Auditing Offline Demonstration Pruning for Online Robot Deployment

Joy Zheyun Yang [*1] Socrates Osorio [*2]

Abstract

Pruning offline demonstrations is a deployment decision. A scoring rule can beat random pruning and still lose to a cheap baseline that would be the better online choice. We present a four-gate offline-to-online audit for pruning methods. The gates are a paired random comparison, a cheap-baseline audit, a held-out stress test, and a mechanism check. As a case study we use TRAK-Traj, a trajectory-level adaptation of TRAK-style attribution. TRAK-Traj beats matched random pruning on curated RoboMimic Can MH by 4.7 percentage points across 10 paired seeds and 300 rollouts per condition, winning 9 of 10 seeds (paired $t(9) = 2.43$, one-sided $p = 0.019$). The audit then changes the recommendation. TracIn-style scoring ties TRAK-Traj, trajectory length is stronger on curated Can, and a held-out mixed-quality block shows length outperforming both TRAK-Traj and random. A secondary Square MH audit and a Lift ceiling check are consistent with the same lesson: pruning can help, but the practical winner depends on the cheap baselines. A cost audit shows that attribution adds a scoring pass that length and loss do not need. The contribution is a reproducible audit template for data-curation claims before robot deployment.

1. Introduction

Offline robot learning is often limited less by the amount of data than by the quality and consistency of the demonstrations available for training. Behavior cloning and imitation learning can turn logged demonstrations into deployable policies (Pomerleau, 1989; Ross et al., 2011; Osa et al., 2018), but multi-human datasets can contain strong demonstrations, hesitant executions, corrective behavior, and failed strategies (Mandlekar et al., 2021). Training on all trajectories is the default, but it is not a neutral choice. The learner may average across inconsistent action modes or spend a limited optimization budget fitting trajectories that are not useful for deployment.

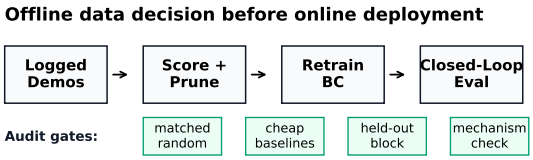

Figure 1. Offline-to-online audit view. The intervention is an offline data decision and the outcome is closed-loop success after retraining. We require matched random, cheap-baseline, held-out, and mechanism checks before turning a pruning result into a deployment recommendation.

This paper asks how to evaluate the offline decision to prune demonstrations under a fixed retraining budget. This fits the offline-to-online workflow studied in offline RL and offline imitation learning, where a batch dataset is processed before the learned policy is evaluated in closed-loop deployment (Fu et al., 2020; Levine et al., 2020). We study the simplest operational form. We assign one score to each trajectory, remove the lowest-scoring trajectories, retrain behavior cloning from scratch on the retained set, and ask whether the resulting online policy improves. TRAK-Traj is our attribution case study, but the audit is the main contribution. We propose a four-gate protocol that checks matched random pruning, cheap deterministic baselines, held-out mixed-quality seeds, and mechanism evidence before turning a pruning result into a deployment recommendation.

We propose TRAK-Traj, a trajectory-level adaptation of TRAK (Park et al., 2023). The method replaces per-example attribution with per-trajectory attribution, the natural unit at which robot demonstrations are collected and filtered. The goal is not to solve online adaptation directly. The goal is to improve the

[*]Equal contribution [1]University of Oxford, Oxford, United Kingdom [2]University of California, Berkeley, California, USA. Correspondence to: Joy Zheyun Yang <joy@robots.ox.ac.uk>.

offline data substrate from which policies are learned.

The key design choice is that the evaluation is cold-start and paired. After pruning, every policy is retrained from random initialization with the same architecture and budget. Random pruning and TRAK-Traj pruning share the same seed and prune ratio, so the comparison asks whether the ranking itself adds value. This is a demanding test for a 10% pruning method, because the method removes only 9 of 90 curated demonstrations and leaves little room for large effects.

Our primary evidence is a high-rollout cold-start evaluation on RoboMimic Can MH (Mandlekar et al., 2021). On curated demonstrations, TRAK-Traj improves over matched random pruning in 9 of 10 seeds, with complementary parametric, non-parametric, permutation, bootstrap, and Bayesian checks supporting a positive effect. A 10-seed deterministic-baseline audit then changes the practical recommendation. Loss is slightly below TRAK-Traj, TracIn-style scoring ties it, and trajectory length is higher. This motivates a sharper conclusion. Attribution is a useful candidate signal in the controlled curated protocol, but random-only evaluation would overstate its practical value.

Our contributions are the following.

- A four-gate cold-start audit protocol for offline robot demonstration pruning.
- A trajectory-level TRAK case study showing a statistically significant advantage over matched random pruning, with a paired analysis that separates seed-level evidence from pooled rollout counts.
- A completed cheap-baseline audit, a secondary Square and Lift cross-regime check, and a cost audit showing that this positive random-baseline result does not imply practical dominance.
- A mechanism analysis explaining why trajectory length is strong in Can MH and why attribution and length prune largely different demonstrations.

2. Related Work

Offline robot imitation and decision making. Behavior cloning from human demonstrations remains a common foundation for robot manipulation (Pomerleau, 1989; Ross et al., 2011; Osa et al., 2018), but performance is sensitive to demonstration source and quality. RoboMimic provides standardized multi-human datasets and evaluation protocols for studying these effects (Mandlekar et al., 2021), while modern robot policy classes such as diffusion policies and action-chunking transformers further increase the importance of the offline training distribution (Chi et al., 2023; Zhao et al., 2023). Offline RL benchmarks and algorithms similarly emphasize that batch data quality and support determine whether a policy can be safely improved before deployment (Fu et al., 2020; Kumar et al., 2020; Kostrikov et al., 2022). Recent imitation-learning work formalizes data quality as a central object rather than a nuisance variable (Belkhale et al., 2023). Data generation and curation systems such as MimicGen (Mandlekar et al., 2023), IntervenGen (Hoque et al., 2024), and ReMix (Hejna et al., 2025) show that policy quality depends on which demonstrations are included, not merely on dataset size.

Training data attribution. Influence functions (Koh & Liang, 2017), TracIn (Pruthi et al., 2020), datamodels (Ilyas et al., 2022), and TRAK (Park et al., 2023) estimate how training data affect model behavior. TRAK scales attribution through projected gradients and linearized predictors. TRAK-Traj follows this line but changes the attribution unit from examples to complete trajectories, aligning the method with robot data collection and pruning. This lineage also motivates caution, because influence-style linearizations can be fragile in deep nonconvex models or can answer a proxy question rather than exact leave-one-out retraining (Basu et al., 2021; Bae et al., 2022).

Data valuation and pruning. Data Shapley (Ghorbani & Zou, 2019), DVRL (Yoon et al., 2020), coreset-style selection (Sener & Savarese, 2018), forgetting events (Toneva et al., 2019), dataset cartography (Swayamdipta et al., 2020), and gradient-matching subset selection (Killamsetty et al., 2021) study how to value or select training data. Our work is closest in spirit to practical pruning. We score once, remove low-value units, and retrain under the same budget. The audit result is aligned with recent data-centric lessons. Simple dataset statistics can be strong baselines and should be reported before claiming that a learned valuation method is practically dominant.

Evaluation methodology. Deep RL evaluation studies repeatedly find that seed-level variation can dominate apparent method differences (Henderson et al., 2018; Agarwal et al., 2021). Broader machine learning evaluation work warns that the experimental unit, the statistical power, and the multiplicity of comparisons must match the scientific claim (Cohen, 1988; Benjamini & Hochberg, 1995). Robot-learning evaluation guidelines likewise call for explicit rollout counts, suc-

cess criteria, and statistical analysis rather than only aggregate success rates (Kress-Gazit et al., 2024). We instantiate these principles for offline demonstration pruning.

3. Method

3.1. Problem Setup

Let $\mathcal{D} = \{\tau_i\}_{i=1}^{N}$ be an offline imitation dataset, where each trajectory is

$$\tau_i = \{(o_t, a_t)\}_{t=1}^{T_i}. \tag{1}$$

We train a behavior cloning policy f_θ and seek a scalar score s_i for each full trajectory. For prune ratio p, we retain the top $(1-p)N$ trajectories by score and retrain from random initialization using unchanged architecture, optimizer, and training budget. This cold-start design makes the evaluation conservative. Gains must come from the retained data, not from warm-starting or per-condition tuning.

3.2. Trajectory-Level TRAK

TRAK-Traj adapts TRAK-style attribution to the demonstration level. For checkpoint c, we define a trajectory surrogate objective $q_{\theta^{(c)}}(\tau_i)$ and compute its parameter gradient

$$g_i^{(c)} = \nabla_\theta q_{\theta^{(c)}}(\tau_i). \tag{2}$$

In our experiments, q is computed over sampled transitions from the trajectory, and gradients are taken with respect to the policy head for efficiency. For behavior cloning, the surrogate is label-agnostic,

$$q_\theta(\tau_i) = \frac{1}{|\mathcal{S}_i|} \sum_{(o,a)\in\mathcal{S}_i} \|f_\theta(o)\|_2^2, \tag{3}$$

where $\mathcal{S}_i$ is a deterministic sample of 16 transitions from trajectory τ_i. This follows the D-TRAK-style intuition that gradients of model output energy can provide a useful local linearization without using the action label directly in the attribution objective (Zheng et al., 2024). The policy is still trained with standard behavior cloning loss. The surrogate is used only for scoring.

We project gradients using a Johnson–Lindenstrauss matrix $R \in \mathbb{R}^{d\times P}$,

$$\phi_i^{(c)} = R g_i^{(c)}, \quad d = 2048. \tag{4}$$

For each checkpoint, projected features are ridge-preconditioned,

$$H^{(c)} = \frac{1}{N} \sum_i \phi_i^{(c)} \phi_i^{(c)\top} + \lambda I. \tag{5}$$

Given target features from the held-out better_valid split, the score is

$$s_i = \frac{1}{C} \sum_{c=1}^{C} \phi_i^{(c)\top} (H^{(c)})^{-1} \psi^{(c)}. \tag{6}$$

Operationally, the implementation featurizes the exact training trajectories used by the baseline and the held-out target trajectories, builds the ridge matrix from the training features, solves for the target direction, and scores each training trajectory by its dot product with that direction. Higher scores are kept and lower scores are pruned.

We average ranks across three seed offsets $\{0, 97, 211\}$ and multiple saved checkpoints to reduce variance. Before aggregation, member score vectors are winsorized at ± 2 standard deviations, then converted to ranks and averaged. This makes the pruning decision depend on relative order rather than raw score scale, which varies across checkpoints and projection seeds.

Why target high-quality validation trajectories? The target set is not used for retraining. It defines the behavior whose linearized features the method should preserve. In Can MH, better_valid provides a small held-out set from the same operator-quality tier as the curated training split. In the mixed-quality experiment, using the same target split tests whether a high-quality target direction can identify useful demonstrations inside the full heterogeneous training set. This is a practical but not assumption-free design, because it presumes access to a held-out validation subset whose quality is representative of the desired behavior.

3.3. Four-Gate Pruning Audit

The full pipeline is the following.

1. Train a baseline behavior cloning policy on the available offline dataset.
2. Score each trajectory with TRAK-Traj.
3. Remove the bottom p fraction of trajectories.
4. Retrain the policy from scratch on the retained subset.
5. Evaluate closed-loop success with strict success parsing.

Around this pipeline we apply four gates. Gate 1 is a matched random comparison that tests whether the ranking adds information beyond deleting the same amount of data. Gate 2 is a cheap-baseline audit

against loss, TracIn-style scores, and trajectory length, which define the practical null hypothesis for an expensive attribution pass. Gate 3 is a held-out stress test on new seeds and a more heterogeneous data regime. Gate 4 is a mechanism inspection of the lengths, quality tiers, and pruned-set overlaps. Passing Gate 1 supports an attribution-over-random claim. Passing the remaining gates, together with cost accounting, is required before turning that claim into a deployment recommendation.

4. Experimental Setup

Task. We evaluate on RoboMimic Can MH (PickPlaceCan), a multi-human manipulation dataset with 300 demonstrations across operator quality tiers. We use the standard 90/10 train-validation split. The curated condition uses the 90 "better" training demonstrations. The mixed-quality condition uses all 270 training demonstrations. We also include secondary RoboMimic Square MH and Lift MH audits to test whether the reporting conclusions extend beyond Can. These tasks use the same low-dimensional behavior cloning setup and 300 rollouts per condition. They are not presented as a second confirmatory TRAK-Traj win. They test whether pruning, cheap baselines, and paired seed-level reporting remain informative on other manipulation tasks.

Why lead with curated data? The curated condition is not the most realistic deployment setting, but it is the cleanest controlled test of whether attribution improves over random removal when every demonstration is nominally useful. Because only 9 of 90 trajectories are removed, random pruning is already a strong baseline and the expected effect is small. The mixed-quality condition is closer to the motivating data-curation use case, but its fixed-budget behavior cloning baselines can collapse. We therefore treat curated Can as the confirmatory comparison and mixed-quality Can as a stress test.

Policy. All conditions use a residual MLP behavior cloning policy with 5 residual layers, hidden dimension 768, dropout 0.05, observation history 2, and action history 1. Training uses Adam with learning rate 3×10^{-4}, batch size 128, and 8000 gradient updates.

Attribution configuration. TRAK-Traj uses projection dimension 2048, 16 trajectory samples for gradient estimation, fp32 precision, three split seed offsets $\{0, 97, 211\}$, rank-mean aggregation, head-only gradients, and score winsorization at $\pm 2\sigma$.

Table 1. Can MH curated demonstrations at 10% prune ratio. Each seed uses 300 rollouts per condition.

Seed	Baseline	Random	TRAK	Δ
607	.863	.770	.800	+.030
608	.723	.780	.823	+.043
609	.617	.797	.827	+.030
610	.793	.680	.803	+.123
611	.800	.770	.870	+.100
612	.783	.727	.863	+.137
613	.787	.767	.783	+.017
614	.837	.847	.773	-.073
615	.770	.773	.830	+.057
616	.810	.787	.797	+.010
Mean	.778	.770	.817	+.047

Evaluation. Each reported condition is evaluated with 300 environment rollouts using state-anchored reset, per-rollout seeds, and strict success parsing. This follows recent calls for robot-learning evaluations to report rollout counts, success criteria, and statistical analysis rather than only aggregate success rates (Kress-Gazit et al., 2024). In the curated Can condition, we run 10 paired seeds (607–616). In the mixed-quality condition, we use seeds 701–710 as an exploratory scan and seeds 711–716 as a held-out audit block. Square uses seeds 601–610 at 10% and seeds 601–605 and 607–609 at 30%. Lift uses seeds 621–625 as a ceiling check. We treat mixed-quality results as diagnostic, because several seeds are near the floor and the mixed-quality setting has higher seed-to-seed variance.

Baselines. The main confirmatory baseline is matched random pruning. For each seed and prune ratio, a random subset of the same size is removed and the policy is retrained from scratch. This directly tests whether attribution provides information beyond deleting the same amount of data. We also audit cheap deterministic comparators, including loss, trajectory length, and TracIn-style scores. That audit is diagnostic rather than primary. It tests whether a positive result against random survives contact with obvious low-cost alternatives.

5. Results

5.1. Curated Demonstrations: TRAK Beats Random

Figure 2 and Table 1 show the primary result. TRAK-Traj improves over random pruning in 9 of 10 seeds. The mean gain over random is 4.7 percentage points, despite pruning only 9 of 90 trajectories. TRAK-Traj also exceeds the unpruned baseline mean and is higher than the unpruned baseline in 6 of 10 seeds. We read this as evidence that the ranking can remove trajectories that are harmful or redundant for this training regime. By itself, it is not evidence that the same score

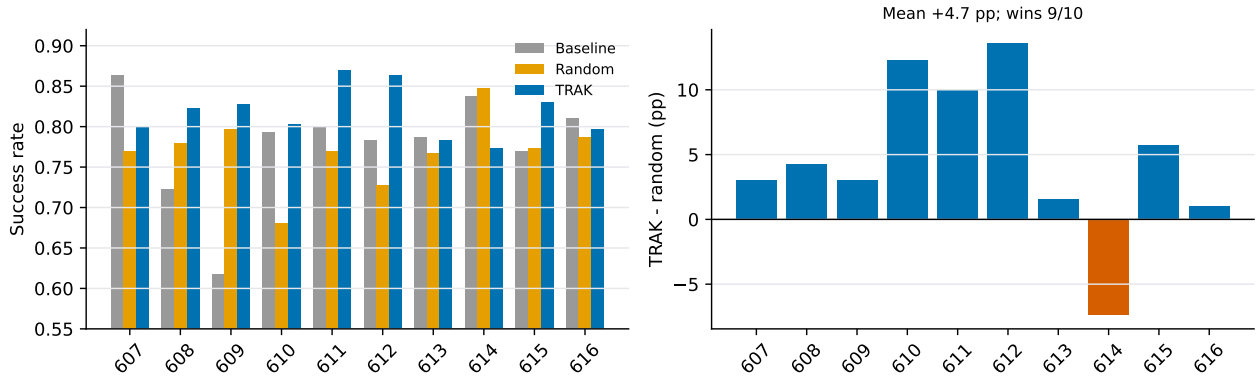

Figure 2. Primary Can MH curated-data result. Left: per-seed closed-loop success rates for the unpruned baseline, matched random pruning, and TRAK-Traj pruning at 10%. Right: paired seed-level deltas show that TRAK-Traj beats random pruning in 9 of 10 seeds, with a 4.7 percentage point mean advantage.

Table 2. Statistical checks for the curated Can MH comparison. Tests are one-sided for the directional hypothesis TRAK > random.

Check	Statistic	Result
Paired t-test	$t(9) = 2.43$	$p = 0.019$
Wilcoxon	p_{1s}	0.019
Sign test	9/10 wins	$p = 0.011$
Permutation	50K draws	$p = 0.023$
Bootstrap CI	10K resamples	$[+1.0, +8.3]$ pp
Cohen's d_z	paired effect	0.77
Bayesian	$P(\mu > 0)$	0.993

will dominate all pruning heuristics or all robot tasks.

Table 2 summarizes complementary statistical checks. The conclusion is not driven by one test family. Parametric, non-parametric, permutation, bootstrap, and Bayesian summaries (Benavoli et al., 2017) all support a positive TRAK-Traj advantage in this paired setting. We treat these as robustness checks on one paired claim rather than separate discoveries. We therefore do not apply a multiple-testing procedure such as Benjamini–Hochberg false-discovery-rate control (Benjamini & Hochberg, 1995). The effect is modest in absolute size but meaningful for a minimal intervention. The training set changes by only 10%, and every pruned policy is retrained from scratch. Power calculations help keep this scope realistic (Cohen, 1988).

5.2. Why Pairing Matters

Robot learning papers often evaluate few training seeds with many rollouts. Pooling rollouts can make a method comparison look more certain than the seed-level evidence supports. A rollout is a binary outcome from one trained policy. The method comparison is replicated only when the training pipeline is rerun under a new seed, split, or task. Let Y_s^T and Y_s^R denote success rates for TRAK-Traj and random pruning under seed s. The paired comparison studies the seed-level difference $D_s = Y_s^T - Y_s^R$, which controls for seed-level factors shared by both methods, including the data split, the training initialization family, the pruning ratio, and the evaluation seed schedule.

Table 3. Pooled rollout analysis reports stronger significance because it treats 3000 rollouts per method as independent observations. We report it as a diagnostic, not as the primary test.

Unit	Random	TRAK	Test result
Rollouts	2309/3000	2451/3000	$z = 4.53$, $p < 10^{-5}$
Seeds	10 paired	10 paired	$t(9) = 2.43$, $p = 0.019$

Table 4. Curated Can MH cheap-heuristic audit at 10% pruning. Each entry uses 300 rollouts. This table is an audit, not the primary confirmatory comparison.

Seed	TRAK	Loss	TracIn	Length
607	.800	.777	.870	.857
608	.823	.750	.827	.873
609	.827	.773	.833	.847
610	.803	.857	.827	.897
611	.870	.887	.867	.890
612	.863	.800	.770	.803
613	.783	.703	.777	.813
614	.773	.823	.753	.777
615	.830	.793	.867	.827
616	.797	.860	.780	.870
Completed seeds	10	10	10	10
Mean available	.817	.802	.817	.845

Table 3 shows the scale of the difference. The pooled rollout analysis asks whether two bags of binary outcomes differ. The paired analysis asks whether the scoring rule tends to improve a retrained policy when seed and pruning ratio are held fixed. The curated result remains positive under the paired test, but its evidential strength is appropriately more modest than the pooled view would suggest.

Table 5. Paired cheap-baseline audit on the same 10 curated seeds. Δ is TRAK-Traj minus the comparator.

Comparator	TRAK	Other	Δ	Paired test
Loss	.817	.802	+1.5pp	$t(9) = 0.83$, $p_{2s} = .430$
TracIn	.817	.817	-0.0pp	$t(9) = -0.01$, $p_{2s} = .988$
Length	.817	.845	-2.8pp	$t(9) = -2.07$, $p_{2s} = .069$

Table 6. Curated Can MH sensitivity across prune ratios. The 10% ratio is the primary minimal-pruning setting. The 20% and 30% ratios are exploratory.

Prune	Random	TRAK	Δ	Wins
10%	.770	.817	+.047	9/10
20%	.774	.751	-.022	3/10
30%	.749	.777	+.028	5/10

5.3. Cheap-Heuristic Audit

Tables 4 and 5 are the current check against simple non-random comparators. They do not support a broad claim that TRAK-Traj dominates every cheap heuristic. Loss is slightly below TRAK-Traj on mean success (.802 versus .817), TracIn-style scoring ties TRAK-Traj (.817), and trajectory length is higher (.845). On the same paired seeds, TRAK-Traj trails length by 2.8 percentage points and wins only 2 of 10 paired comparisons. This strengthens the need for the scoped claim used here. The established result is that TRAK-Traj provides a reproducible attribution signal beyond matched random pruning in the curated paired protocol. Practical dominance over cheaper scoring rules is not established by these experiments. For data curation, cheap baselines are not a courtesy. They define the practical null hypothesis. If a scoring method requires gradients, trained checkpoints, and projection features, then beating random pruning is not enough to justify deployment.

5.4. Prune-Ratio Sensitivity

Table 6 shows that the result should not be interpreted as a monotonic pruning law. TRAK-Traj is strongest in the minimal 10% deletion regime and remains directionally positive at 30%, but underperforms random at 20%. We treat this as evidence of ranking and retraining instability rather than as a tunable success story. The current evidence supports trajectory attribution as a useful ranking signal under a controlled pruning protocol, not as an automatically optimal prune-ratio selector.

5.5. Mixed-Quality Data: Held-Out Audit

Figure 3 and Table 7 show the held-out mixed-quality audit. This block is not favorable to a TRAK-forward method claim. TRAK-Traj averages .143 success versus .162 for random and wins only 3 of 6 paired seeds. The length baseline averages .278 and wins 5 of 6 paired seeds against random. The earlier exploratory scan on seeds 701–710 was directionally positive for TRAK-Traj (4.2pp over random, 5 of 10 wins), but the held-out block shows that this effect is not stable in the full mixed-quality regime. The practical conclusion is therefore the audit conclusion. In heterogeneous Can MH, trajectory duration is the stronger and more reliable pruning signal, while TRAK-Traj remains supported only in the curated paired protocol.

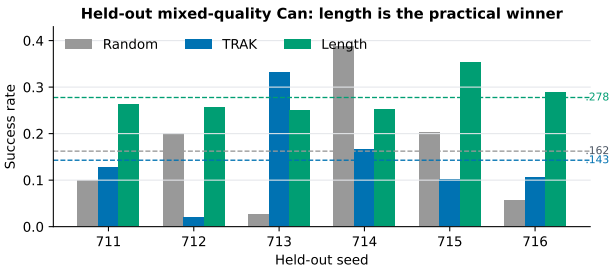

Figure 3. Held-out mixed-quality Can audit. Length is the strongest practical pruning rule in this regime. TRAK-Traj does not beat random on mean success. Dashed lines show method means.

Table 7. Held-out mixed-quality Can MH audit, seeds 711–716, at 10% pruning. Each condition uses 300 rollouts.

Seed	Baseline	Random	Length	TRAK
711	.163	.100	.263	.127
712	.023	.200	.257	.020
713	.017	.027	.250	.333
714	.303	.387	.253	.167
715	.267	.203	.353	.103
716	.100	.057	.290	.107
Mean	.146	.162	.278	.143
Wins vs. random	–	–	5/6	3/6

5.6. Cross-Regime Audit: Square and Lift

Table 8 adds two additional RoboMimic task families. The pattern supports the audit framing rather than a single-method story. On Square, pruning itself improves a weak full-data baseline. At 10%, the matched 10-seed Square block gives a small TRAK-minus-random mean delta of 0.7pp, with 5 of 10 wins and one tie ($t(9) = 0.47$). Length-minus-random is larger at 4.1pp, with 8 of 10 wins ($t(9) = 2.93$). At 30%, the eight-seed Square block gives TRAK a small mean edge over both random and length (1.9pp over random), but only 5 of 8 positive TRAK-minus-random deltas. Lift MH is a ceiling check. Baseline success is already .978, all pruning rules are within 0.2 percentage points of one another, and the row should not be read as a method win. The broader table strengthens the audit conclusion without becoming a second confirmatory TRAK-forward claim. The audit

Table 8. Cross-regime audit summary. The Role column separates the confirmatory Can result from exploratory breadth, ceiling checks, and held-out stress tests. Square and Lift use 300-rollout curated runs with matched random, TRAK-Traj, and length rows at each prune ratio. Non-contiguous seed ranges reflect completed matched-method runs, not result filtering.

Task / split	Role	Seeds	Prune	Base	Random	TRAK	Length
Can curated	Confirm.	607–616	10%	.778	.770	.817	.845
Square curated	Explor.	601–610	10%	.133	.182	.189	.224
Square curated	Explor.	601–605,607–609	30%	.137	.270	.288	.285
Lift curated	Ceiling	621–623,625	10%	.978	.983	.982	.982
Can mixed holdout	Stress	711–716	10%	.146	.162	.143	.278

Table 9. Mechanism audit for 10% pruning. Length pruning removes much longer trajectories than TRAK and, in the mixed-quality split, almost directly targets the worse tier. "Overlap" is the mean intersection between TRAK-pruned and length-pruned sets.

Split	TRAK len.	Rule len.	Overlap	Worse
Curated	139.3	196.9	0.7/9	–
Mixed	208.2	462.5	1.9/27	260/270

identifies when attribution helps, when cheap rules are stronger, and when the task is too saturated for pruning differences to matter.

5.7. Mechanism Audit

Table 9 explains why the cheap baseline is not a nuisance detail. In curated Can, all training demonstrations are from the better tier, yet length pruning removes the longest better demonstrations and achieves the best mean success. In seed 607, for example, length-only deletions include long demonstrations that TRAK ranks as ordinary or useful (236 steps at the 55th TRAK percentile, and 194 steps at the 66th percentile), while TRAK-only deletions include shorter demonstrations with low attribution scores (129 steps at the 0th TRAK percentile, and 126 steps at the 2nd percentile). In mixed-quality Can, the mechanism is starker. Across 10 seeds, length pruning removes 260 worse-tier trajectories out of 270 pruned, while TRAK-Traj removes a mixed set of 68 better, 107 okay, and 95 worse trajectories. The pruned-set overlap is small, so length is not merely approximating TRAK-Traj. It is exploiting a strong duration-quality confound in this benchmark.

5.8. Claim Boundaries

Table 10 states the intended interpretation. The current evidence is enough for a focused workshop contribution about offline data selection and evaluation, but not enough for a broad deployment claim.

6. Discussion

What the current evidence supports. The curated Can MH result is a strong focused workshop result.

Table 10. Scope of the current evidence.

The paper claims	The paper does not claim
TRAK-Traj beats matched random pruning in one 10-seed curated Can MH paired protocol.	TRAK-Traj is a general robot data-curation solution.
Trajectory-level attribution is a plausible offline data-selection primitive for imitation learning.	Attribution always beats cheap heuristics such as length, loss, or TracIn.
The mixed-quality experiment is a useful stress test that exposes a stronger length heuristic.	Mixed-quality Can is a second TRAK confirmatory win.
Minimal pruning can help, but the best prune ratio is empirical.	The score automatically selects the optimal deletion fraction.

Table 11. Measured cost audit on curated Can MH at 10% pruning. Values are median elapsed GPU-hours from 300-rollout runs. Scoring cost for TRAK rows is estimated as the first attribution row minus later prune-ratio rows from the same run. Retrain and eval dominate all rows.

Rule	Score	Retrain+eval	First-row total
Random	0	1.31	1.31
Length	≈ 0	1.23	1.23
TRAK-Traj	0.69	1.25–1.33	2.01
TRAK-len-resid	0.64	1.25	1.91

It uses 10 seeds, paired comparisons, 300 rollouts per condition, cold-start retraining, strict success parsing, and consistent wins against matched random pruning. The heuristic audit shows that simple deterministic rules are competitive, with length outperforming TRAK-Traj on mean success in this setting. The held-out mixed-quality result strengthens the audit story rather than the TRAK method story. Length beats both random and TRAK-Traj, while TRAK-Traj does not replicate its exploratory mixed-quality gain. We therefore present TRAK-Traj as an empirically supported attribution primitive in the curated protocol, not as a solved general-purpose robot-data filter.

Cost is part of the audit. Deployment decisions are constrained by compute as well as success rate. Table 11 shows that closed-loop evaluation and retrain-

ing dominate all methods, but projected-gradient scoring adds about 0.6 to 0.7 GPU-hours in this configuration. This matters because length and loss are essentially free once metadata or training losses are available. A data-attribution method should therefore clear a stronger practical bar than random deletion. It must justify both its accuracy and its extra scoring pass. Because length is effectively free and outperforms TRAK-Traj in the curated Can mean and the held-out mixed audit, a deployment user should choose length here unless attribution wins under the same audit gates.

When is length a fair baseline? Trajectory length is not a universal quality signal. It can reflect hesitation, recovery behavior, or failure in one dataset, but in another setting it may reflect task difficulty, valid long-horizon strategy, or logging conventions. This is why we do not recommend length as a general replacement for attribution. We treat it as an obligatory audit baseline whenever trajectory duration is available. If length wins, the paper should report the mechanism and use the cheaper rule. If length is unavailable, misleading, or orthogonal to quality, the same audit still applies with the domain's obvious metadata and loss-based baselines.

Limitations. The current confirmatory result is on one task family, one prune ratio, and one behavior cloning architecture. The primary comparison is against matched random pruning. Comparisons against cheap deterministic heuristics and learned data valuation methods are necessary before claiming practical dominance. The mixed-quality setting is more realistic but also more unstable. The audit protocol is architecture-agnostic, but the evidence here is not, because the experiments use a residual MLP behavior cloning policy and stronger architectures or different training budgets could change absolute success rates and the best pruning rule. Future work should expand to additional tasks, test diffusion policies, and evaluate scorer robustness when no competent baseline policy is available.

7. How To Use the Audit

The point of the audit is to prevent an offline data decision from becoming a hidden deployment risk. In a normal robot-learning workflow, the pruning method is chosen before online evaluation is available, so it is tempting to select the method that beats random in an exploratory run and report the resulting policy as practically useful. Our results show why that is insufficient. We recommend treating a pruning method as deployment-ready only if it passes four gates. First, it should beat matched random pruning in a paired seed-level comparison. Second, it should remain competitive with cheap deterministic baselines that require no attribution pass. Third, it should be checked on a held-out seed block or task variant that was not used to choose the story. Fourth, the selected and removed trajectories should be inspected for a plausible mechanism. A method that passes the first gate but fails the second or third can still be scientifically interesting, but it should be described as a candidate signal rather than a deployment recommendation.

Table 12. Deployment interpretation of the audit outcomes.

Evidence	Interpretation	Action
TRAK > random	attribution signal exists	keep studying
Length > TRAK	cheap rule stronger	do not claim dominance
Held-out reversal	mixed regime unstable	scope the claim
Mechanism found	duration-quality confound	report and exploit

Why the negative held-out result helps. The held-out mixed-quality result makes the paper less like a method advertisement and more like an audit of an offline decision. Offline-to-online learning often needs to decide what to trust before online interaction is available. A result that only says "our method beats random" gives little guidance when a cheap baseline is stronger. A result that says "the first comparison is positive, but the deployment recommendation changes after cheap-baseline and held-out checks" gives a concrete evaluation recipe for future data-curation systems. Table 12 makes the current answer explicit. On curated Can, TRAK-Traj is a valid candidate signal because it repeatedly beats matched random pruning under cold-start retraining. On mixed-quality Can, the stronger decision rule is length pruning, not TRAK-Traj.

8. Reproducibility Notes

The experiments use the public RoboMimic Can MH low-dimensional dataset and the fixed configuration files configs/pipeline/cold_start_can.yaml and configs/pipeline/full_train_can.yaml. The curated runs use task.demo_split=better_train. The mixed-quality runs use task.demo_split=train. Reported success rates are parsed from runs/*_pipeline/metrics.jsonl. The workshop figures and tables are generated from these artifacts rather than manually entered evaluation logs. The paper assets can be regenerated with python3

scripts/build_paper_assets.py. All evaluations use eval.success_contract=strict, which avoids the legacy failure mode where non-empty environment info dictionaries could be misread as successful rollouts. Table 18 summarizes the source run directories for each result block.

9. Conclusion

TRAK-Traj adapts gradient-based attribution to the trajectory level for offline robot imitation data curation. In a 10-seed cold-start curated Can MH study, it significantly outperforms matched random pruning while preserving a simple workflow of scoring, pruning, and retraining. The cheap-baseline audit, the cost audit, and the held-out mixed-quality audit show why this should remain a scoped claim. Trajectory length is stronger and cheaper in this benchmark, and TRAK-Traj does not beat random on mean success in the mixed-quality regime. The result is best read as a focused, reproducible audit showing that practical data curation needs paired testing, cheap-baseline audits, prune-ratio sensitivity, cost accounting, and broader validation across tasks and policy classes.

References

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A., and Bellemare, M. G. Deep reinforcement learning at the edge of the statistical precipice. In Advances in Neural Information Processing Systems, 2021.

Bae, J., Ng, N., Lo, A., Ghassemi, M., and Grosse, R. B. If influence functions are the answer, then what is the question? In Advances in Neural Information Processing Systems, 2022.

Basu, S., Pope, P., and Feizi, S. Influence functions in deep learning are fragile. In International Conference on Learning Representations, 2021.

Belkhale, S., Cui, Y., and Sadigh, D. Data quality in imitation learning. In Advances in Neural Information Processing Systems, 2023.

Benavoli, A., Corani, G., Demšar, J., and Zaffalon, M. Time for a change: A tutorial for comparing multiple classifiers through Bayesian analysis. Journal of Machine Learning Research, 18(77):1–36, 2017.

Benjamini, Y. and Hochberg, Y. Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B, 57(1):289–300, 1995.

Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. In Robotics: Science and Systems, 2023.

Cohen, J. Statistical Power Analysis for the Behavioral Sciences. Lawrence Erlbaum Associates, 2 edition, 1988.

Efron, B. and Tibshirani, R. J. An Introduction to the Bootstrap. Chapman and Hall/CRC, 1993.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: Datasets for deep data-driven reinforcement learning. arXiv preprint arXiv:2004.07219, 2020.

Ghorbani, A. and Zou, J. Data Shapley: Equitable valuation of data for machine learning. In International Conference on Machine Learning, 2019.

Hejna, J., Bhateja, C. A., Jiang, Y., Pertsch, K., and Sadigh, D. ReMix: Optimizing data mixtures for large scale imitation learning. In Conference on Robot Learning, 2025.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. In AAAI Conference on Artificial Intelligence, 2018.

Hoque, R., Mandlekar, A., Garrett, C., Goldberg, K., and Fox, D. IntervenGen: Interventional data generation for robust and data-efficient robot imitation learning. In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2024.

Ilyas, A., Park, S. M., Engstrom, L., Leclerc, G., and Madry, A. Datamodels: Understanding predictions with data and data with predictions. In International Conference on Machine Learning, 2022.

Killamsetty, K., Durga, S., Ramakrishnan, G., and De, A. Grad-Match: Gradient matching based data subset selection for efficient deep model training. In International Conference on Machine Learning, 2021.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In International Conference on Machine Learning, 2017.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit Q-learning. In International Conference on Learning Representations, 2022.

Kress-Gazit, H., Hashimoto, K., Kuppuswamy, N., Shah, P., Horgan, P., Richardson, G., Feng, S., and

Burchfiel, B. Robot learning as an empirical science: Best practices for policy evaluation. arXiv preprint arXiv:2409.09491, 2024.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative Q-learning for offline reinforcement learning. In Advances in Neural Information Processing Systems, 2020.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643, 2020.

Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., Fei-Fei, L., Savarese, S., Zhu, Y., and Martín-Martín, R. What matters in learning from offline human demonstrations for robot manipulation. In Conference on Robot Learning, 2021.

Mandlekar, A., Nasiriany, S., Wen, B., Akinola, I., Narang, Y., Fan, L., Zhu, Y., and Fox, D. MimicGen: A data generation system for scalable robot learning using human demonstrations. In Conference on Robot Learning, 2023.

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., and Peters, J. An algorithmic perspective on imitation learning. Foundations and Trends in Robotics, 7(1-2):1–179, 2018.

Park, S. M., Georgiev, K., Ilyas, A., Leclerc, G., and Madry, A. TRAK: Attributing model behavior at scale. In International Conference on Machine Learning, 2023.

Pomerleau, D. A. ALVINN: An autonomous land vehicle in a neural network. In Advances in Neural Information Processing Systems, 1989.

Pruthi, G., Liu, F., Kale, S., and Sundararajan, M. Estimating training data influence by tracing gradient descent. In Advances in Neural Information Processing Systems, 2020.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In International Conference on Artificial Intelligence and Statistics, 2011.

Sener, O. and Savarese, S. Active learning for convolutional neural networks: A core-set approach. In International Conference on Learning Representations, 2018.

Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A., and Choi, Y. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In Conference on Empirical Methods in Natural Language Processing, 2020.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. In International Conference on Learning Representations, 2019.

Wilcoxon, F. Individual comparisons by ranking methods. Biometrics Bulletin, 1(6):80–83, 1945.

Yoon, J., Arik, S. O., and Pfister, T. Data valuation using reinforcement learning. In International Conference on Machine Learning, 2020.

Zhao, T. Z., Kumar, V., Levine, S., and Finn, C. Learning fine-grained bimanual manipulation with low-cost hardware. In Robotics: Science and Systems, 2023.

Zheng, X., Pang, T., Du, C., Jiang, J., and Lin, M. Intriguing properties of data attribution on diffusion models. In International Conference on Learning Representations, 2024.

A. Implementation Details and Hyperparameters

Table 13 collects the full configuration used for every run in the paper. The same policy architecture and training budget are used for the baseline, random, TRAK-Traj, and cheap-baseline conditions, so a method comparison only changes the retained demonstration set. The attribution settings are held fixed across tasks. We did not tune them per task, which keeps the cross-regime audit honest.

B. Statistical Procedure Details

All confirmatory tests operate on paired seed-level deltas $D_s = Y_s^{\text{TRAK}} - Y_s^{\text{Random}}$ for the 10 curated Can seeds. The paired t-test tests whether the mean delta is positive. The Wilcoxon signed-rank test uses the signed ranks of nonzero seed deltas (Wilcoxon, 1945). The sign test uses only the number of TRAK wins. The permutation test randomly flips the sign of each paired delta under the null that method labels are exchangeable within seed, using 50K random sign-flip draws. The bootstrap confidence interval resamples the 10 seed-level deltas with replacement and reports the percentile interval over the mean (Efron & Tibshirani, 1993). The Bayesian value in Table 2 is descriptive rather than the basis for thresholding. It summarizes posterior mass on a positive mean effect under a simple seed-level normal model.

Table 13. Full configuration. Policy and training settings are shared by all conditions. Attribution settings are used only for TRAK-Traj scoring.

Component	Setting
Policy and training (all conditions)	
Policy class	residual MLP, 5 residual blocks
Hidden dimension	768
Dropout	0.05
Observation history	2 frames
Action history	1
Optimizer	Adam, learning rate 3×10^{-4}
Batch size	128
Training updates	8000
TRAK-Traj scoring	
Projection dimension	2048
Trajectory samples	16 transitions
Precision	fp32
Split seed offsets	$\{0, 97, 211\}$
Aggregation	rank-mean over checkpoints
Gradients	policy head only
Score winsorization	$\pm 2\sigma$
Closed-loop evaluation	
Rollouts per condition	300
Reset	state-anchored, per-rollout seeds
Success parsing	strict contract

C. Pooled and Paired Evidence

Figures 4 and 5 visualize the argument in Table 3. The pooled view treats 3000 rollouts per method as independent and reports a very small p-value. The paired view treats the 10 training seeds as the experimental unit and reports a more honest and more modest level of significance. The two figures make the gap between these framings explicit, which is why we report the pooled test only as a diagnostic.

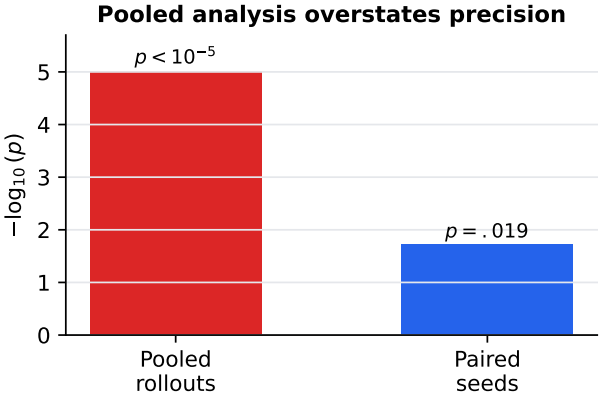

Figure 4. Pooled versus paired evidence for the curated Can MH comparison. Pooling rollouts overstates certainty relative to the seed-level paired analysis that we treat as primary.

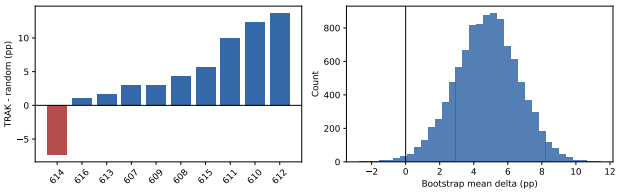

Figure 5. Summary of the paired statistical checks for the curated Can MH comparison. The parametric, nonparametric, permutation, bootstrap, and Bayesian summaries agree on the direction of the effect.

D. Exploratory Mixed-Quality Scan

The exploratory mixed-quality scan on seeds 701–710 was directionally positive for TRAK-Traj, with a mean gain of 4.2 percentage points over random and 5 of 10 positive seed deltas. We treat this as exploratory rather than confirmatory, because the held-out block on seeds 711–716 did not replicate it. Figure 6 and Table 14 report the per-seed detail. The high variance across seeds is the reason the held-out audit matters. A single exploratory scan in this regime can point in the wrong direction.

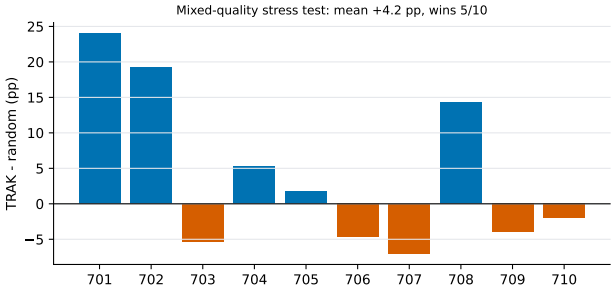

Figure 6. Exploratory mixed-quality Can MH scan, seeds 701–710, at 10% pruning. Seed-to-seed variance is large, which is why this block is exploratory and the held-out block is the reported audit.

Table 14. Per-seed exploratory mixed-quality scan, seeds 701–710, at 10% pruning. Each entry uses 300 rollouts.

Seed	Baseline	Random	TRAK	TRAK–Rand.
701	.077	.063	.303	+.240
702	.197	.080	.273	+.193
703	.097	.163	.110	-.053
704	.133	.123	.177	+.053
705	.157	.233	.250	+.017
706	.130	.107	.060	-.047
707	.033	.277	.207	-.070
708	.310	.083	.227	+.143
709	.200	.203	.163	-.040
710	.003	.073	.053	-.020
Mean	.134	.141	.182	+.042

E. Configuration and Sensitivity Analysis

Table 15 extends the prune-ratio sensitivity in the main text. The 10% setting is the calibrated primary regime. The 20% and 30% rows show that the result is not a monotonic pruning law. The reduced-rollout and out-of-seed rows are included for transparency. They are encouraging but underpowered, so we do not treat them as replications of the primary 10-seed result. The method-knob rows vary the scorer rather than the policy. A projection dimension of 2048 is a reasonable operating point, and ensembling checkpoints reduces seed-level variance.

Table 15. Configuration and sensitivity analysis on curated Can MH. Rows are grouped by what they vary. Reduced-rollout and out-of-seed rows are underpowered and are not replications of the primary 10-seed comparison. Method-knob deltas are relative to d=2048.

Config	Axis	Rand	TRAK	Δ	Seeds
Primary 607–616 (n=10), calibrated at 10%					
10%, d=2048	base	.770	.817	+4.7	607–616
20%	ratio	.774	.751	−2.2	607–616
30%	ratio	.749	.777	+2.8	607–616
10%, 100 roll.	budget	.723	.847	+12.4	607–609
Out-of-seed (n=2), not a replication					
20%, 300 roll.	seeds	.792	.908	+11.6	601/603
Method knobs (vary scorer, 10%, vs. d=2048)					
proj d=512	dim		−2 pp		607–616
proj d=4096	dim		+0.3 pp		607–616
ckpts C=1→3	ens.		std 6.2→4.1%		607–616

F. Attribution Surrogate Validation

Table 16 reports validation probes for the label-agnostic scoring surrogate in Equation (3). The probes ask whether the surrogate separates better-tier from worse-tier demonstrations, and whether the data and environment pipelines agree. The surrogate does carry a weak quality signal, but the ranking probes also show why the cheap baselines are strong. Trajectory length separates quality tiers better than the surrogate score or TracIn-style scoring on this dataset. The environment sanity checks confirm that the closed-loop evaluation is faithful to the dataset, so the audit conclusions are not an artifact of a broken pipeline.

Table 16. Attribution surrogate validation on Can MH. Tier separation and AUC probes measure how well each signal distinguishes better-tier from worse-tier demonstrations. The lower block reports environment and data sanity checks.

Validation probe	Metric	Result
Norm surrogate (Eq. 3)	tier sep. (gap)	44
MSE-loss surrogate	tier sep. (gap)	25
Action-confidence (no grad.)	rank vs. random	none
Surrogate (mean-agg.)	tier AUC	0.64
TracIn-style	tier AUC	0.78
Trajectory length	tier AUC	0.93
Surrogate score	rank corr. ρ	0.23
Action replay (sanity)	success / 20	14/20
Env–data obs. agree	mean L_2	6.2×10^{-8}
Open-loop action fit	RMSE / MSE-red.	0.134 / 90.3%

G. Relative Compute Accounting

Table 17 complements the wall-clock cost audit in the main text by reporting scoring work as a fraction of a single behavior cloning retrain. Trajectory length is free because it uses only metadata. Training loss and the gradient-based scores add a small fraction of one retrain. The dominant cost in every pipeline is retraining and closed-loop evaluation, not scoring. This is why the cost argument focuses on whether an attribution pass is justified relative to a free metadata baseline, not on the absolute scoring time.

Table 17. Relative compute accounting on curated Can MH. Scoring work is reported as a fraction of one behavior cloning retrain, which is the reference unit.

Curation rule	Gradient/forward work	Cost vs. one retrain
Trajectory length	none (metadata only)	0%
Training loss	$N\times16$ forwards	≈ 0.14%
TracIn-style	$N\times16$ head-grads	≈ 0.56%
TRAK-Traj	$N\times16\times C$ head-grads	≈ 0.56%
One BC retrain	8000×128 updates	100% (reference)

H. Source Artifact Summary

Table 18. Primary artifact sources. Each run directory contains a metrics.jsonl file with baseline and pruned evaluation records.

Result	Seeds	Run-name pattern
Curated Can, primary	607–616	can_cold_can_mh_s*
Mixed-quality scan	701–710	can_full_can_mh_s*
Mixed-quality holdout	711–716	can_full_holdout_s*
Curated heuristic audit	607–616	hcrit_can_cold_s*
Mixed heuristic audit	701–710	hcrit_can_full_s*

The heuristic audit rows are not primary evidence, because they answer a different question from the pre-specified random-pruning comparison. They show that loss, trajectory length, and TracIn-style scores are serious comparators.