Extended Abstract

Motivation Data collection for robot manipulation is expensive and time-consuming, and large-scale collection inevitably produces demonstrations of mixed quality as human operators vary in skill and strategy. A natural response is to filter or augment the dataset before training, but it is unclear whether standard curation intuitions, such as preferring demonstrations from more proficient operators, generalize across policy architectures. Prior work has studied the effect of operator quality on downstream performance but has not systematically examined how data curation interacts with model architecture.

Method We conduct a systematic study of data curation strategies for robot manipulation imitation learning using the robomimic framework on the Square nut assembly task. We evaluate two complementary approaches: heuristic trajectory-level filtering, which selects subsets of demonstrations based on properties such as trajectory length and action smoothness, and synthetic data augmentation, which generates additional demonstrations by rolling out trained policies with Gaussian noise added to executed actions. We compare these strategies across two policy architectures, BC-RNN and Diffusion Policy, using the multi-human (MH) demonstration dataset of 300 demonstrations collected by operators of varying skill.

Implementation Heuristic filters select 100 demonstrations from the full MH dataset based on trajectory-level properties computed without access to ground-truth quality labels. For synthetic augmentation, a trained policy executes noisy actions to reach diverse states while clean policy actions are recorded as training labels, keeping only the successful rollouts. We study the effect of noise level $\sigma \in \{0.02, 0.05, 0.08, 0.10\}$, the number of independently trained policy seeds used for rollout collection, and the source training dataset of the rollout policies. All augmented datasets combine 300 synthetic rollouts with the original 300 MH demonstrations for 600 total demonstrations. To characterize dataset diversity, we compute average pairwise state distance and state variance over fixed samples of 1000 states per dataset.

Results Heuristic filtering provides modest and inconsistent benefits. For BC-RNN, the best filter (most gripper changes, 0.467 ± 0.041) slightly exceeds the full dataset baseline (0.433 ± 0.062), but most filters perform above the random-100 baseline but below the full MH dataset baseline. For Diffusion Policy, all evaluated filters underperform the full dataset baseline (0.753 ± 0.025), with the best filter achieving 0.687 ± 0.050. Synthetic augmentation produces substantially larger gains when designed appropriately. Single-policy augmentation improves BC-RNN to 0.693 ± 0.052 but hurts Diffusion Policy (0.647 ± 0.019). Increasing noise level and policy diversity both improve Diffusion Policy performance, with higher noise levels generally increasing state space coverage and success rate, while BC-RNN shows large gains across all noise levels without a consistent trend. Surprisingly, the strongest Diffusion Policy result is obtained using augmentation from policies trained on the proficient human dataset, despite Diffusion Policy performing poorly when trained directly on that dataset.

Discussion The optimal curation strategy differs between architectures. BC-RNN benefits from consistent, high-quality action labels and improves with augmentation that exposes it to states outside the demonstration distribution. Diffusion Policy may benefit from broad state space coverage and be hurt by strategies that reduce dataset diversity, including quality filtering and single-policy augmentation. Average pairwise state distance correlates with Diffusion Policy success rate across augmentation conditions ($r = 0.76$, $p = 0.05$) but not with BC-RNN performance ($r = -0.59$, $p = 0.12$), suggesting that coverage is a more informative dataset property than operator quality labels for generative policy architectures. Both models also exhibit high sensitivity to which specific trajectories are included in training, with selection and training variance large relative to performance differences between many filter conditions.

Conclusion Data curation for imitation learning must take into account the architecture. The properties that make a dataset effective for BC-RNN, namely consistency and clean action labels, are in tension with the properties that benefit Diffusion Policy, namely diversity and broad state space coverage. Synthetic augmentation via noisy policy rollouts is a practical and low-cost strategy for improving performance, but must be designed for the target architecture.

Architecture-Dependent Effects of Data Curation in Robot Manipulation Imitation Learning

Kailana Baker-Matsuoka
Department of Electrical Engineering
Stanford University
kailana@stanford.edu

Abstract

Data curation is a critical component of imitation learning for robot manipulation. In this work, we conduct a systematic study of data curation strategies applied to a mixed-quality multi-human demonstration dataset, comparing heuristic trajectory-level filters and synthetic data augmentation across two policy architectures: BC-RNN and Diffusion Policy. We find that the optimal curation strategy is model-dependent. Our analysis demonstrates that BC-RNN benefits from high-quality, consistent demonstrations and improves substantially from synthetic augmentation that exposes it to states outside the training distribution, while Diffusion Policy benefits from broad state space coverage and is more sensitive to reductions in dataset diversity. Through analysis of state space coverage, we find evidence that average pairwise state distance is more predictive of Diffusion Policy performance than operator quality labels in our experimental setting. We further find that both models are affected by which individual trajectories are included in the data-starved regime, suggesting that results from any single experiment should be interpreted carefully. Our synthetic augmentation approach, which collects noisy rollouts from trained policies using multiple seeds and varying noise levels, achieves up to 0.807 success rate for Diffusion Policy and 0.720 for BC-RNN on the Square nut assembly task, exceeding the unaugmented baselines.

1 Introduction

Data curation is often treated as a model-agnostic preprocessing step in robot imitation learning. Given a mixed-quality demonstration dataset, a common assumption is that selecting demonstrations from higher proficiency operators should improve downstream performance regardless of the policy architecture. In this work, the same curation strategy can improve BC-RNN while degrading Diffusion Policy, indicating that the effectiveness of data curation depends on the learning architecture.

We conduct a systematic study of data curation strategies for robot manipulation using the robomimic framework (Mandlekar et al., 2021) on the Square nut assembly task. We evaluate two complementary approaches: heuristic data filtering and synthetic data augmentation via noisy policy rollouts. We compare two policy architectures: Behavioral Cloning with a recurrent policy (BC-RNN) and Diffusion Policy (Chi et al., 2024). BC-RNN appears to benefit from high-quality demonstrations and improves further with synthetic augmentation that exposes it to diverse out-of-distribution states, while Diffusion Policy benefits from broad state space coverage and is often hurt by quality filtering that reduces diversity. As a result, curation strategies that help one architecture often hurt the other. We further find that both models suffer from significant overfitting in the data-starved regime, leading to high sensitivity to which specific trajectories are included. Through analysis of state space coverage, we find that average pairwise state distance is correlated with Diffusion Policy performance, while operator quality labels show no statistically significant relationship. We also see that diverse

multi-policy augmentation with sufficient noise can match or exceed the performance of the original human demonstration dataset.

Our main contributions are:

- A systematic comparison of heuristic data filters across BC-RNN and Diffusion Policy, revealing a model-dependent interaction between data quality, diversity, and performance.
- Evidence that state space coverage may be more informative of Diffusion Policy performance than operator-quality, suggesting that coverage may be a more informative dataset property for generative policy architectures.
- A post-hoc synthetic augmentation strategy using noisy policy rollouts that substantially improves BC-RNN performance, and with multi-policy diversity and sufficient noise, also matches or exceeds the full human demonstration baseline for Diffusion Policy.
- An empirical analysis of training variance in data-starved regimes, showing that results are sensitive to both trajectory selection and training stochasticity, highlighting the difficulty of drawing definitive conclusions from small demonstration datasets.

2 Related Work

Mandlekar et al. (2021) introduced robomimic as a large-scale empirical study of imitation learning and offline RL methods for robot manipulation, establishing BC-RNN with a Gaussian mixture model output head as a strong baseline on suboptimal human data. A key finding was that operator quality has a large effect on downstream performance, with performance degrading substantially on lower-quality demonstrations. Our work builds directly on this benchmark, using BC-RNN as a primary baseline.

Chi et al. (2024) introduced Diffusion Policy, which models the action distribution as a denoising diffusion process and has emerged as one of the strongest-performing methods on dexterous manipulation benchmarks. Its ability to represent multimodal action distributions suggests it may be particularly well-suited to multi-human datasets. However, these methods do not explicitly address dataset quality, leaving it unclear whether architectural improvements can compensate for suboptimal data or whether different architectures respond differently to data curation.

Belkhale et al. (2023) formally studied data quality in imitation learning through the lens of distribution shift, showing that the same algorithm can have substantially different performance depending on dataset composition. Work on data scaling laws in robot manipulation found that diversity of environments and objects matters far more than raw demonstration count, and that adding redundant demonstrations beyond a threshold yields diminishing returns (Hu et al., 2025). These works diagnose the importance of data quality and diversity but do not provide practical, low-cost curation strategies, nor do they study how quality effects interact with model architecture.

A closely related line of work addresses distribution shift in imitation learning through data collection strategies that expose the policy to diverse states. DAgger (Ross et al., 2011) proposes iteratively collecting corrective demonstrations by having a human expert provide correct actions for states visited by the learned policy. DART (Laskey et al., 2017) takes a complementary approach by injecting noise into human demonstrations during data collection, exposing the policy to off-nominal states during training without requiring iterative online correction.

In contrast to these approaches, our method extends the synthetic augmentation intuition to a fully post-hoc setting requiring no human involvement after an initial policy is trained. Unlike DAgger and DART, which rely on human experts during data collection, we generate augmented demonstrations entirely from trained policy rollouts. This study systematically evaluates trajectory-level heuristics alongside synthetic augmentation within a controlled benchmark, explicitly comparing effects across model architectures to understand how dataset properties such as state space coverage relate to downstream policy performance.

3 Method

3.1 Heuristic Data Filtering

We evaluate a set of trajectory-level heuristic filters that select a subset of demonstrations from the full MH dataset based on properties of the trajectories, without requiring ground-truth quality labels.

These filters are motivated by practical settings where operator quality labels are unavailable, and are designed to be low-cost and easy to compute from the demonstration data alone. We chose 100 demonstrations because it corresponds to the size of a single operator-quality subset and creates a sufficiently data-constrained regime where filtering effects are observable.

Length-based filtering ranks demonstrations by total trajectory length in timesteps, selecting either the shortest or longest demonstrations. The shortest 100 demonstrations contain 73% better-operator demos, while the longest 100 contain 69% worse-operator demos, indicating that trajectory length is correlated with operator quality in this dataset.

Action smoothness filtering ranks demonstrations according to how smoothly actions change throughout a trajectory. We evaluate both a naive smoothness metric computed over the full action vector and component-wise variants that separately consider position, orientation, and gripper behavior. Because these components have different physical meanings and scales, we additionally evaluate a ranked-combination filter that aggregates the component-wise metrics without requiring manual weighting.

Additional details of the smoothness metrics and ranked-combination procedure are provided in Appendix B.

3.2 Baselines and Variance Analysis

We establish two primary baselines. The **full dataset baseline** trains on all 300 MH demonstrations without any filtering or augmentation, representing the upper bound of what is achievable with the available human data. The **random-100 baseline** trains on 100 randomly sampled demonstrations, providing a size-matched reference for evaluating whether filtered subsets perform better than naive subsampling.

A key methodological concern in small-dataset settings is whether observed performance differences between filter conditions reflect genuine signal or are artifacts of which specific trajectories happen to be included. To obtain a preliminary estimate of this, we conduct a variance decomposition study with three components. First, we fix the trajectory selection seed and vary the training seed across three runs to measure *training variance*. Second, we fix the training seed and vary the trajectory selection seed across three independent random subsets to measure *selection variance*. Third, we repeat the exact same run with identical trajectory selection and training seed to measure *irreducible stochasticity* from non-determinism in GPU computation and data loading. This decomposition allows us to assess how much of the variance observed across filter conditions is attributable to the filter criterion itself versus noise inherent to the small-data training regime.

3.3 Synthetic Data Augmentation

Our augmentation procedure generates synthetic demonstrations by rolling out a policy trained on the full MH dataset with Gaussian noise added to the executed actions. The key approach is to decouple state diversity from action label quality by adding noise to executed actions to reach diverse states while recording the clean policy action as the training label.

At each timestep t, the policy observes the current state and produces a clean action a_t^{clean}. A noisy action is then computed as:

$$a_t^{\text{noisy}} = \text{clip}(a_t^{\text{clean}} + \epsilon_t, -1, 1), \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2 I) \tag{1}$$

The environment is stepped with a_t^{noisy} to encourage placing the robot in a slightly out-of-distribution state. Then a_t^{clean} is saved as the training label. Noise is applied only to the 6 continuous action dimensions (end-effector position and orientation deltas) and not to the gripper, which remains binary. Only successful rollouts are retained in the final dataset.

We study two variations in the augmentation procedure. First, we vary the noise level $\sigma \in \{0.02, 0.05, 0.08, 0.10\}$ to control the degree of state perturbation. Higher noise increases state space coverage but reduces the rollout success rate, creating a practical tradeoff. Second, we vary the number and source of policies used for rollout collection. Using multiple independently trained policy seeds introduces diversity in seen actions beyond what action noise alone can provide, since different seeds may learn different strategies for solving the task. We also explore using policies trained on the proficient human (PH) dataset as an additional source of policy diversity. We refer

to datasets collected from a single policy seed as *single-policy* augmentation and from multiple seeds as *multi-policy* augmentation. All augmented datasets are combined with the original 300 MH demonstrations to form training sets of 600 total demonstrations.

3.4 State Space Coverage Analysis

To characterize dataset diversity and its relationship to policy performance, we quantify state coverage using two statistics computed over a fixed set of 1000 sampled states per dataset. Average pairwise distance measures the mean Euclidean separation between randomly sampled states:

$$D_{\text{pair}} = \frac{1}{|\mathcal{S}|(|\mathcal{S}|-1)} \sum_{i \neq j} \|s_i - s_j\|_2 \tag{2}$$

This metric captures the overall spread of the dataset in state space. State variance measures the average squared deviation of states from their empirical mean:

$$\text{Var}(S) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \|s - \mu_S\|_2^2 \tag{3}$$

Pairwise distance measures the typical separation between states and captures the overall spread of the dataset. State variance measures dispersion relative to a central mean state. Together, these metrics provide complementary views of dataset coverage while remaining simple to compute directly from low-dimensional observations. The state vectors consist of end-effector position and object state variables capturing the task-relevant configuration of the robot and environment. To ensure fair comparisons across datasets of different sizes, all metrics are computed on fixed-size samples. State dimensions are not normalized prior to computing distances, meaning dimensions with larger absolute variance contribute more to the metric. Since all datasets are drawn from the same task and observation space, we apply the metric consistently across conditions and use it only for relative comparisons rather than as an absolute measure of coverage.

4 Experimental Setup

4.1 Task and Dataset

We evaluate all methods on the Square task from the robomimic benchmark (Mandlekar et al., 2021), in which a robot arm must pick up a square nut and place it onto a peg. We use the multi-human (MH) demonstration dataset, which contains 300 demonstrations collected by six operators with two of each level of operator proficiency, labeled worse, okay, and better, with 50 demonstrations per operator. We additionally use the proficient human (PH) dataset for the Square task, which contains 200 demonstrations from a single skilled operator. The PH dataset is used both as a baseline and as a source of rollout policies for synthetic data augmentation experiments. We chose the Square task because it is sufficiently challenging that data quality effects are clearly visible. Simpler tasks such as the Can task reach near-perfect success rates, making it difficult to distinguish the effects of different filtering approaches. All experiments use low-dimensional state observations consisting of the robot end-effector configuration, gripper state, and object pose.

4.2 Policy Architectures

We evaluate two policy architectures. BC-RNN uses a recurrent neural network with a Gaussian Mixture Model action head. Diffusion Policy (Chi et al., 2024) models the action distribution as a denoising diffusion process over a prediction horizon using a 1D temporal convolutional UNet. Both use default hyperparameters from the robomimic paper configs and are trained for 2000 epochs.

4.3 Evaluation Protocol

We evaluate each method by rolling out the best-checkpoint policy for 50 episodes and reporting task success rate. We report mean and standard deviation across 3 random training seeds. The best checkpoint is selected based on rollout success rate evaluated every 50 epochs. Unless otherwise noted, all filtered subsets contain 100 demonstrations to enable fair comparison.

5 Results

5.1 Baseline Performance

We first establish baseline performance across operator quality splits to study how data quality affects different policy architectures. Table 1 reports results for both architectures trained on the full MH dataset, a random subset of 100 MH demonstrations for size comparison, and each operator quality split.

Condition	BC-RNN	Diffusion
Full MH (300)	0.433 ± 0.062	$\mathbf{0.753 \pm 0.025}$
Full PH (200)	$\mathbf{0.707 \pm 0.019}$	0.440 ± 0.049
Better-only (100)	0.407 ± 0.062	0.547 ± 0.034
Okay-only (100)	0.320 ± 0.016	0.513 ± 0.009
Worse-only (100)	0.367 ± 0.034	0.387 ± 0.057
Random-100	0.253 ± 0.038	0.607 ± 0.019

Table 1: Baseline success rates on the Square task, averaged over 3 training seeds. MH denotes the multi-human dataset with 300 demonstrations from operators of varying skill. PH denotes the proficient human dataset with 200 demonstrations from a single skilled operator.

BC-RNN trained on better-only demonstrations achieves performance comparable to training on the full dataset (0.407 ± 0.062 vs. 0.433 ± 0.062), suggesting that BC-RNN may benefit more from demonstration quality than from additional demonstrations in this setting. In contrast, Diffusion Policy trained on the full dataset outperforms the better-only subset (0.753 ± 0.025 vs. 0.547 ± 0.034), indicating that it benefits from information present in the full dataset that is lost when only better-operator demonstrations are used. Notably, the random subsampling of 100 demonstrations (0.607 ± 0.019) outperforms selecting only better-quality demonstrations for Diffusion Policy, suggesting that diversity of demonstrations is more important than quality for this architecture. Additionally, BC-RNN trained on the PH dataset (0.707 ± 0.019) substantially outperforms full MH (0.433 ± 0.062) despite containing demonstrations from a single operator, while Diffusion Policy trained on PH (0.440 ± 0.049) performs far below its full MH baseline (0.753 ± 0.025). This illustrates the model-dependent nature of data diversity: BC-RNN benefits from the quality and consistency of a single skilled operator, while Diffusion Policy requires the broader policy diversity provided by multiple operators to reach its best performance.

5.2 Heuristic Filtering

We evaluate a range of heuristic filters applied to the MH dataset, selecting 100 demonstrations based on trajectory-level properties without access to ground-truth quality labels. We selected the best-performing and most informative BC-RNN filters to evaluate on Diffusion Policy.

Dual-Architecture Conditions				BC-RNN-Only Filtering Conditions (N=100)	
Condition / Filter (100)	**N**	**BC-RNN**	**Diffusion**	**Filter Strategy**	**BC-RNN Success**
Full MH Baseline	300	0.433 ± 0.062	$\mathbf{0.753 \pm 0.025}$	Shortest Trajectories	0.380 ± 0.071
Full PH Baseline	200	$\mathbf{0.707 \pm 0.019}$	0.440 ± 0.049	Longest Trajectories	0.327 ± 0.066
Random-100 Reference	100	0.253 ± 0.038	0.607 ± 0.019	Smoothest Position	0.387 ± 0.116
Better-only Cut	100	0.407 ± 0.062	0.547 ± 0.034	Least Smooth Position	0.300 ± 0.082
Okay-only Cut	100	0.320 ± 0.016	0.513 ± 0.009	Smoothest Orientation	0.240 ± 0.028
Worse-only Cut	100	0.367 ± 0.034	0.387 ± 0.057	Least Smooth Orien.	0.293 ± 0.066
Smoothest (All Dims)	100	0.460 ± 0.075	0.687 ± 0.050	Fewest Gripper Changes	0.287 ± 0.109
Most Gripper Changes	100	0.467 ± 0.041	0.633 ± 0.074	Least Smooth (Ranked)	0.307 ± 0.075
Smoothest (Ranked)	100	0.380 ± 0.086	0.587 ± 0.019	Least Smooth (All Dims)	0.380 ± 0.033

Table 2: Task success rates across full benchmarks, size baselines, and granular heuristic pruning constraints.

For BC-RNN, heuristic filtering provides little evidence that removing demonstrations improves performance. Although a small number of filters slightly outperform the full MH dataset, the observed

gains are comparable to the training and selection variance measured in Section 5.3, making it difficult to determine whether these improvements reflect meaningful signal. Most filters perform above the random-100 baseline but below the full MH dataset baseline, suggesting that filtering can recover some performance lost from reducing dataset size but provides limited benefit beyond simply using the full dataset in this regime.

For Diffusion Policy, all evaluated heuristic filters perform below the full dataset baseline. This suggests that for Diffusion Policy, reducing dataset size through filtering is generally detrimental in our setting, and that dataset diversity may be more important than demonstration quality labels alone.

To better understand what factors contribute to filter performance, we analyze the operator quality composition and total training timesteps of each filtered subset. We find no statistically significant correlation between these metrics and policy performance. Operator diversity, measured as the entropy of the distribution across quality tiers, shows near-zero correlation with BC-RNN performance ($r = -0.07$, $p = 0.86$), and total training timesteps show only a weak positive trend ($r = 0.23$, $p = 0.55$), as shown in Figure 1 and Figure 2.

Total training timesteps represent a potential confounding variable: while all filter conditions select exactly 100 demonstrations, trajectories vary substantially in length, resulting in timestep counts ranging from 16,824 (shortest filter) to 39,471 (longest filter), meaning some filters expose the model to more training examples and gradient updates per epoch than others. However, the weak and insignificant correlation suggests that timestep count alone does not explain performance differences.

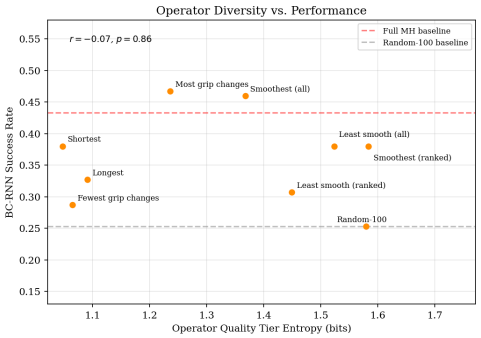

Figure 1: BC-RNN success rate vs. operator quality tier entropy ($r = -0.07$, $p = 0.86$).

Figure 2: BC-RNN success rate vs. total training timesteps ($r = 0.23$, $p = 0.55$).

Interestingly, filter performance is not explained by operator quality composition alone. The most-gripper-changes filter contains predominantly worse-operator demonstrations yet achieves the strongest BC-RNN performance, while the fewest-gripper-changes filter contains almost exclusively better and okay demonstrations but performs poorly. This suggests that trajectory-level characteristics may capture aspects of demonstration usefulness that are not reflected by operator quality labels alone. Full operator composition details are provided in the Appendix A.

5.3 Sensitivity to Trajectory Selection

A key concern in small-dataset imitation learning is whether observed performance differences reflect genuine signal from the filter criterion or simply the particular trajectories that happen to be selected. We conduct a variance decomposition study to quantify three sources of variation, all using BC-RNN trained on 100 randomly selected demonstrations.

Training the same 100 randomly selected demonstrations with three different training seeds yields results of 0.200, 0.280, and 0.280, a spread of 0.080. Varying which 100 demonstrations are selected while fixing the training seed produces results of 0.200, 0.280, and 0.320, a spread of 0.120. Repeating the exact same run three times with identical settings produces identical results (0.200 each), suggesting that training was effectively deterministic under the hardware and software conditions used in these experiments. Many filter conditions differ by less than 0.05 success rate, which is comparable to the measured training and selection variance, making results from individual

Variance Source	Run 1	Run 2	Run 3	Mean	Std
Training variance (fixed selection, varied training seed)	0.200	0.280	0.280	0.253	0.038
Selection variance (fixed training seed, varied selection)	0.200	0.280	0.320	0.267	0.049
Irreducible stochasticity (identical runs repeated)	0.200	0.200	0.200	0.200	0.000

Table 3: Variance decomposition for BC-RNN trained on random-100. Training variance fixes the trajectory selection and varies the training seed. Selection variance fixes the training seed and varies which 100 demonstrations are selected. Irreducible stochasticity repeats the exact same run with identical selection and training seed.

runs difficult to interpret. We therefore report means across multiple seeds throughout this work. Due to computational constraints, these variance estimates are based on only three runs per condition and should be interpreted as approximate. We note that the random-100 seed 0 run locally produced 0.340, compared to 0.200 when run on cloud hardware, a difference of 0.140 attributable to hardware and software environment differences.

5.4 Single-Policy Augmentation

The filtering experiments suggest that removing demonstrations is unlikely to be the most effective way to improve performance. Both architectures exhibit clear signs of overfitting when trained on 100-demonstration subsets, as shown in Figure 3. This indicates that the models are data-limited rather than being primarily harmed by low-quality demonstrations. This combined with the variance analysis and the strong performance of Diffusion Policy on the full MH dataset motivate exploring whether adding data through synthetic augmentation is more effective than removing demonstrations through filtering.

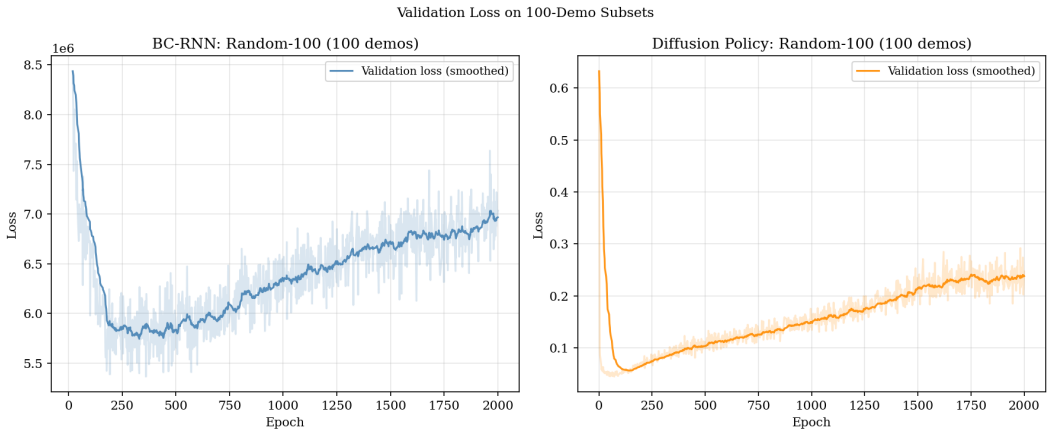

Figure 3: Validation loss curves for BC-RNN (left) and Diffusion Policy (right) trained on random-100. Both architectures show clear overfitting: validation loss decreases initially but then increases steadily throughout training, indicating that 100 demonstrations are insufficient to prevent overfitting in this data-limited regime.

We generate synthetic demonstrations by rolling out a trained Diffusion Policy (trained on full MH, seed 1) with Gaussian noise $\sigma = 0.02$ injected into executed actions, collecting 300 successful rollouts. We evaluate training on synthetic data alone (100 randomly sampled synthetic rollouts) and combined with the original MH dataset (600 demos total).

Synthetic random-100 outperforms random-100 (human) for BC-RNN (0.307 vs 0.253), suggesting that BC-RNN benefits from the consistent action labels provided by a trained policy rollout. This is consistent with the broader trend that BC-RNN benefits from consistent action labels across several experiments. For Diffusion Policy the opposite is true: synthetic random-100 performs worse than random-100 (human) (0.487 vs 0.607), suggesting that Diffusion Policy benefits more from the diversity present in multi-human demonstrations than from the consistency of single-policy rollouts. This is supported by the baseline results, where BC-RNN trained on the single-operator PH

Condition	N	BC-RNN	Diffusion
Full MH	300	0.433 ± 0.062	$\mathbf{0.753 \pm 0.025}$
Random-100	100	0.253 ± 0.038	0.607 ± 0.019
Synthetic random-100	100	0.307 ± 0.034	0.487 ± 0.025
MH + Synthetic (600)	600	$\mathbf{0.693 \pm 0.052}$	0.647 ± 0.019

Table 4: Single-policy augmentation results. Synthetic rollouts are collected from a Diffusion Policy trained on full MH with noise $\sigma = 0.02$. Random-100 is included as a size-matched baseline using randomly selected human demonstrations.

dataset (0.707 ± 0.019) exceeds full MH performance, while Diffusion Policy performs worse on PH (0.440 ± 0.049) than on the more diverse MH dataset (0.753 ± 0.025), suggesting that consistency of the action source matters more than operator diversity for BC-RNN, while the opposite holds for Diffusion Policy.

When combining all 300 synthetic rollouts with the original 300 MH demonstrations, BC-RNN achieves a large improvement ($0.433 \rightarrow 0.693$, a 60% relative improvement), while Diffusion Policy performance drops below its full MH baseline ($0.753 \rightarrow 0.647$). To understand this asymmetry, we analyze the state space coverage of each dataset using two metrics: average pairwise distance and state variance, both computed on a fixed sample of 1000 states for size-independent comparison.

Dataset	N Demos	Pairwise Distance	State Variance
MH original (300)	300	1.637	0.186
Random-100 (human)	100	1.640	0.185
Synthetic random-100	300	1.520	0.182
MH + Synthetic (600)	600	1.571	0.184

Table 5: State space coverage of each dataset condition. Pairwise distance is computed on a fixed sample of 1000 states averaged over 5 trials for size-independent comparison.

The synthetic rollouts have substantially lower state space coverage than the original MH data (pairwise distance 1.520 vs 1.637), and combining them with the MH dataset reduces the overall coverage relative to MH alone (1.571 vs 1.637). All rollouts originate from a single policy following similar behavioral strategies, whereas human demonstrations reflect diverse approaches and skill levels across operators. This reduction in state coverage may help explain why Diffusion Policy is hurt by single-policy augmentation: it relies on broad state space coverage to learn a good generative model of the action distribution, and the synthetic rollouts dilute the diversity of the training set.

On the other hand, BC-RNN benefits from this additional training signal regardless of the lower state coverage. This suggests that the two architectures exploit training data in different ways, motivating an investigation into whether increasing the diversity of the synthetic rollouts through multiple policy seeds and higher noise levels can improve Diffusion Policy performance.

5.5 Multi-Policy Augmentation

The state coverage analysis motivates two strategies for improving augmentation quality: using multiple independently trained policy seeds to increase diversity of seen actions, and increasing the noise level to force the robot into more diverse states. We evaluate both strategies and also explore using policies trained on the proficient human (PH) dataset as an alternative source of policy diversity.

5.5.1 Effect of Policy Diversity

We first compare single-policy augmentation to multi-policy augmentation at the same noise level ($\sigma = 0.02$), holding all other factors fixed.

Multi-policy augmentation substantially improves Diffusion Policy performance ($0.647 \rightarrow 0.740$), nearly recovering its full MH baseline of 0.753. This improvement is accompanied by an increase in state space coverage (pairwise distance $1.571 \rightarrow 1.597$), consistent with the hypothesis that state coverage contributes to Diffusion Policy performance. For BC-RNN, multi-policy augmenta-

Condition	N	BC-RNN	Diffusion
Full MH	300	0.433 ± 0.062	0.753 ± 0.025
MH + single-policy std=0.02	600	0.693 ± 0.052	0.647 ± 0.019
MH + multi-policy std=0.02	600	0.660 ± 0.071	0.740 ± 0.028

Table 6: Effect of policy diversity on augmentation performance. Multi-policy uses 100 rollouts from each of 3 independently trained policy seeds at the same noise level.

tion slightly reduces performance ($0.693 \rightarrow 0.660$), suggesting BC-RNN benefits more from the consistent behavior of a single policy than from increased diversity.

5.5.2 Effect of Noise Level

We next vary the noise level σ while using multi-policy augmentation from 3 MH-trained policy seeds. Higher noise levels increase state space coverage but reduce the success rate of collected rollouts, with success rates dropping from approximately 79% at $\sigma = 0.02$ to 54% at $\sigma = 0.10$. Beyond $\sigma = 0.10$, success rates drop below 50%, making reliable data collection impractical.

Noise Level	BC-RNN	Diffusion	Pairwise Dist.	State Var.
MH original	0.433 ± 0.062	0.753 ± 0.025	1.637	0.186
$\sigma = 0.02$	0.660 ± 0.071	0.740 ± 0.028	1.597	0.184
$\sigma = 0.05$	0.587 ± 0.066	0.753 ± 0.019	1.604	0.185
$\sigma = 0.08$	0.707 ± 0.019	—	1.591	0.185
$\sigma = 0.10$	0.720 ± 0.043	0.773 ± 0.050	1.625	0.186

Table 7: Effect of noise level on multi-policy augmentation performance and state space coverage. Pairwise distance computed on a fixed sample of 1000 states averaged over 5 trials. Diffusion Policy was not evaluated at $\sigma = 0.08$ due to time constraints.

For Diffusion Policy, performance generally increases with noise level across the evaluated conditions ($\sigma \in \{0.02, 0.05, 0.10\}$), with $\sigma = 0.10$ achieving 0.773 ± 0.050, exceeding the full MH baseline. This trend is broadly consistent with the state coverage values, which also generally increase with noise level. For BC-RNN, performance improves substantially over the unaugmented baseline across all noise levels but does not show a consistent trend with noise level, suggesting BC-RNN benefits primarily from exposure to states outside the human demonstration distribution rather than from increased policy diversity.

5.5.3 Alternative Policy Sources

Finally, we explore whether using policies trained on a qualitatively different dataset as rollout sources can produce different rollout distributions. We collect rollouts from policies trained on the proficient human (PH) dataset, which contains demonstrations from a single skilled operator, in contrast to the multi-operator MH dataset used for all previous augmentation experiments. We evaluate two conditions: rollouts from 3 PH-trained policy seeds only (100 rollouts each), and a mixed condition using 50 rollouts from each of 3 MH-trained and 3 PH-trained seeds, both at $\sigma = 0.05$.

Condition	N	BC-RNN	Diffusion
Full MH	300	0.433 ± 0.062	0.753 ± 0.025
MH + MH multipolicy std=0.10	600	$\mathbf{0.720 \pm 0.043}$	0.773 ± 0.050
MH + PH multipolicy std=0.05	600	0.667 ± 0.052	$\mathbf{0.807 \pm 0.090}$
MH + mixed MH+PH multipolicy std=0.05	600	0.533 ± 0.034	0.760 ± 0.057

Table 8: Effect of rollout policy source on augmentation performance. PH multipolicy uses 100 rollouts from each of 3 PH-trained policy seeds. Mixed MH+PH multipolicy uses 50 rollouts from each of 3 MH-trained and 3 PH-trained seeds, all at $\sigma = 0.05$.

Using PH-trained policies as rollout sources yields the strongest Diffusion Policy result in this study (0.807 ± 0.090), substantially exceeding the full MH baseline and all MH-policy augmentation

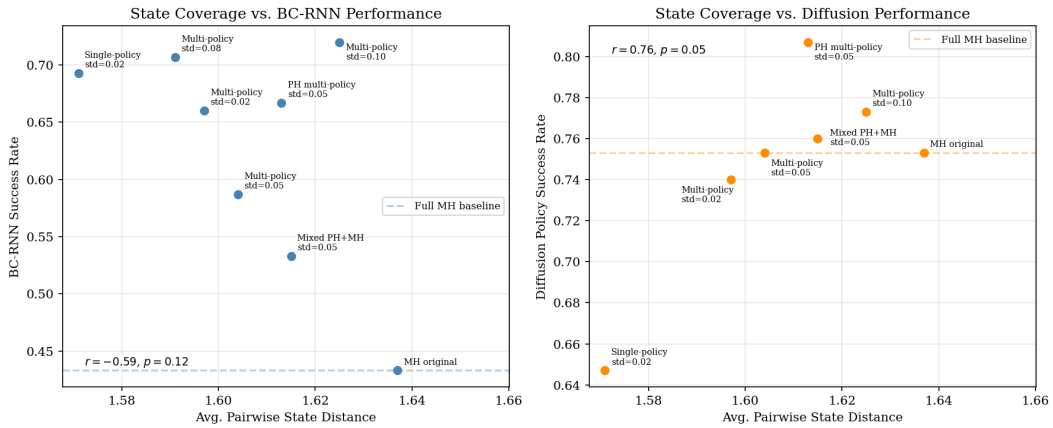

Figure 4: Average pairwise state distance vs. success rate for all augmentation conditions. Left: BC-RNN. Right: Diffusion Policy. Each point represents one augmentation condition. The dashed line shows the full MH baseline for each architecture.

conditions. One possible explanation is that PH policies visit regions of state space that are underrepresented in MH-policy rollouts. This may provide complementary policy diversity that Diffusion Policy can exploit. The mixed MH+PH condition (0.760) also exceeds the full MH baseline but underperforms pure PH rollouts, suggesting that diluting the PH rollouts with MH rollouts reduces the marginal diversity added to the already MH-heavy training set.

For BC-RNN, the pattern reverses. PH multipolicy augmentation (0.667) underperforms the best MH multipolicy result (0.720), and the mixed condition performs poorly (0.533), falling well below the full MH baseline. This suggests that BC-RNN is sensitive to the consistency of the action label distribution. When rollout sources are mixed from different training sets, the resulting behavioral inconsistency makes it harder for BC-RNN to generalize. Diffusion Policy, by contrast, exploits this diversity, further reinforcing the model-dependent nature of augmentation design.

6 Discussion

The central trend in the results is that BC-RNN and Diffusion Policy respond to the same data in different ways. On the full MH dataset, Diffusion Policy substantially outperforms BC-RNN, while on the PH dataset this is reversed. The two architectures appear to exploit training data through different mechanisms, and understanding this difference is key to designing effective curation strategies.

One possible explanation is that BC-RNN's direct state-to-action mapping makes it more sensitive to action-label consistency. Noisy demonstrations provide inconsistent supervision signals that are difficult for the model to learn from. This may explain why BC-RNN benefits from several augmentation conditions despite differences in dataset diversity. The primary benefit of augmentation for BC-RNN appears to be exposure to states outside the human demonstration distribution.

Diffusion Policy, by contrast, learns a generative model of the action distribution conditioned on state. Accurately modeling this distribution requires broad coverage of the state-action space, which means that any curation strategy reducing dataset diversity is likely to hurt performance. This trend is seen across the experiments: quality filtering and single-policy augmentation can dilute existing diversity and reduce Diffusion Policy performance relative to the full MH baseline. The correlation between average pairwise state distance and Diffusion Policy success rate ($r = 0.76$, $p = 0.05$) across augmentation conditions supports this interpretation, suggesting that state coverage may be a more informative dataset property than operator quality labels for this architecture. This coverage-performance relationship does not hold for BC-RNN ($r = -0.59$, $p = 0.12$), consistent with the view that coverage matters for distribution learning but not for direct regression.

The alternative policy source experiments reinforce this picture. Multi-policy augmentation with PH-trained policies achieves the strongest Diffusion Policy result in this study (0.807). One possible explanation is that PH-trained policies follow behavioral strategies that differ from those learned

from the MH dataset, producing trajectories that provide complementary state coverage. However, this improvement could also arise from differences in action-label quality or other characteristics of the rollout distribution. For BC-RNN the same condition may perform poorly because mixing action labels from policies trained on different distributions introduces inconsistency that a direct regression model struggles to learn from.

A key limitation is that both models exhibit high sensitivity to which specific trajectories are included in training, particularly in the data-starved regime. Training and selection variance are both large relative to the performance differences between many filter conditions, which means that results from individual experiments should be interpreted carefully. The performance of a curation strategy may partly reflect which trajectories happened to be selected rather than the strategy itself, and evaluating across multiple seeds is essential for drawing reliable conclusions.

Together, these findings suggest that data curation for imitation learning must be architecture specific. The properties that make a dataset good for BC-RNN, namely consistency, quality, and clean action labels, are in tension with the properties that make a dataset good for Diffusion Policy, namely diversity, coverage, and broad behavioral variation. In practice, for the best performance, data curation and model selection must be considered together. These results caution against evaluating data curation strategies using a single policy architecture, since conclusions about dataset quality may not transfer across learning algorithms.

Limitations This study is limited to a single simulated task using low-dimensional observations, and results may not generalize to image-based policies, real robot settings, or more complex manipulation tasks. The synthetic augmentation approach also has practical limitations: generating rollouts requires a simulation environment and an initial policy of sufficient quality to produce useful demonstrations. The high variance observed in small-dataset settings makes it difficult to draw definitive conclusions from individual experiments, and a more thorough statistical analysis with more seeds would strengthen the findings. Additionally, several augmentation factors vary simultaneously across experiments, including dataset size, state coverage, and rollout source. Future work should isolate these factors through controlled experiments.

Future Work Future work could explore whether these findings generalize to larger demonstration datasets, where quality filtering may become more beneficial as the ratio of low-quality to high-quality demonstrations increases and the data-starved regime no longer dominates. Evaluating these curation strategies on image-based policies and real robot tasks would also be valuable, as would investigating whether state space coverage metrics can be used proactively to guide data collection rather than as a post-hoc analysis tool. A useful control experiment would compare synthetic augmentation against simply duplicating or resampling existing demonstrations to determine whether gains arise from increased dataset size, increased state coverage, or both.

7 Conclusion

Our results show that the optimal data curation strategy is dependent on the target architecture. For discriminative models like BC-RNN, maximizing consistency is important while for generative models like Diffusion Policy, dataset diversity and state coverage contribute to performance. Additionally, state space coverage is more strongly associated with Diffusion Policy performance than demonstration quality labels, and synthetic augmentation via noisy policy rollouts can match or exceed the performance of the original human demonstration dataset when designed to maximize policy diversity.

These findings have practical implications for robot learning practitioners. When working with a mixed-quality demonstration dataset, the appropriate curation strategy should be chosen based on the target architecture. For BC-RNN-style discriminative models, modest quality filtering and synthetic augmentation are effective. For generative models like Diffusion Policy, preserving and expanding dataset diversity matters more than removing low-quality demonstrations. More broadly, our results suggest that effective data curation depends not only on the dataset itself, but also on the way a policy architecture extracts information from that dataset.

8 Team Contributions

As the sole member of this project, I conducted all experiments, analysis, and wrote the paper independently.

Changes from Proposal The project was completed individually as proposed. The primary change from the original proposal was an expanded focus on synthetic data augmentation and state space coverage analysis, which emerged as a key contribution based on early experimental findings showing that heuristic filtering alone provided limited benefit.

AI Tools Disclosure Claude (Anthropic) was used to assist with dataset processing scripts, figure generation code, and cloud training pipeline setup via Modal.

References

Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. 2023. Data Quality in Imitation Learning. arXiv:2306.02437 [cs.RO] https://arxiv.org/abs/2306.02437

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. 2024. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. arXiv:2303.04137 [cs.RO] https://arxiv.org/abs/2303.04137

Yingdong Hu, Fanqi Lin, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. 2025. Data Scaling Laws in Imitation Learning for Robotic Manipulation. arXiv:2410.18647 [cs.RO] https://arxiv.org/abs/2410.18647

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. 2017. DART: Noise Injection for Robust Imitation Learning. arXiv:1703.09327 [cs.LG] https://arxiv.org/abs/1703.09327

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. 2021. What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. In *5th Annual Conference on Robot Learning*. https://openreview.net/forum?id=JrsfBJtDFdI

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. 2011. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. arXiv:1011.0686 [cs.LG] https://arxiv.org/abs/1011.0686

A Heuristic Filter Operator Composition

Table 9 reports the full operator composition of each heuristic filter, showing how many demonstrations are selected from each operator quality tier.

B Action Smoothness Metrics

The robomimic action space consists of a 7-dimensional action vector containing end-effector position deltas (dimensions 0–2), orientation deltas represented as axis-angle rotations (dimensions 3–5), and a binary gripper command (dimension 6).

For the naive smoothness metric, we compute the mean magnitude of action changes along a trajectory:

$$\text{smoothness}(a) = \frac{1}{T-1} \sum_{t=1}^{T-1} \|a_{t+1} - a_t\|_2. \tag{4}$$

Demonstrations with lower values are considered smoother, while larger values indicate more abrupt or jittery action sequences.

Filter	Worse	Okay	Better
Random-100	30	37	33
Shortest	6	21	73
Longest	69	26	5
Smoothest (all)	47	10	43
Least smooth (all)	24	29	47
Fewest grip changes	1	45	54
Most grip changes	65	10	25
Smoothest (ranked)	35	33	32
Least smooth (ranked)	37	15	48

Table 9: Operator quality tier composition of each heuristic filter condition. All filters select 100 demonstrations from the full MH dataset of 300. The table illustrates that filter performance is not explained solely by operator quality composition. For example, the most-gripper-changes filter contains predominantly worse-operator demonstrations (65%) yet achieves the strongest BC-RNN performance, while the fewest-gripper-changes filter contains almost exclusively okay and better demonstrations (99%) but performs substantially worse.

Because the position, orientation, and gripper components operate on different physical scales and units, we additionally evaluate component-wise smoothness metrics.

Position Smoothness Position smoothness is computed using the Euclidean distance between consecutive position action components:

$$s_{\text{pos}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \|a_{t+1}^{\text{pos}} - a_t^{\text{pos}}\|_2. \tag{5}$$

Orientation Smoothness Since orientation actions represent rotations, Euclidean distance is not geometrically meaningful. We therefore compute orientation smoothness using the geodesic distance between consecutive rotations:

$$d_{\text{ori}}(R_t, R_{t+1}) = \arccos\left(\frac{\text{tr}\left(R_t^\top R_{t+1}\right) - 1}{2}\right). \tag{6}$$

Orientation smoothness is the average of these distances across the trajectory.

Gripper Smoothness The gripper action is binary ($\{-1, +1\}$), making continuous smoothness measures inappropriate. Instead, we count the number of gripper state transitions:

$$s_{\text{grip}} = \sum_{t=1}^{T-1} \mathbf{1}[g_{t+1} \neq g_t]. \tag{7}$$

Fewer transitions indicate more stable gripper behavior.

Ranked Combination Filter To combine component-wise metrics without introducing scale-dependent weighting, each demonstration is ranked independently according to its position, orientation, and gripper smoothness scores. The ranks are then summed:

$$s_{\text{ranked}} = r_{\text{pos}} + r_{\text{ori}} + r_{\text{grip}}. \tag{8}$$

Demonstrations with the lowest summed ranks are selected as the smoothest combined filter, while those with the highest summed ranks form the least-smooth combined filter.