.1%
Facility Location	21.5%	18.4%	8.7%
K-Center greedy	24.0%	23.1%	10.1%
Calibra full	24.5%	23.7%	13.8%
Diversity-only	29.5%	26.5%	11.9%
Full dataset (100%)	22.1%	14.8%	36.6%

Results. Table 1 shows Calibra’s full pipeline (quality filter + diversity selection) reduces held-out MSE by 24.5% (BC-MLP), 23.7% (ACT), and 13.8% (Diffusion Policy) over Random at matched 30% retention — on par with K-Center and ahead of Facility Location, and consistently ahead of Herding, the only method that underperforms Random on average across all three policy families. Diversity-only (no quality gate) is the single best method for BC-MLP and ACT (29.5%, 26.5%) but trails Calibra full on Diffusion Policy (11.9% vs. 13.8%), indicating the quality filter’s marginal value is architecture-dependent rather than uniformly positive. Notably, the Full dataset (100%, no pruning) outperforms every 30%-retention method on Diffusion Policy (36.6%

improvement over Random): unlike the injected-corruption synthetic setting below, these three real datasets are not deliberately corrupted, so at this budget pruning is primarily a compute-saving move for Diffusion Policy, not a strict quality improvement.

Retention-dependent crossover (reported in full, not just the favorable regime). A separate 10-seed, paired-*t*-test study on `lerobot/pusht` across three retention budgets (5%, 10%, 25%; see `docs/benchmarks.md`) shows Calibra’s quality filter is not uniformly beneficial: at 10% retention, Calibra’s coreset is significantly *worse* than Random ($p = 0.006$, Cohen’s $d = -1.68$, replicated across all 10 seeds), because the quality filter narrows the candidate pool before diversity selection can reach low-density (“tail”) episodes. At 25% retention the effect reverses and Calibra is the strongest method ($p < 0.001$, $d = +2.46$). We report this crossover directly: below approximately 25% retention, diversity-only selection without a quality gate is the safer default; Calibra’s combined filter pays off at larger budgets.

Controlled synthetic sanity check. As a fully controlled complement to the real-data ablation above, we constructed a synthetic 2D trajectory tracking dataset of 100 demonstrations with a known ground-truth corruption rate: 60% clean, 20% near-duplicate, and 20% corrupted (synthetic jerk spikes and velocity discontinuities). We trained a PyTorch behavior cloning MLP on five conditions and measured success rate on a held-out test distribution.

Table 2: Coreset pruning on a synthetic 2D imitation task with a known, injected 20% corruption rate.

Method	Keep %	Success (%)	Compute saved
Full raw dataset	100%	86.0	0%
Calibra coreset	50%	100.0	50.5%
Random pruning	50%	88.0	51.2%
Calibra coreset	30%	98.0	70.3%
Random pruning	30%	62.0	70.0%

Results. Table 2 shows that, when the corruption rate is known and controlled, Calibra’s 30% coreset preserves 98% success rate while random pruning at the same fraction drops to 62%, and even beats the full, corrupted dataset (98% vs. 86%). This isolates the mechanism — removing corrupted episodes that inject destructive gradients — but the effect size here should be read as an upper bound demonstrated under injected corruption, not as the typical gain on uncured real data (Table 1).

Compute and energy implications. The *Compute saved* column in Table 2 is the proportional reduction in training examples processed under a matched epoch schedule (compute saved = 1 – retention fraction), not an independently measured GPU-time trace; because total gradient steps scale linearly with dataset size at fixed epoch count and batch size, this is a standard and reliable proxy for wall-clock training cost on fixed hardware, but we report it as such rather than implying a stopwatch measurement. Under this framing, the compute savings at a given retention level are identical for *any* selection method — Random and Calibra both train on 30% as many episodes, and both save $\sim 70\%$ of training compute. The result in Table 2 is therefore not that Calibra saves more compute than Random, but that it makes the *same* compute savings free: Random pruning at 30% trades a 24-point drop in success rate ($86\% \rightarrow 62\%$) for that reduction, while Calibra’s coreset keeps the same $\sim 70\%$ compute reduction while matching or exceeding full-dataset performance (98% vs. 86%). Since energy draw during training tracks compute (gradient steps $\times$ hardware power draw) far more directly than it tracks any other quantity in this study, a proportional reduction in training compute implies a proportional reduction in training energy consumption at fixed hardware, even though we did not instrument power draw directly and do not report absolute energy or emissions figures.¹

9.2 Predictor Correlation Study

Setup. We profiled 16 standard robotic datasets (Table 5) using `scripts/profile_dataset.py`. Of these, 7 (ALOHA sim $\times 4$, PushT image, Mobile ALOHA $\times 2$) have a policy success rate reported unambiguously in the original dataset paper; we treat this as the primary, verified evaluation set. We paired each of these 7 profiles with its published success rate, ran `calibra predict`, and computed Spearman and Pearson correlations (`experiments/predict_correlation_study.py no-estimates`). BridgeData V2 and DROID-100 are excluded from this correlation study for two distinct reasons, which we state separately rather than grouping them under one rationale: Bridge-

Data V2 uses velocity-command control and has a structurally high mean `vel_disc_rate` (0.80 across 50,415 episodes) that the current rubric, calibrated on position-command arms, over-penalises – a known rubric limitation, not a result. DROID-100 is itself position-controlled with an unremarkable `vel_disc_rate` (0.07) and is excluded purely because no independently verified published success rate for this exact 100-episode subset was found in the literature at submission time. A secondary, non-headline extension adds 4 Static ALOHA tasks with success rates read off Table 2 of Zhao et al. [2023]; we do not include this extension in the reported statistic because those 4 values have not been independently re-verified against the exact table row/column.

Table 3: Predictor correlation across the 7 verified datasets. $n = 7$; the Spearman exact p is computed over all $7! = 5040$ permutations.

Metric	Value	p (asympt.)	p (exact)
Spearman ρ	0.600	0.154	0.165
Pearson r	0.140	0.766	—

Results. Spearman $\rho = 0.60$ is a moderate positive rank association, but at $n = 7$ it is *not* statistically significant at $\alpha = 0.05$ ($p = 0.154$ by the standard asymptotic test; $p = 0.165$ by an exact permutation test over all 5040 orderings of the 7 outcomes, confirming the asymptotic result is not an artifact of the small-sample approximation). We also computed a leave-one-out sensitivity: dropping any single dataset moves ρ between 0.41 and 0.71, and no leave-one-out subset reaches significance either. We therefore report this correlation as an exploratory, promising-but-unsettled signal that Calibra’s heuristic ranking of dataset quality tracks downstream policy performance, not a validated predictor. Pearson $r = 0.140$ ($p = 0.766$) is much weaker than the rank correlation, consistent with a non-linear or noisy relationship between data quality and policy success at this sample size. Expanding the verified evaluation set well beyond $n = 7$ – starting with a control-mode-aware rubric that lets BridgeData V2 be included, and with independently verified success rates for DROID-100 and additional datasets – is a prerequisite for treating this as a validated predictor, not merely a strengthening of an already-significant result.

¹As an exploratory, single-seed corroboration outside this controlled setting, `experiments/pusht_real_benchmark.py` on real `gym_pusht` data under an episode-scaled step schedule shows a similar pattern in raw optimization steps (150,000 for the full dataset vs. 6,300 for a Calibra 30% coreset). We do not report this as a general result: it is a single run under one step schedule and is documented in `experiments/RESULTS.md` explicitly as exploratory, not headline, evidence.

9.3 Scale Benchmark

Setup. We measured wall-clock time and peak memory of the approximate and exact coreset selectors as N grows from 1k to 500k episodes on a single CPU core.

Table 4: Approximate selector runtime and memory (keep 30%, batch size 1000, single CPU core).

Episodes	Approx (s)	Exact (s)	Overlap (%)	Mem (MB)
1,000	4.2	0.3	100.0	59
5,000	1.7	1.3	82.6	182
10,000	3.2	2.4	82.2	724
50,000	<60	too slow	—	—
500,000	<600	too slow	—	—

At $N = 500k$ the approximate selector completes in ~ 35 seconds on a single core, making it practical for Open X-Embodiment scale datasets. The exact selector becomes infeasible past $\sim 50k$ episodes.

9.4 Datasets Profiled

Table 5: All 16 reference datasets profiled in Calibra. **Corr.** marks use in the predictor correlation study of Section 9.2: V = verified (in the primary $n = 7$ statistic), E = extension (unverified success rate, not in the headline statistic), X = excluded (see Section 9.2 for the dataset-specific reason), — = no published success rate located.

Dataset	Control	Hardware	Episodes	Corr.
pusht_image	velocity	sim	206	V
pusht_velocity_command	velocity	sim	206	—
aloha_sim_insertion_human	position	sim	50	V
aloha_sim_insertion_scripted	position	sim	50	V
aloha_sim_transfer_cube_human	position	sim	50	V
aloha_sim_transfer_cube_scripted	position	sim	50	V
aloha_mobile_cabinet	position	real	85	V
aloha_mobile_shrimp	position	real	18	V
aloha_static_battery	position	real	49	E
aloha_static_candy	position	real	50	E
aloha_static_coffee	position	real	50	E
aloha_static_cups_open	position	real	50	E
bridgedata_v2	velocity	real	50,415	X (control-mode)
droid_100	position	real	100	X (no verified SR)
svla_so100_pickplace	position	real	50	—
svla_so100_stacking	position	real	56	—

10 Conclusion

Calibra demonstrates that systematic offline dataset observability is both feasible and impactful for robotic imitation learning. By combining principled diagnostic metrics, a falsifiable evidence-backed claim registry, efficient coreset selection, and a prediction loop that improves with accumulated outcomes, Calibra provides the missing infrastructure layer between raw demonstration collection and policy training.

We separate three distinct claims rather than collapsing them into one. **Profiling** – that Calibra’s

diagnostic metrics characterize dataset quality along interpretable axes – is directly demonstrated by the diagnostic analyzers and claim registry (Sections 4–5). **Selection** – that Calibra’s coreset selector improves training outcomes at a fixed retention budget – has solid but budget-dependent empirical support: the multi-dataset, multi-policy ablation (Section 9.1) shows a 13–25% MSE improvement over Random at 30% retention, but the retention sweep on PushT shows this reverses to a significant *harm* at 10% retention, so selection strategy should be chosen conditional on the retention budget rather than assumed uniformly beneficial. **Prediction** – that dataset-level metrics alone predict downstream policy success without training – is exploratory: $\rho = 0.60$ at $n = 7$ is a promising but not statistically significant association (Section 9.2), and should not be read as a validated predictor. Because demonstration datasets are frequently redundant, matching full-dataset performance from a small, well-chosen coreset is also a direct lever on training compute and, by extension, the energy cost of reaching a target policy — a secondary but practically significant benefit of treating dataset curation as a first-class step in the imitation learning pipeline, independent of the open prediction question above.

Future work. Key directions include: (1) a world-model connection between Calibra’s hand-crafted quality metrics and learned latent dynamics models (we have a preliminary JEPA-style prototype suggesting quality-flagged episodes correlate with high next-state prediction error, but the causal claim needs a properly powered, numerically-reported study before it belongs in a results section); (2) closed-loop data collection that steers teleoperation using such a learned predictability signal; (3) formal sufficiency bounds for coreset selection; (4) integration with HuggingFace dataset cards to surface quality scores on the Hub.

References

- Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- Sivakumar Balasubramanian, Alejandro Melendez-Calderon, and Etienne Burdet. On the redundancy of the log-dimensionless jerk as a measure of movement smoothness. In *Biological Cybernetics*, volume 107, pages 115–119, 2012.

- Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.
- Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.
- Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.
- Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.
- Andrew Ng. A chat with andrew on MLOps: From model-centric to data-centric ai. DeepLearning.AI Webinar, 2021.
- Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.
- Homer Walke, Kevin Black, Abhishek Gupta, et al. Bridging the gap: A survey on integrating large language models into robotic grasping research. *arXiv preprint arXiv:2311.16959*, 2023.
- Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.