RoboDrop: Curating VLA Post-Training Data via Local Gradient Compatibility

Anonymous authors
Paper under double-blind review

Abstract

Vision–language–action (VLA) models acquire broad generalization through large-scale pretraining, yet adapting them to a new task and robot embodiment still requires post-training on newly collected data. Unlike pretraining, post-training targets task- and embodiment-specific adaptation, making it particularly sensitive to data quality. In practice, collected robot datasets often contain heterogeneous errors, including execution mistakes, sensor drift, and timestamp misalignment, which can impair post-training and policy performance. Manual inspection is costly, while existing data-cleaning methods are typically tailored to particular corruption types. To address these challenges, we introduce RoboDrop, a data-curation framework that audits supervision using local gradient compatibility measured along the training trajectory as a proxy for its effect on post-training performance. During a one-epoch warm-up run, RoboDrop scores each candidate sample online by comparing its gradient with those of task-semantic and visually matched validation samples. The resulting sample scores are aggregated at the episode level, and a simple automatic post-processing rule converts them into filtering decisions. We evaluate RoboDrop on controlled observation–action corruptions, naturally suboptimal demonstrations in simulation, and real-robot datasets containing non-expert collection errors. Across these settings, RoboDrop more accurately distinguishes unreliable demonstrations than prior methods, while post-training on the curated data consistently yields stronger downstream policies, with average real-robot rollout success rising from 35.0% to 67.5%. These results establish training-trajectory-aware, context-conditioned supervision auditing as an effective approach to robust VLA post-training.

1 Introduction

Large-scale vision–language–action (VLA) models have demonstrated broad generalization across diverse robotic tasks and environments (Black et al., 2024; 2025; NVIDIA et al., 2025; Liu et al., 2024; Brohan et al., 2023; Kim et al., 2024; Team et al., 2024). However, achieving reliable performance on a new task and robot embodiment still requires post-training on task-specific demonstrations collected in the target environment (Kim et al., 2025; O'Neill et al., 2024). Ensuring the quality of these post-training data is challenging, as real-world collection is susceptible to imperfect teleoperation, execution mistakes, sensor drift, and temporal misalignment between observations and actions (Li et al., 2026; 2025). These imperfections need not make an entire trajectory unsuccessful. Instead, they often induce locally unreliable observation–action supervision at particular task states, producing policy updates that conflict with the desired behavior and ultimately reduce downstream task success. Identifying such episodes by manual inspection is prohibitively expensive.

Existing robot data-selection methods address related but distinct challenges. Simple proxies, such as demonstration length or trajectory-level outcomes, efficiently identify clearly inefficient or failed trajectories, but these coarse signals miss localized sensing and temporal errors within otherwise successful ones. Retrieval-based approaches select data according to visual, motion, or behavioral similarity to a target task (Memmel et al., 2025; Du et al., 2023; Lin et al., 2024); similarity identifies relevant data, but does not determine whether the action paired with a particular observation provides reliable supervision. Performance-aware approaches, such as DataMIL (Dass et al., 2026), estimate the utility of data for improving a target policy and enable task-aware selection from large, heterogeneous data collections. In contrast, we focus on identifying locally unreliable supervision

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

within task-relevant post-training demonstrations. This calls for a fine-grained, context-conditioned assessment of each observation–action pair's contribution to post-training performance.

Gradient-based data attribution provides a useful perspective for this problem. Under a first-order approximation of validation-loss change, the contribution of a training example can be characterized by its gradient alignment with held-out validation behavior (Koh & Liang, 2017; Pruthi et al., 2020; Wang et al., 2025). However, robot manipulation data are highly heterogeneous: a single task can contain visually and behaviorally distinct reaching, grasping, transport, and placement stages, while visually similar observations can require different actions under different language instructions. Reliable validation behavior should therefore be defined locally, using samples that are compatible with both the task semantics and interaction state of the candidate. Moreover, for flow-matching (Lipman et al., 2022) VLAs, gradient estimates evolve throughout training and depend on stochastic noising variables, making a single static comparison insufficiently robust.

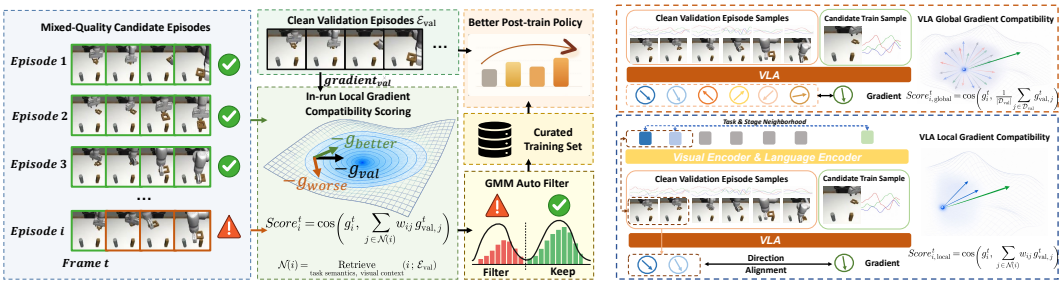
We introduce **RoboDrop**, a training-trajectory-aware framework for automated data curation based on *task-semantic and visual-context-conditioned gradient compatibility*. Under this local view, reliable supervision should induce an update compatible with the update supported by validation samples from the same task semantics and interaction state. During a one-epoch warm-up run, RoboDrop scores each candidate sample once, at the optimization step when it is encountered, against the most recently refreshed local validation reference. It then aggregates these online sample-level scores at the trajectory level to preserve the temporal integrity of demonstrations. In practice, the episode scores can be used with a standard fixed-ratio filtering rule. For settings where the contamination rate is unknown, we additionally provide an optional BIC-based post-processing procedure that adaptively converts the episode scores into filtering decisions.

We evaluate RoboDrop across three complementary sources of imperfect supervision: controlled synthetic corruptions that explicitly disrupt observation–action correspondence, naturally suboptimal demonstrations collected by less-skilled operators in simulation, and real-robot datasets containing execution mistakes from non-expert data collection. RoboDrop consistently provides stronger episode-level discrimination than prior methods across multiple VLA backbones, achieving a mean AUROC of 88.33% for distinguishing proficient from suboptimal demonstrations across four simulated manipulation tasks. On real-robot data, RoboDrop automatically determines the filtering budget without access to the contamination rate, and post-training on the curated data improves rollout success from 35.0% to 67.5%. Our contributions are as follows:

- We introduce a *context-conditioned local gradient reference* for auditing VLA post-training supervision. By constructing the reference from clean validation samples matched by task semantics and visual interaction context, it provides a candidate-specific measure of whether each sample induces a desirable policy update.
- We develop a *training-trajectory-aware online scoring procedure* that evaluates candidate samples as training progresses. It captures fine-grained variations in gradient compatibility during optimization, producing quality estimates that reflect evolving training dynamics.
- We validate RoboDrop on controlled supervision corruptions, naturally suboptimal simulated demonstrations, and real-robot collection errors, demonstrating improved filtering quality and downstream policy success across multiple VLA backbones.

2 Related Work

Robot Data Curation and Selection. Prior work has explored robot-data selection at multiple granularities (Hejna et al., 2024). Trajectory-level methods assess demonstration quality using mutual information (Hejna et al., 2025) or manually specified quality signals (Yuan et al., 2026), while retrieval-based approaches select data that are visually, behaviorally, or temporally related to a target task (Memmel et al., 2025; Du et al., 2023; Lin et al., 2024; Kumar et al., 2025). Rollout-based methods leverage policy execution feedback, either by learning success–failure classifiers to filter demonstrations (Chen et al., 2025) or by estimating their influence on closed-loop returns (Agia et al., 2025). Fine-grained curation methods further identify suboptimal transitions through task-progress estimation or remove redundant state–action patterns (Zhang et al., 2025). DataMIL estimates data utility for a target policy to select beneficial trajectories from heterogeneous prior datasets (Dass et al., 2026; Engstrom et al., 2024). QoQ applies gradient-based influence estimation to robot cu-



(a) RoboDrop pipeline. (b) Global versus local reference gradients.

Figure 1: RoboDrop overview and global-versus-local reference construction.

ration by scoring transitions at a fixed checkpoint against individual validation samples (Lee et al., 2026). However, static or task-level quality signals can be insufficient for multi-stage robot behavior: supervision that is appropriate for one interaction state may be incompatible with another, and its utility may change as the policy is post-trained. RoboDrop addresses these challenges jointly by auditing supervision relative to context-matched clean behavior along the post-training trajectory.

Gradient-Based Data Attribution. Data valuation methods quantify the contribution of training examples to model behavior or validation performance (Hammoudeh & Lowd, 2024). Data Shapley provides an axiomatic formulation based on expected marginal utility over data subsets, but typically requires repeated retraining and is consequently expensive (Ghorbani & Zou, 2019). Influence functions and gradient-based estimators provide tractable approximations to a sample's effect on validation loss (Koh & Liang, 2017; Pruthi et al., 2020; Park et al., 2023). In-Run Data Shapley further accumulates first-order training–validation gradient interactions over a single optimization trajectory (Wang et al., 2025). Although gradient-based attribution is broadly applicable, recent formulations have primarily been developed for data valuation in language-model training. Robot demonstrations exhibit a different structure: an episode traverses heterogeneous interaction stages, and supervision may be unreliable only at particular states. A single global validation objective can therefore obscure whether a candidate action is appropriate for its specific task and interaction context. RoboDrop adapts training-trajectory attribution to robot post-training by constructing context-conditioned validation references, thereby assessing whether local supervision supports clean behavior at a comparable task state.

3 Preliminaries

We study the post-training of a pretrained VLA policy π_{θ_0} on a collection of task-specific candidate episodes $\mathcal{E}_{\text{cand}} = \{\tau_i\}_{i=1}^{N}$, where $\tau_i = \{z_{i,r}\}_{r=1}^{L_i}$ and $z_{i,r} = (o_{i,r}, \ell_i, A_{i,r})$ comprises a multimodal observation, a language instruction, and an action chunk.

We consider flow-matching VLA post-training. For a sample $z = (o, \ell, A)$, we draw a Gaussian noise vector $A^0 \sim \mathcal{N}(0, I)$ and a flow time $\lambda \sim \mathcal{U}(0, 1)$, forming the interpolated action $A^\lambda = (1-\lambda)A^0 + \lambda A$. The velocity field $v_\theta(A^\lambda, o, \ell, \lambda)$ is trained to transport A^0 toward the demonstrated action chunk A. For a set of episodes $\mathcal{S}$, post-training minimizes the flow-matching objective

$$\mathcal{L}_{\text{FM}}(\theta; \mathcal{S}) = \mathbb{E}_{z \sim \mathcal{Z}(\mathcal{S}),\, A^0 \sim \mathcal{N}(0,I),\, \lambda \sim \mathcal{U}(0,1)} \left[\left\| v_\theta(A^\lambda, o, \ell, \lambda) - (A - A^0) \right\|_2^2 \right], \tag{1}$$

where $\mathcal{Z}(\mathcal{S})$ denotes the set of samples contained in episodes $\mathcal{S}$, and $\theta^\star(\mathcal{S})$ denotes the parameters obtained by minimizing $\mathcal{L}_{\text{FM}}(\theta; \mathcal{S})$ from initialization θ_0.

Although all candidate episodes are collected for the target setting, they may contain locally unreliable observation–action supervision due to imperfect teleoperation or sensing errors. The ideal curation objective is to select the episodes that maximize the downstream task success rate of the resulting post-trained policy:

$$\mathcal{S}^\star \in \arg\max_{\mathcal{S} \subseteq \mathcal{E}_{\text{cand}}} \text{Success}\left(\pi_{\theta^\star(\mathcal{S})}\right). \tag{2}$$

4 ROBODROP

4.1 OVERVIEW

Figure 1(a) summarizes RoboDrop. Given mixed-quality candidate episodes $\mathcal{E}_{\text{cand}}$ and a small clean reference set $\mathcal{E}_{\text{val}}$, RoboDrop performs a one-epoch warm-up run and scores each candidate sample online. Each score measures the sample's gradient compatibility with clean validation behavior from similar context, thereby capturing supervision quality along the evolving post-training trajectory. RoboDrop then aggregates the sample-level scores by episode. When the contamination rate is unknown, besides fixed-ratio filtering, we provide an optional BIC-based procedure that recommends a removal ratio based on the episode-score distribution. Finally, the warm-up policy is discarded, and a fresh pretrained policy is post-trained on the retained episodes together with $\mathcal{E}_{\text{val}}$.

4.2 FROM VALIDATION LOSS TO GRADIENT COMPATIBILITY

Directly optimizing Equation 2 is intractable. Rollout success is non-differentiable with respect to data selection, and evaluating a candidate subset requires post-training a policy and deploying it for environment rollouts. Repeating this procedure over possible subsets is prohibitively expensive. We therefore seek a differentiable proxy that can be evaluated within a single training run.

We assess the policy using its flow-matching loss on the small collection of clean target-task episodes $\mathcal{E}_{\text{val}}$, consistent with the use of held-out imitation loss as a surrogate for policy quality in model-aware data curation (Dass et al., 2026). Let $\mathcal{D}_{\text{val}}$ denote all samples contained in $\mathcal{E}_{\text{val}}$. Because flow matching introduces stochasticity through the noise action and flow time, let $\xi_j^{(k)}$ denote their realization for validation sample z_j at optimization step k. The corresponding stochastic estimate of the clean validation loss is

$$\widehat{\mathcal{L}}_{\text{val}}^{(k)}(\theta) := \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{z_j \in \mathcal{D}_{\text{val}}} \ell_{\text{FM}}\big(\theta; z_j, \xi_j^{(k)}\big). \tag{3}$$

This objective is differentiable and requires no environment interaction. Under this proxy, a useful candidate sample should induce a parameter update that decreases the loss on clean validation behavior.

Consider a candidate sample $z_{i,r}$ encountered at step k. Its stochastic gradient and the gradient induced by the clean validation set are

$$\begin{aligned} g_{i,r}^{(k)} &:= \nabla_\theta \ell_{\text{FM}}\big(\theta_k; z_{i,r}, \xi_{i,r}^{(k)}\big), \\ \bar{g}_{\text{val}}^{(k)} &:= \nabla_\theta \widehat{\mathcal{L}}_{\text{val}}^{(k)}(\theta_k) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{z_j \in \mathcal{D}_{\text{val}}} \nabla_\theta \ell_{\text{FM}}\big(\theta_k; z_j, \xi_j^{(k)}\big). \end{aligned} \tag{4}$$

Here, $-\bar{g}_{\text{val}}^{(k)}$ is the descent direction preferred by the clean validation set as a whole. To estimate the effect of training on $z_{i,r}$, consider the counterfactual one-step update $\theta_{k,i,r}^{+} = \theta_k - \eta_k g_{i,r}^{(k)}$. Conditioning on the sampled flow-matching randomness and assuming standard local smoothness, a Taylor expansion around θ_k gives

$$\widehat{\mathcal{L}}_{\text{val}}^{(k)}(\theta_{k,i,r}^{+}) = \widehat{\mathcal{L}}_{\text{val}}^{(k)}(\theta_k) - \eta_k \left\langle \bar{g}_{\text{val}}^{(k)}, g_{i,r}^{(k)} \right\rangle + O\left(\eta_k^2 \left\| g_{i,r}^{(k)} \right\|_2^2\right). \tag{5}$$

The inner product determines the first-order change in clean validation loss: positive values indicate alignment with the clean objective, while negative values indicate conflict (Koh & Liang, 2017; Wang et al., 2025). However, its magnitude also depends on gradient norms, which vary with the sampled noise action, flow time, and local loss scale. To measure directional compatibility while reducing sensitivity to these scales, we therefore normalize both gradients and define the global compatibility score

$$s_{i,r,\text{global}}^{(k)} := \cos\left(g_{i,r}^{(k)}, \bar{g}_{\text{val}}^{(k)}\right) = \frac{\left\langle g_{i,r}^{(k)}, \bar{g}_{\text{val}}^{(k)} \right\rangle}{\left\| g_{i,r}^{(k)} \right\|_2 \left\| \bar{g}_{\text{val}}^{(k)} \right\|_2}. \tag{6}$$

This normalization preserves the sign and directional agreement motivated by the first-order analysis while suppressing nuisance variation in gradient scale. This provides a scale-robust, sample-level

proxy for supervision utility. However, the global reference averages supervision across potentially unrelated tasks and interaction stages. The next section addresses this limitation by constructing a context-conditioned local validation reference.

4.3 Task-Semantic and Visual-Context Neighborhood Reference

The global reference in Equation 4 averages gradients over the complete validation set. As illustrated in Fig. 1b, this can be problematic for robot manipulation data. A single episode typically traverses several interaction phases, such as reaching, grasping, transporting, and placing, which involve distinct visual contexts and action modes. Multi-task collections further introduce heterogeneity in task semantics. Averaging across these modes can mix irrelevant or conflicting gradients, producing a reference direction that is poorly matched to the candidate sample being evaluated.

RoboDrop instead constructs a candidate-specific reference through task-semantic and visual-context retrieval. Given candidate sample $z_{i,r} = (o_{i,r}, \ell_i, A_{i,r})$, we use T5 instruction embeddings (Raffel et al., 2020) to construct a task-semantic validation pool $\mathcal{C}_i$. If all episodes share the same instruction, this step leaves the validation pool unchanged. Within $\mathcal{C}_i$, we compute the visual similarity $\rho_{i,r,j} = \cos(f_{\mathrm{DINO}}(o_{i,r}), f_{\mathrm{DINO}}(o_j))$ using frozen DINO features and retain the K_{vis} most similar validation samples:

$$\mathcal{N}(i,r) := \mathrm{TopK}_{K_{\mathrm{vis}}}\left(\{z_j \in \mathcal{C}_i : \rho_{i,r,j}\}\right). \tag{7}$$

Cross-episode frames that are more similar in DINO (Oquab et al., 2023) feature space exhibit substantially higher gradient-direction similarity; Appendix C.1 provides the experimental protocol and analysis.

The retrieved validation gradients are aggregated using softmax-normalized visual similarities. Specifically, $w_{i,r,j} \propto \exp(\rho_{i,r,j}/\gamma)$ over $z_j \in \mathcal{N}(i,r)$, where the temperature is set to $\gamma = 0.1$ by default, and $g^{(k)}_{\mathrm{val},j} = \nabla_\theta \ell_{\mathrm{FM}}(\theta_k; z_j, \xi^{(k)}_j)$ denotes the gradient of validation sample z_j. Replacing the global validation gradient with this weighted local reference yields

$$s^{(k)}_{i,r,\mathrm{local}} := \cos\left(g^{(k)}_{i,r}, \sum_{z_j \in \mathcal{N}(i,r)} w_{i,r,j}\, g^{(k)}_{\mathrm{val},j}\right), \qquad w_{i,r,j} = \frac{\exp(\rho_{i,r,j}/\gamma)}{\sum_{z_m \in \mathcal{N}(i,r)} \exp(\rho_{i,r,m}/\gamma)}. \tag{8}$$

Unlike the global score, this local compatibility evaluates a candidate update against clean supervision matched to both its task semantics and visual interaction context.

4.4 Online In-Run Scoring and Episode-Level Aggregation

A compatibility score computed at a fixed checkpoint captures only an instantaneous update direction, which can change substantially as the policy evolves during post-training. RoboDrop therefore scores candidate samples online along a one-epoch warm-up trajectory rather than evaluating the entire candidate set at a single model state. Ideally, the clean validation gradients would be recomputed at every optimization step so that each candidate gradient is compared with a reference at exactly the same model state. Doing so, however, would require an additional gradient computation over the validation set at every step, substantially increasing the cost of the warm-up run. RoboDrop therefore refreshes the validation gradients once every M steps and caches them between refreshes. Before the one-epoch warm-up, candidate samples are shuffled. Let $k(i,r)$ denote the unique optimization step at which sample $z_{i,r}$ enters a minibatch. At step $k(i,r)$, its gradient is compared with the most recently cached local validation reference. Each sample thus receives exactly one compatibility score, which is then aggregated within its candidate episode:

$$\bar{s}_{i,r} = s^{(k(i,r))}_{i,r,\mathrm{local}}, \qquad S_i = \frac{1}{L_i}\sum_{r=1}^{L_i} \bar{s}_{i,r}. \tag{9}$$

Frame-level shuffling makes the optimization step assigned to each sample independent of its episode identity. Consequently, the samples of a sufficiently long episode probe compatibility at multiple points along the warm-up trajectory, and their average provides a one-pass estimate of that episode's compatibility across evolving model states. The resulting episode-level score distribution is used for data filtering.

4.5 Optional Automatic Filtering

The episode scores $\{S_i\}_{i=1}^N$ provide a reliability ranking, but turning this ranking into filtering decisions still requires a pre-specified removal ratio. To support flexible deployment on real-world datasets, where the contamination rate is typically unknown and the candidate set may contain no unreliable episodes, we additionally provide an optional automatic filtering procedure. Specifically, the procedure uses BIC to select among a single Gaussian ($K = 1$) and two-component GMMs ($K = 2$) with either shared or component-specific variances:

$$\mathcal{M}^\star = \arg \min_{\mathcal{M} \in \{\mathcal{M}_1, \mathcal{M}_{2,\text{shared}}, \mathcal{M}_{2,\text{specific}}\}} \text{BIC}(\mathcal{M}), \qquad K^\star = K(\mathcal{M}^\star). \tag{10}$$

If $K^\star = 1$, RoboDrop retains all candidate episodes. If $K^\star = 2$, we denote the lower-mean component by $c^- = \arg\min_{c \in \{1,2\}} \mu_c$ and interpret it as unreliable supervision, since lower scores indicate weaker compatibility with the clean validation reference. The retained episode set is

$$\widehat{\mathcal{E}}_{\text{keep}} = \begin{cases} \mathcal{E}_{\text{cand}}, & K^\star = 1, \\ \{\tau_i \in \mathcal{E}_{\text{cand}} \mid p(c^- \mid S_i, \mathcal{M}^\star) \le q\}, & K^\star = 2, \end{cases} \tag{11}$$

where q is a posterior-confidence threshold. It controls the retention–purity trade-off: a larger value removes only episodes classified as unreliable with high confidence and retains more data, whereas a smaller value filters borderline episodes more aggressively to obtain a cleaner retained set. After filtering, we discard the warm-up policy and post-train a fresh copy of the pretrained policy on the retained candidate episodes together with $\mathcal{E}_{\text{val}}$.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate RoboDrop on three sources of mixed-quality robot data with increasing realism. For *LIBERO* (Liu et al., 2023), we construct separate action- and temporal-corruption datasets by corrupting 10% of the episodes in each of the four task suites. *Robomimic-MH* (Mandlekar et al., 2021) comprises four tasks with 300 human teleoperation episodes per task, including 100 collected by proficient operators. Our *real-world* dataset comprises four tasks with 100 episodes per task, including 80 collected by proficient operators. Demonstrations collected by less-skilled operators often contain execution errors and inefficient behavior, making operator proficiency a useful proxy for demonstration quality. We use these provenance labels to evaluate episode-score separability, while assessing the practical utility of curation through downstream policy success. Dataset construction, quality-label definitions, and validation splits are detailed in Appendix A.

Baselines. We compare RoboDrop with four data-selection baselines. *Behavior Retrieval* measures similarity to clean demonstrations using VAE-encoded state–action representations. *DataMIL* estimates data utility through datamodels, using clean validation loss as a differentiable proxy for policy performance. *Scizor* identifies suboptimal and redundant transitions through self-supervised progress estimation and state–action similarity. *QoQ* scores demonstrations using gradient-based influence on validation examples at a fixed policy checkpoint without task-semantic or visual-context conditioning. For a fair comparison, *QoQ* uses the final checkpoint from the same one-epoch warm-up run used by RoboDrop.

Training and evaluation. Since all methods ultimately assign scalar scores to transitions or episodes, we obtain episode-level scores using each method's native aggregation when available and simple averaging otherwise. We then assess how well these scores separate reliable from unreliable episodes using AUROC and best balanced accuracy. To evaluate the practical effect of curation, we post-train a fresh copy of a pretrained policy on the data retained by each method and report rollout task success. Unless otherwise specified, all experiments use the open-source $\pi_{0.5}$-Base model (Black et al., 2025); methods requiring clean validation references are provided with the same validation episodes. RoboDrop retrieves the $K_{\text{vis}} = 10$ nearest validation samples in DINO feature space and refreshes the cached validation reference gradients every $M = 200$ optimization steps during warm-up training. The warm-up stage lasts for one epoch, covering every candidate sample once. To reduce the memory cost of scoring, RoboDrop computes gradients

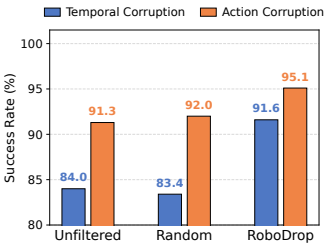


Figure 2: LIBERO rollout success rate after data curation.

Table 1: Corruption discrimination and downstream curation performance on LIBERO-10. T and A denote temporal and action corruption. All results are percentages.

	AUROC ↑			Best bal. acc. ↑			Curated success ↑		
Method	T	A	Avg.	T	A	Avg.	T	A	Avg.
BR	81.8	91.0	86.4	78.7	86.6	82.7	83.5	86.0	84.8
DataMIL	68.6	**100.0**	84.3	65.7	**100.0**	82.9	81.0	**95.0**	88.0
Scizor	47.7	47.4	47.6	54.9	56.0	55.5	76.0	85.5	80.8
QoQ	61.6	**100.0**	80.8	61.1	**100.0**	80.5	81.0	**95.0**	88.0
RoboDrop	**96.1**	98.7	**97.4**	**91.6**	95.3	**93.5**	**89.0**	94.5	**91.8**

only with respect to the action expert parameters and applies CountSketch (Charikar et al., 2002; Schioppa, 2024; Xia et al., 2024) to compress each gradient to 4096 dimensions before computing compatibility scores. Detailed training and evaluation procedures are provided in Appendix D.

5.2 Controlled Corruptions on LIBERO

We evaluate RoboDrop on all four LIBERO task suites under both temporal and action corruption. In each setting, RoboDrop scores the candidate episodes, removes the lowest-scoring 10%, and post-trains a fresh policy on the retained data. We compare its rollout success with post-training on the unfiltered data and with randomly removing the same 10% budget. As shown in Figure 2, RoboDrop improves the average success rate from 84.0% (unfiltered) and 83.4% (random removal) to 91.6% under temporal corruption, and from 91.3% and 92.0% to 95.1% under action corruption. These results show that the episode ranking translates directly into more effective post-training across distinct corruption mechanisms. Per-suite AUROC, best balanced accuracy, and rollout-success comparisons are reported in Appendix B.1.

Comparison with data-selection baselines. Table 1 compares the episode rankings produced by RoboDrop and the baselines on LIBERO-10. RoboDrop performs consistently across both corruption types. It substantially outperforms all baselines under temporal corruption, achieving an AUROC of 96.1% compared with 81.8% for the strongest baseline. DataMIL and QoQ perfectly separate the controlled action corruption, but deteriorate markedly under temporal misalignment. RoboDrop achieves the strongest average discrimination across the two corruptions and the highest average post-filtering success rate. These results suggest that action-space and fixed-checkpoint signals can be effective for direct action perturbations, whereas detecting observation–action misalignment benefits from a context-conditioned reference evaluated online during training.

5.3 Naturally Varying Demonstrations and Backbone Generality

The controlled LIBERO experiments isolate specific corruption mechanisms, but real demonstration quality is rarely binary or synthetically generated. We therefore evaluate whether RoboDrop can distinguish naturally occurring quality variation in Robomimic-MH, where demonstrations are collected from operators with different proficiency levels. Following the protocol in Section 5.1, we treat the designated proficient demonstrations as reliable and the remaining demonstrations as suboptimal. These labels provide an evaluation criterion rather than supervision to the curation methods.

Comparison with data-selection baselines. Table 2 summarizes episode-score separability averaged over the four Robomimic-MH tasks. RoboDrop uses $\pi_{0.5}$-Base for scoring and achieves the highest mean AUROC and balanced accuracy, indicating the strongest overall discrimination. We next assess downstream policy performance by discarding the lowest-ranked 50% of demonstrations under each curation method and post-training a fresh X-VLA-Base model on the retained data. As shown in Figure 3, RoboDrop achieves the highest success rate on all tasks except Lift, where performance is near saturation. It attains 74.67% mean success across the four tasks, exceeding unfiltered post-training (52.34%) by 22.33 percentage points. This substantial gain suggests that suboptimal behavior within ultimately successful demonstrations can impair policy learning: completing a task does not ensure that every intermediate observation–action pair provides useful supervision. The

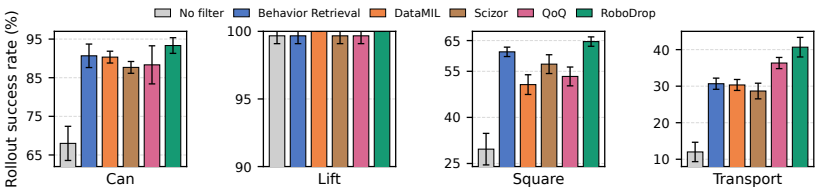


Figure 3: Downstream rollout success after curating the Robomimic-MH post-training data.

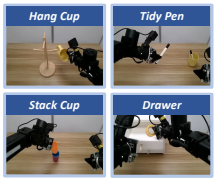


Figure 4: The four real-world tasks.

Table 2: Suboptimal-episodes discrimination on Robomimic-MH (%).

Method	AUROC ↑	Best bal. acc. ↑
Behavior Retrieval	80.99	75.02
DataMIL	76.46	72.00
Scizor	80.52	77.24
QoQ	83.67	76.20
RoboDrop	**88.33**	**81.83**

Table 3: Suboptimal-episodes discrimination for RoboDrop across VLA backbones on Robomimic-MH (%).

Backbone	AUROC ↑	Best bal. acc. ↑
$\pi_{0.5}$-Base	**88.33**	**81.83**
GR00T	85.87	81.21
X-VLA	85.35	78.46
X-VLA (Florence)	80.12	75.32

results highlight the value of post-training data curation beyond simply retaining successful demonstrations. Detailed per-task results are reported in Appendix B.2.

Generalization across VLA backbones. RoboDrop defines compatibility through gradients of the post-training objective and is therefore not tied to a particular VLA architecture. We test this property using $\pi_{0.5}$-Base, GR00T, and X-VLA (Zheng et al., 2026). As shown in Table 3, RoboDrop maintains a mean AUROC above 85% with each pretrained VLA backbone, with relatively small variation in mean balanced accuracy. We further evaluate a weaker X-VLA initialization with a Florence-initialized vision–language backbone and a randomly initialized action expert. RoboDrop retains a mean AUROC of 80.12%, suggesting that task-specific training can yield useful compatibility signals without action pretraining. Fully pretrained X-VLA improves mean AUROC and balanced accuracy by 5.23 and 3.14 percentage points, indicating that VLA pretraining strengthens these signals but is not essential. Per-task results are provided in Appendix B.2.

5.4 Real-World Data Curation

We finally evaluate RoboDrop on demonstrations collected through real-robot teleoperation, where collection errors arise naturally. The four tasks are illustrated in Figure 4. Detailed task descriptions and the complete dataset and validation split are provided in Appendix A.3.

Table 4: Average non-expert identification on real-robot tasks (%).

Method	AUROC ↑	Best bacc ↑
Behavior Retrieval	94.77	90.89
DataMIL	86.31	85.09
Scizor	79.49	80.00
QoQ	83.31	82.14
RoboDrop	**98.38**	**95.63**

Comparison with data-selection baselines. We first examine whether RoboDrop remains sensitive to demonstration quality when errors arise naturally during real-robot collection. As shown in Table 4, RoboDrop achieves the highest average AUROC and best balanced accuracy across the four tasks. The gains over the best baseline Behavior Retrieval indicate that local gradient compatibility goes beyond representation-level similarity, better capturing whether a demonstration induces an update consistent with clean, task-relevant supervision. We next examine whether the stronger episode-level discrimination translates into improved downstream policies. To compare episode-ranking quality independently of budget selection, we apply a common 20% removal budget to all methods and remove the lowest-scoring episodes under each ranking. We post-train a fresh $\pi_{0.5}$-Base policy on each curated dataset and evaluate 20 rollouts per task. As shown in Table 5, RoboDrop improves average success from 35.00% without filtering to 63.75%, outperforming all baselines. These results show that the quality distinctions captured by local gradient compatibility translate into more effective post-training data selection.

Budget selection in practice. The fixed-budget comparison above evaluates which episodes are selected under a common removal budget, but assumes prior knowledge of how much data to remove. As a practical implementation when this information is unavailable, we fit one- and two-

Table 5: Per-task real-world rollout success after curation (successful rollouts / 20).

		Fixed 20% removal					RoboDrop variants		
Task	Unfiltered	BR	DataMIL	Scizor	QoQ	RoboDrop	Fixed 10%	Fixed 30%	Automatic
HANG CUP	2/20	8/20	5/20	4/20	5/20	9/20	6/20	8/20	9/20
TIDY PEN	7/20	12/20	8/20	8/20	9/20	12/20	9/20	12/20	14/20
STACK CUP	10/20	13/20	11/20	14/20	12/20	14/20	14/20	13/20	14/20
DRAWER	9/20	14/20	15/20	10/20	11/20	16/20	11/20	16/20	17/20
Average SR (%)	35.0	58.75	48.75	45.0	46.25	63.75	50.0	61.25	**67.5**

component Gaussian mixture models to the RoboDrop episode scores, select between them using BIC, and remove episodes assigned to the lower-score component with posterior confidence above $q = 0.8$. Without access to the contamination rate, this procedure achieves a competitive average success rate of 67.50% relative to manually specified fixed budgets. Additional diagnostics across contamination rates from 10% to 40% are reported in Appendix B.3.

5.5 ABLATION STUDIES

We ablate RoboDrop on the Robomimic-MH CAN task using $\pi_{0.5}$-Base over three runs with different clean validation sets and training seeds. RoboDrop achieves 87.67 ± 0.87 AUROC and 80.82 ± 1.45 balanced accuracy, indicating stable performance under changes in both reference-set composition and training stochasticity. As shown in Table 6, replacing the local gradient reference with a gradient averaged over the full validation set lowers AUROC and balanced accuracy by 8.03 and 7.61 points. This suggests that global averaging obscures state-relevant supervision, supporting a local reference for evaluating update compatibility. Fixed-checkpoint scoring evaluates all candidates at the checkpoint obtained after the one-epoch warm-up run and achieves 83.50 AUROC and 75.81 balanced accuracy, supporting online scoring along the evolving warm-up trajectory rather than evaluation at its final model state.

Table 6: Robomimic-MH CAN ablations over three runs with different clean sets and training seeds (mean $\pm$ std, %).

Variant	AUROC ↑	Best bacc ↑
Global gradient reference	79.64 ± 0.36	73.21 ± 0.66
Fixed checkpoint	83.50 ± 0.39	75.81 ± 0.43
DINO similarity only	75.86 ± 1.00	69.23 ± 1.29
DINO + Action similarity	79.73 ± 0.26	73.79 ± 0.74
Dot product score	81.78 ± 2.44	74.60 ± 2.24
RoboDrop	$\mathbf{87.67 \pm 0.87}$	$\mathbf{80.82 \pm 1.45}$

We further compare two non-gradient alternatives. *DINO similarity only* scores each candidate using the softmax-weighted visual similarities of its 10 nearest validation samples, while *DINO + Action similarity* additionally incorporates action similarity over the subsequent 10 steps. Adding action information improves AUROC from 75.86 to 79.73, but remains 7.94 points below RoboDrop. This suggests that visual and action similarity provide useful but incomplete quality signals; gradient compatibility instead relates observations and actions through the policy loss to assess their induced update directions. Finally, replacing cosine similarity with the unnormalized gradient dot product lowers AUROC and balanced accuracy by 5.89 and 6.22 points. Cosine also reduces the AUROC standard deviation from 2.44 to 0.87, supporting normalization of overall gradient scales for both discrimination and stability. Additional sensitivity analyses are reported in Appendix C.

6 CONCLUSION

RoboDrop curates VLA post-training data by auditing supervision through context-conditioned local gradient compatibility evaluated along the warm-up trajectory. Across controlled corruptions, naturally varying demonstrations and real-robot collection errors, RoboDrop consistently discriminates data quality and improves downstream policy success, establishing training-dynamics-aware supervision auditing as a practical tool for VLA data curation. A remaining limitation is its reliance on a clean reference set, whose collection or verification may still require expert effort; reducing this dependence is an important direction for future work.

AI USE STATEMENT

For activities requiring disclosure, we used generative AI tools to assist in designing comparisons with baseline methods. We also used AI-assisted programming tools for codebase organization and project management. We did not use generative AI to generate synthetic datasets; develop theoretical models or conceptual frameworks; formulate mathematical claims; provide critical ingredients for mathematical proofs; propose or refine hypotheses; clean or reformat datasets; conduct qualitative or thematic data analysis. For activities for which disclosure is recommended, we used generative AI to assist with language editing and improve the clarity of author-written text, including portions of the introduction and related work sections. The authors reviewed all AI-assisted experimental plans, baseline selections, manuscript text, and code changes. All functional code implementations were inspected, tested, and validated by the authors before use. We take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the complete mathematical formulation of RoboDrop, including its gradient-compatibility score, local validation reference, online in-run scoring, and automatic filtering rule, in Section 4. The datasets, baselines, scoring metrics, validation protocol, and rollout evaluation setup are described in Section 5.1. Complete per-suite and per-task results are reported in Appendix B, while dataset construction, corruption procedures, quality-label definitions, and train–validation splits are documented in Appendix A. Additional ablations and hyperparameter sensitivity analyses are provided in Appendix C, and the warm-up, filtering, post-training, and compute configurations are specified in Appendix D.

REFERENCES

Christopher Agia, Rohan Sinha, Jingyun Yang, Rika Antonova, Marco Pavone, Haruki Nishimura, Masha Itkina, and Jeannette Bohg. Cupid: Curating data your robot loves with influence functions. *arXiv preprint arXiv:2506.19121*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *International colloquium on automata, languages, and programming*, pp. 693–703. Springer, 2002.

Annie S Chen, Alec M Lessing, Yuejiang Liu, and Chelsea Finn. Curating demonstrations using online experience. *arXiv preprint arXiv:2503.03707*, 2025.

Shivin Dass, Alaa Khaddaj, Logan Engstrom, Aleksander Madry, Andrew Ilyas, and Roberto Martín-Martín. Datamil: Selecting data for robot imitation learning with datamodels. In *International Conference on Learning Representations*, volume 2026, pp. 1311–1336, 2026.

Maximilian Du, Suraj Nair, Dorsa Sadigh, and Chelsea Finn. Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets. *arXiv preprint arXiv:2304.08742*, 2023.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. Dsdm: Model-aware dataset selection with datamodels, 2024. URL https://arxiv.org/abs/2401.12926.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.

Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: a survey. *Machine Learning*, 113(5):2351–2403, March 2024. ISSN 1573-0565. doi: 10.1007/s10994-023-06495-7. URL http://dx.doi.org/10.1007/s10994-023-06495-7.

Joey Hejna, Chethan Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. Re-mix: Optimizing data mixtures for large scale imitation learning, 2024. URL https://arxiv.org/abs/2408.14037.

Joey Hejna, Suvir Mirchandani, Ashwin Balakrishna, Annie Xie, Ayzaan Wahid, Jonathan Tompson, Pannag Sanketi, Dhruv Shah, Coline Devin, and Dorsa Sadigh. Robot data curation with mutual information estimators. 2025. URL https://arxiv.org/abs/2502.08623.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. Pmlr, 2017.

Sateesh Kumar, Shivin Dass, Georgios Pavlakos, and Roberto Martín-Martín. COLLAGE: Adaptive fusion-based retrieval for augmented policy learning. 2025.

Haeone Lee, Taywon Min, Junsu Kim, Sinjae Kang, Fangchen Liu, Lerrel Pinto, and Kimin Lee. Quality over quantity: Demonstration curation via influence functions for data-centric robot learning. *arXiv preprint arXiv:2603.09056*, 2026.

Shalfun Li, Victor Yao, Charles Yang, Truth Qu, Regis Cheng, Ryan Yu, Howard Lu, Newton Von, Vincent Chen, Yohann Tang, et al. Wall-wm: Carving world action modeling at the event joints. *arXiv preprint arXiv:2606.01955*, 2026.

Yunfei Li, Xiao Ma, Jiafeng Xu, Yu Cui, Zhongren Cui, Zhigang Han, Liqun Huang, Tao Kong, Yuxiao Liu, Hao Niu, et al. Gr-rl: Going dexterous and precise for long-horizon robotic manipulation. *arXiv preprint arXiv:2512.01801*, 2025.

Li-Heng Lin, Yuchen Cui, Amber Xie, Tianyu Hua, and Dorsa Sadigh. Flowretrieval: Flow-guided data retrieval for few-shot imitation learning. *arXiv preprint arXiv:2408.16944*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021.

Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. Strap: Robot sub-trajectory retrieval for augmented policy learning. In *International Conference on Learning Representations*, volume 2025, pp. 63813–63832, 2025.

NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. In *International Conference on Machine Learning (ICML)*, 2023.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in neural information processing systems*, 33: 19920–19930, 2020.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Andrea Schioppa. Efficient sketches for training data attribution and studying the loss landscape. *Advances in Neural Information Processing Systems*, 37:37692–37735, 2024.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Jiachen Tianhao Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data shapley in one training run. In *International conference on learning representations*, volume 2025, pp. 12358–12395, 2025.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026.

Yu Zhang, Yuqi Xie, Huihan Liu, Rutav Shah, Michael Wan, Linxi Fan, and Yuke Zhu. Scizor: A self-supervised approach to data curation for large-scale imitation learning. *arXiv preprint arXiv:2505.22626*, 2025.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

A DATASETS

A.1 LIBERO

LIBERO is a benchmark for lifelong robot manipulation comprising four suites with ten tasks each. LIBERO-Spatial evaluates sensitivity to spatial relationships between objects, LIBERO-Object emphasizes generalization across different object categories, LIBERO-Goal tests the execution of diverse language-specified goals in related scenes, and LIBERO-Long focuses on long-horizon tasks requiring multiple sequential manipulation skills, with each suite containing ten tasks. We construct the action- and temporal-corruption datasets separately by corrupting 10% of the episodes for every task. Action corruption adds zero-mean Gaussian noise with standard deviation $0.25\sigma_a$ to the demonstrated actions, where σ_a is the empirical action standard deviation. Temporal corruption shifts the action sequence by two seconds relative to the observations. For validation, we reserve two uncorrupted episodes per task, giving 20 clean validation episodes for each suite.

A.2 ROBOMIMIC-MH

Robomimic-MH contains 300 human teleoperation episodes for each of the four tasks. We designate 100 proficient episodes per task as reliable and treat the remaining 200 as naturally suboptimal. The proficient episodes correspond to the `better` split for most tasks; for TRANSPORT, we combine the `better` and `okay_better` splits to obtain 100 episodes. We reserve ten proficient episodes per task as clean validation episodes, leaving 290 candidate episodes for scoring. These provenance labels are used only for evaluation and are not supplied to the curation methods.

A.3 REAL-WORLD DATASET

The real-world dataset contains four tasks with 100 teleoperated episodes per task, collected using ARX-series robot arms. For each task, 80 episodes are collected by proficient operators and 20 by non-expert operators. We reserve ten proficient episodes as clean validation data, leaving 90 candidate episodes comprising 70 proficient and 20 non-expert demonstrations. The non-expert subset contains many suboptimal trajectories characterized by grasp failures, unnecessary or inefficient motions, and other execution errors. Operator provenance is used only for evaluation and is unavailable to RoboDrop and the baselines.

Figure 5 shows representative execution sequences for all four tasks. In HANG CUP, one arm grasps the cup and places the opening of its handle over a branch of the cup rack. In TIDY PEN, the right arm first picks up the marker on the right and places it in the pen holder; the left arm then picks up the marker on the left, hands it to the right arm, and the right arm places it in the holder. In STACK CUP, the right arm stacks the rightmost cup onto the center cup, after which the left arm places the leftmost cup on top. In DRAWER, the left arm opens the drawer, the right arm picks up the roll of tape and places it inside, and the left arm closes the drawer.

B DETAILED RESULTS

B.1 DETAILED LIBERO RESULTS

Table 7 provides the complete breakdown for the four LIBERO task suites and both corruption types. RoboDrop maintains strong score separability across all eight suite–corruption settings, with an average AUROC of 96.20% and best balanced accuracy of 93.13%. Its ranking also yields consistent downstream gains: removing the lowest-scoring 10% of episodes raises average rollout success from 87.63% with unfiltered post-training and 87.69% with random removal to 93.38%. The improvement is larger under temporal corruption ($+7.63$ percentage points over unfiltered data) than under action corruption ($+3.88$ points), consistent with temporal misalignment being particularly harmful when left unfiltered. RoboDrop also recovers most of the 95.69% success achieved by the oracle that removes exactly the corrupted episodes.

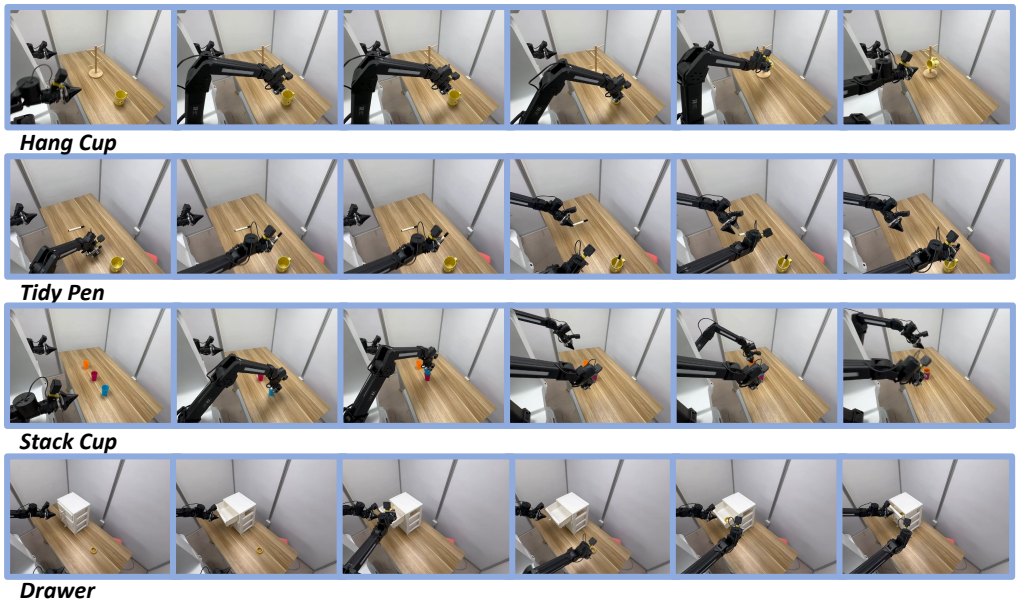


Figure 5: Detailed execution sequences for the four real-world manipulation tasks. Each row shows the principal stages of one task.

Table 7: Per-suite controlled-corruption results on LIBERO (%).

		Score separability		Rollout success			
Suite	Corruption	AUROC	Best bal. acc.	Unfiltered	Random	**RoboDrop**	Oracle
LIBERO-Object	Temporal	98.27	93.84	87.5	82.5	95.0	96.0
	Action	98.78	96.09	97.5	95.5	97.5	97.5
LIBERO-Goal	Temporal	93.96	90.33	79.5	82.5	89.0	94.0
	Action	87.66	89.71	87.5	91.5	95.0	95.5
LIBERO-Spatial	Temporal	98.46	93.96	85.5	89.0	93.5	97.5
	Action	97.68	94.11	92.5	93.0	93.5	97.0
LIBERO-10	Temporal	96.11	91.62	83.5	79.5	89.0	93.5
	Action	98.70	95.34	87.5	88.0	94.5	94.5
Average	Temporal	96.70	92.44	84.00	83.38	**91.63**	95.25
	Action	95.71	93.81	91.25	92.00	**95.13**	96.13

B.2 Detailed RoboMimic Results

Tables 9 and 10 provide the per-task results underlying the four-task averages in Tables 2 and 3. For data selection, RoboDrop achieves the best AUROC on Can and Square, remains close to the strongest method on Lift and Transport, and obtains the best balanced accuracy on three of the four tasks. In contrast, the strongest baselines vary by task: Scizor performs best on Lift, while QoQ attains the highest AUROC on Transport. This variation explains why RoboDrop's consistent task-level performance yields the strongest overall averages.

Across backbones, $\pi_{0.5}$-Base gives the strongest results on Can and Square and the highest mean performance. GR00T is competitive on Lift, while X-VLA performs best on Transport, showing that the compatibility signal transfers across architectures despite task-specific variation. The Florence-initialized X-VLA remains above chance on all tasks but is consistently weaker on average, supporting the benefit of a pretrained action model while confirming that RoboDrop does not require a particular backbone.

Table 8 reports the per-task rollout results underlying Figure 3. Methods follow the same order as the figure legend, and the average is the arithmetic mean of the four task means.

Table 8: Per-task downstream rollout success on Robomimic-MH (mean ± std, %).

Method	CAN	LIFT	SQUARE	TRANSPORT	Average
No filter	68.00 ± 4.42	99.67 ± 0.58	29.67 ± 5.13	12.00 ± 2.65	52.34
Behavior Retrieval	90.67 ± 3.04	99.67 ± 0.58	61.33 ± 1.53	30.67 ± 1.53	70.59
DataMIL	90.33 ± 1.53	100.00 ± 0.00	50.67 ± 3.21	30.33 ± 1.50	67.83
Scizor	87.67 ± 1.53	99.67 ± 0.58	57.33 ± 3.06	28.67 ± 2.14	68.34
QoQ	88.33 ± 4.93	99.67 ± 0.58	53.33 ± 3.06	36.33 ± 1.53	69.42
RoboDrop	$\mathbf{93.33 \pm 2.03}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{64.67 \pm 1.53}$	$\mathbf{40.67 \pm 2.68}$	**74.67**

Table 9: Per-task identification of naturally suboptimal Robomimic-MH demonstrations. Entries are AUROC / best balanced accuracy (%).

Method	CAN	LIFT	SQUARE	TRANSPORT	Average
Behavior Retrieval	85.59 / 75.89	88.54 / 79.58	82.61 / 77.94	67.22 / 66.67	80.99 / 75.02
DataMIL	78.86 / 74.89	79.04 / 71.39	74.84 / 71.89	73.11 / 69.83	76.46 / 72.00
Scizor	79.29 / 76.33	**90.51 / 89.72**	87.63 / 81.06	64.63 / 61.86	80.52 / 77.24
QoQ	83.41 / 75.39	82.78 / 74.83	80.17 / 74.00	**88.32** / 80.58	83.67 / 76.20
RoboDrop	**88.88 / 82.75**	87.23 / 79.44	**89.07 / 83.14**	88.15 / **81.97**	**88.33 / 81.83**

Table 10: Per-task RoboDrop results across VLA backbones on Robomimic-MH. Entries are AUROC / best balanced accuracy (%).

Backbone	CAN	LIFT	SQUARE	TRANSPORT	Average
$\pi_{0.5}$-Base	**88.88 / 82.75**	**87.23** / 79.44	**89.07 / 83.14**	88.15 / 81.97	**88.33 / 81.83**
GR00T	87.87 / 81.00	86.78 / **82.64**	83.10 / 79.97	85.72 / 81.22	85.87 / 81.21
X-VLA	88.46 / 80.39	74.51 / 70.11	88.74 / 79.39	**89.68 / 83.94**	85.35 / 78.46
X-VLA (Florence init.)	77.84 / 72.64	74.15 / 68.22	81.73 / 77.14	86.74 / 83.28	80.12 / 75.32

B.3 DETAILED REAL-WORLD FILTERING RESULTS

Table 11 gives the per-task results underlying the averages in Table 4. RoboDrop obtains the strongest AUROC and best balanced accuracy on every task. In contrast, the relative strength of the baselines varies by task, indicating that no single trajectory-level heuristic consistently captures non-expert behavior.

Table 11: Per-task real-robot non-expert identification results. Entries are AUROC / best balanced accuracy (%).

Method	HANG CUP	TIDY PEN	STACK CUP	DRAWER	Average
Behavior Retrieval	99.57 / 98.57	93.00 / 85.00	95.79 / 92.14	90.71 / 87.86	94.77 / 90.89
DataMIL	84.50 / 82.14	80.79 / 85.36	87.86 / 88.57	92.07 / 84.29	86.31 / 85.09
Scizor	62.50 / 71.43	81.32 / 76.79	98.32 / 96.79	75.82 / 75.00	79.49 / 80.00
QoQ	81.50 / 85.36	81.43 / 81.07	91.04 / 86.07	79.25 / 76.07	83.31 / 82.14
RoboDrop	**100.00 / 100.00**	**97.57 / 92.86**	**99.79 / 97.86**	**96.14 / 91.79**	**98.38 / 95.63**

Adapting to unknown contamination rates. A discriminative ranking alone does not determine how many episodes should be removed. We therefore test whether RoboDrop's automatic procedure adapts when the prevalence of non-expert data changes. For each task, we construct subsets at contamination rates ranging from 10% to 40% using multiple random samplings. We use the same posterior confidence threshold $q = 0.8$ in all settings without tuning it to the contamination rate. As shown in Table 12, the BIC-selected procedure achieves macro-average F1 scores from 85.6% to 93.0% while maintaining precision above 84.0% and recall above 88.2%. Thus, its filtering decisions remain effective as the unknown contamination rate varies.

Table 13 isolates the accuracy of the automatic filtering procedure. A fixed 10% budget achieves perfect precision but only 45.0% recall on every task, whereas a fixed 30% budget improves recall at the cost of substantially lower precision. Without access to the contamination rate, RoboDrop automatically removes 17–21 episodes per task and achieves mean precision, recall, and F1 scores

Table 12: Automatic filtering under varying non-expert rates (%).

Non-expert rate	Precision	Recall	F1
10%	84.0 ± 7.6	92.9 ± 5.6	85.6 ± 6.8
20%	93.4 ± 1.2	88.2 ± 1.0	90.5 ± 1.2
30%	92.3 ± 1.4	90.6 ± 0.6	91.3 ± 0.6
40%	95.3 ± 4.1	91.2 ± 1.5	93.0 ± 1.3

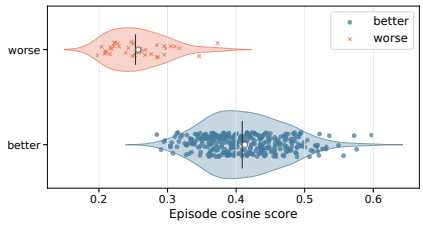
of 93.78%, 90.00%, and 91.70%, respectively, maintaining strong filtering accuracy across all four tasks without a fixed removal budget.

Table 13: Per-task filtering accuracy on 90 non-validation real-robot episodes. Fixed $p\%$ is computed over these 90 candidates (e.g., $20\% = 18$ removed); metrics are percentages.

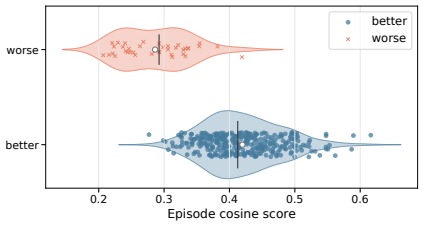
Task	Rule	Rem.	Prec.	Rec.	F1
STACK CUP	Auto.	19	100.0	95.0	**97.4**
	Fixed 20%	18	100.0	90.0	94.7
	Fixed 10%	9	100.0	45.0	62.1
	Fixed 30%	27	74.1	100.0	85.1
DRAWER	Auto.	21	81.0	85.0	82.9
	Fixed 20%	18	94.4	85.0	**89.5**
	Fixed 10%	9	100.0	45.0	62.1
	Fixed 30%	27	66.7	90.0	76.6

Task	Rule	Rem.	Prec.	Rec.	F1
HANG CUP	Auto.	20	100.0	100.0	**100.0**
	Fixed 20%	18	100.0	90.0	94.7
	Fixed 10%	9	100.0	45.0	62.1
	Fixed 30%	27	74.1	100.0	85.1
TIDY PEN	Auto.	17	94.1	80.0	**86.5**
	Fixed 20%	18	88.9	80.0	84.2
	Fixed 10%	9	100.0	45.0	62.1
	Fixed 30%	27	66.7	90.0	76.6

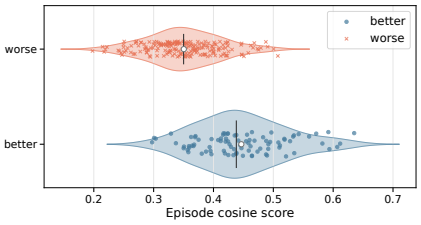
B.4 QUALITATIVE VISUALIZATION



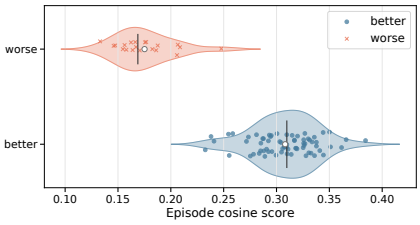
(a) LIBERO-10 action corruption.



(b) LIBERO-10 temporal corruption.



(c) Robomimic-MH SQUARE.



(d) Real-world STACK CUP.

Figure 6: RoboDrop episode cosine-score distributions across simulated and real-world data-quality settings. *Worse* denotes corrupted or non-proficient demonstrations, and *Better* denotes clean or proficient demonstrations.

Figure 6 visualizes the episode-level cosine scores produced by RoboDrop under four data-quality settings. In these plots, *Worse* denotes synthetically corrupted demonstrations or demonstrations collected by non-proficient operators, whereas *Better* denotes clean demonstrations or those collected by proficient operators. The violin plots show the score distributions together with the individual episode scores.

Figure 7 presents representative low-quality trajectories identified by RoboDrop. The examples cover diverse failure modes: knocking over a cup followed by repeated corrective motions, moving a cup around the rack without aligning its handle with the peg, repeatedly adjusting a pen without completing a stable bimanual handover, contacting a drawer handle without producing meaningful displacement, and a prolonged low-progress segment followed by an abrupt late motion near the

target. These cases show that RoboDrop captures locally ineffective or inconsistent behavior within a trajectory, rather than relying only on final task success or simple trajectory-level heuristics.

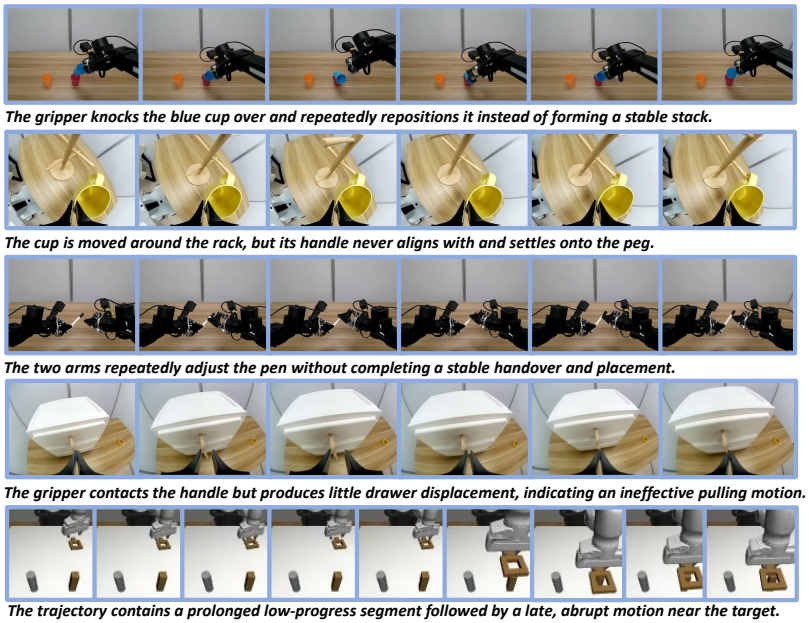


Figure 7: Representative suboptimal trajectories identified by RoboDrop. Each row illustrates a distinct failure mode through selected frames from one trajectory.

C Ablation Studies

The main text studies the effects of the local gradient reference and online in-run scoring. Here, we examine two additional implementation choices: the visual-neighborhood size and the subset of model parameters used for gradient scoring. Table 14 reports their per-task results.

Using only the nearest validation sample ($K_{\text{vis}} = 1$) achieves an average AUROC of 86.54, compared with 88.33 for RoboDrop with $K_{\text{vis}} = 10$. The larger neighborhood provides a more stable local reference across tasks, although the single-neighbor setting is strongest on TRANSPORT. Computing gradients through the full VLA reaches an average AUROC of 87.02, whereas action-expert gradients reach 88.33 and avoid backpropagation through the rest of the model. We therefore use $K_{\text{vis}} = 10$ and action-expert gradients in the default configuration.

Table 14: Additional per-task RoboDrop ablations on Robomimic-MH with $\pi_{0.5}$-Base. Entries are AUROC / best balanced accuracy (%).

Variant	CAN	LIFT	SQUARE	TRANSPORT	Average
Single neighbor ($K_{\text{vis}} = 1$)	85.74 / 79.89	87.08 / 78.33	84.61 / 77.83	**88.72 / 84.64**	86.54 / 80.17
Full-model gradients	86.35 / 79.33	**88.90 / 82.00**	88.19 / 83.03	84.63 / 79.58	87.02 / 80.99
RoboDrop	**88.88 / 82.75**	87.23 / 79.44	**89.07 / 83.14**	88.15 / 81.97	**88.33 / 81.83**

CountSketch dimension. Per-sample action-expert gradients are high-dimensional, so RoboDrop applies CountSketch before computing gradient compatibility. We evaluate target dimensions of 1024, 2048, and 4096 on the four Robomimic-MH tasks. As shown in Table 15, performance is stable across compression levels, while 4096 dimensions achieve the strongest mean AUROC and balanced accuracy (88.33% and 81.83%, respectively). We therefore use 4096 as the default, providing substantial gradient compression without sacrificing the quality signal used for episode ranking.

Validation-gradient refresh interval. RoboDrop refreshes the validation reference gradients every M optimization steps during warm-up. We vary M on the Robomimic-MH SQUARE task while

Table 15: Sensitivity to the CountSketch target dimension on Robomimic-MH with $\pi_{0.5}$-Base (%). Best bacc denotes best balanced accuracy.

	CAN		LIFT		SQUARE		TRANSPORT	
Dimension	AUROC	Best bacc	AUROC	Best bacc	AUROC	Best bacc	AUROC	Best bacc
1024	86.62	79.07	88.40	**80.51**	85.35	79.52	87.78	**82.39**
2048	86.37	79.26	**88.71**	80.01	84.56	78.21	**88.22**	82.33
4096	**88.88**	**82.75**	87.23	79.44	**89.07**	**83.14**	88.15	81.97

holding all other settings fixed. Table 16 shows that $M = 200$ achieves the best AUROC and balanced accuracy, while performance generally declines when the interval is increased beyond 400 steps. Refreshing every 100 steps performs comparably but does not improve over $M = 200$. These results support the default interval of $M = 200$ and indicate that overly stale validation gradients weaken the compatibility signal.

Table 16: Sensitivity to the validation-gradient refresh interval M on Robomimic-MH SQUARE with $\pi_{0.5}$-Base (%).

Refresh interval M	AUROC	Best bacc
100	88.17	82.70
200	**89.07**	**83.14**
400	85.93	79.26
600	84.36	77.08
800	77.83	72.70
1000	78.76	73.32

C.1 DINO SIMILARITY AND CROSS-EPISODE GRADIENT ALIGNMENT

We validate DINO similarity as the visual retrieval signal using $\pi_{0.5}$-Base on the SQUARE task from Robomimic-MH. At a fixed post-training checkpoint, we compute action-expert gradients for individual frames using the same model parameters. To prevent temporally adjacent frames from trivially matching, all comparisons are cross-episode: for each query frame, we exclude its source episode and rank frames from the remaining episodes by cosine similarity between their frozen DINO features. For each neighborhood size k, we then compare the average gradient-direction cosine of the k DINO-nearest frames with that of k randomly sampled cross-episode frames.

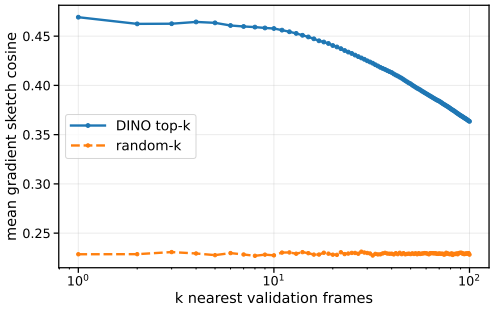


Figure 8: Cross-episode gradient-direction similarity for DINO-nearest and randomly sampled frames on Robomimic-MH SQUARE.

As shown in Figure 8, DINO-nearest frames have substantially higher gradient-direction similarity than random cross-episode frames. Their alignment gradually decreases as k grows and progressively less similar frames enter the neighborhood, while the random baseline remains low. This relationship indicates that visually similar manipulation states tend to support similar update directions and therefore justifies DINO similarity for constructing the local validation reference.

D TRAINING AND EVALUATION DETAILS

Warm-up Training. Unless otherwise specified, experiments use the open-source $\pi_{0.5}$-Base model. RoboDrop performs one warm-up epoch over the complete candidate set. Candidate samples are shuffled before warm-up and each is scored exactly once, at the step when it enters a minibatch. For each candidate sample, RoboDrop retrieves $K_{\text{vis}} = 10$ validation samples after task-semantic matching and uses their most recently cached reference gradients. The validation gradients are refreshed every $M = 200$ optimization steps and reused between refreshes; no additional candidate-

set scoring sweep is performed. Only the action-expert parameters participate in per-sample gradient scoring. Each gradient is compressed to 4096 dimensions with CountSketch before local cosine compatibility is computed and stored for episode-level aggregation. Warm-up training uses four NVIDIA A100 GPUs with a global batch size of 32. A one-epoch warm-up run takes approximately 2 hours on each of LIBERO-Spatial, LIBERO-Object, and LIBERO-Goal, 4 hours on LIBERO-10, 5.4 hours on average across the four Robomimic tasks and 4.5 hours on average across the four real-robot tasks.

Post-training. Unless a fixed removal budget is specified, RoboDrop uses BIC to select between the single-Gaussian and two-component score models described in Section 4.5. It retains every candidate episode when the single Gaussian is selected. When the two-component model is selected, episodes whose posterior probability of belonging to the lower-mean component exceeds $q = 0.8$ are removed. The warm-up policy is then discarded, and a fresh pretrained policy is post-trained on the retained candidate episodes together with the clean validation episodes.

For LIBERO, we fully fine-tune all parameters of $\pi_{0.5}$-Base with a global batch size of 32. Post-training runs for 20,000 optimization steps on each of LIBERO-Object, LIBERO-Spatial, and LIBERO-Goal, and for 30,000 steps on LIBERO-10. For Robomimic-MH, we use the open-source X-VLA-Base model and remove the lowest-scoring 50% of the candidate episodes for all selection methods. We fully fine-tune all model parameters with a global batch size of 32 for 10,000 optimization steps on CAN, LIFT, and SQUARE, and for 30,000 steps on TRANSPORT. For each Robomimic-MH task and curation method, we repeat post-training and rollout evaluation over three runs with different training and evaluation seeds, and report the mean success rate across runs. For each real-world task, post-training uses a global batch size of 32 and runs for 30,000 optimization steps.

Evaluation. We evaluate episode-level scores using AUROC and best balanced accuracy. AUROC measures ranking quality across decision thresholds, with 50% corresponding to chance-level discrimination and 100% to perfect separation. Balanced accuracy averages the true-positive and true-negative rates, giving equal weight to both classes:

$$\mathrm{BAcc}(t) = \frac{\mathrm{TPR}(t) + \mathrm{TNR}(t)}{2}.$$

We report the best balanced accuracy over all score thresholds, $\max_t \mathrm{BAcc}(t)$, to characterize achievable class separation. For synthetic corruptions, evaluation labels indicate whether an episode was corrupted; for naturally collected demonstrations, operator proficiency provides a proxy label for quality. We measure policy success as the fraction of successful rollouts. For LIBERO, each policy is evaluated over 20 rollouts per task using the same evaluation seeds across methods. For Robomimic-MH, each method is post-trained with three different training seeds, and each resulting policy is evaluated over 100 rollouts per task, yielding 300 rollouts per task in total. For the real-world experiments, each policy is evaluated over 20 rollouts per task. Suite-level and cross-task results are computed by averaging task-level success rates.