WHAT YOUR POLICY CANNOT SEE: ROBOT DATA CURATION VIA OBSERVABILITY ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot imitation learning performance depends critically on the quality of demonstration data. Even successful demonstrations may omit action-relevant visual information, an aspect of data quality that remains less explored in robot data curation. We introduce FOCUS, an information-theoretic framework for quantifying visual observability in offline robot demonstrations, defined as the extent to which visual observations contain the action-relevant information needed to determine an appropriate action. Our key idea is to test whether an auxiliary observation provides information about the demonstration action beyond what is contained in the visual observation. We instantiate this idea using conditional mutual information between the auxiliary observation and actions given visual observations. We use the resulting observability scores to rank demonstrations, filter those with low visual observability, and train imitation learning policies on the retained data. Empirically, our scores generally rank demonstrations in agreement with human observability judgments across two simulated and four real-robot manipulation tasks. The best-performing FOCUS configurations improve success rates by 5–45 percentage points over training on the full dataset. The project website is available at `https://focus-observability.github.io/`.

1 INTRODUCTION

Modern robot policies are trained on increasingly large, heterogeneous datasets collected across robot platforms, environments, and teleoperators (Khazatsky et al., 2024), which makes data quality difficult to control. Increasing dataset size alone does not guarantee improved downstream policy performance (Du et al., 2023; Hejna et al., 2025; Zhang et al., 2025), motivating data curation methods that select useful robot data. One fundamental requirement of useful data is that the observations selected as policy inputs contain the information needed to predict an appropriate action. For example, during tool insertion, a tool offset to the left or right of a target hole requires correction in opposite directions. If the insertion region is occluded or outside the camera views, these situations may appear indistinguishable to the policy (Fig. 1). Moreover, the teleoperator may act using information not included in these policy inputs, leaving indistinguishable observations paired with conflicting action targets (Weihs et al., 2021). This ambiguity is especially problematic in high-precision tasks, where small differences in alignment or contact state determine the required correction. We refer to the extent to which the policy inputs contain the information needed to determine an appropriate action as *observability*.

Existing robot-data curation methods characterize useful demonstrations through properties such as execution quality (Kuhar et al., 2023; Zhang et al., 2025), diversity and coverage (Hejna et al., 2025; Sirigiri et al., 2026), relevance to the target task (Du et al., 2023), and downstream influence on policy return or validation loss (Agia et al., 2025; Lee et al., 2026). These criteria capture important dimensions of data quality, but do not explicitly separate missing action-relevant information from other sources of action uncertainty. Consequently, even after curation according to these criteria, a dataset may retain demonstrations with poor observability that provide ambiguous supervision. However, detecting such observability failures from demonstration data is difficult. Conditional action entropy provides one possible proxy: DemInf relates it to the influence of privileged information as part of its mutual-information criterion (Hejna et al., 2025). However, high action uncertainty can also arise from variation in demonstrator behavior or action noise (Gandhi et al., 2023; Belkhale

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.

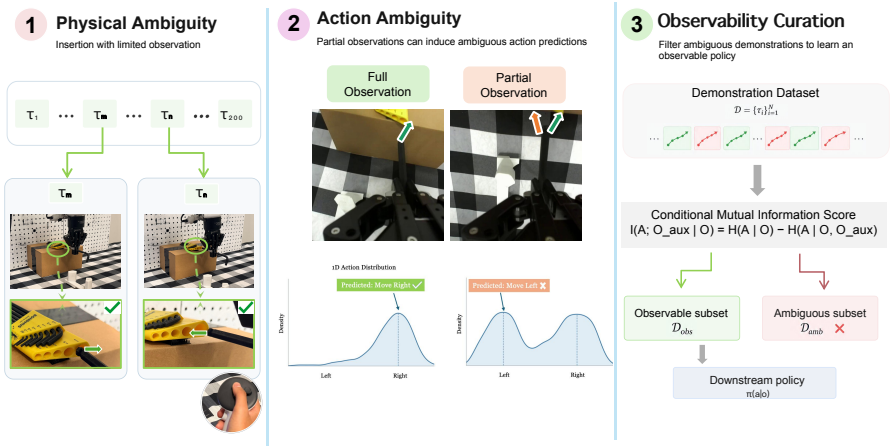


Figure 1: **Visual observability in tool insertion.** A tool offset to the left or right of the holder requires a different corrective action. When the insertion region is occluded in the policy's camera views, these distinct states can appear similar, creating ambiguous action supervision.

et al., 2023; Hejna et al., 2025). Action uncertainty alone therefore cannot determine whether a demonstration is poorly observable.

We introduce FOCUS, a **F**ramework for **O**bservability-based **Cu**ration and **S**coring of offline robot demonstrations grounded in information theory. Our key idea is to estimate observability by measuring how much action-relevant information is missing from the demonstration observations. Rather than relying on action uncertainty alone, we use conditional mutual information (CMI), $I(A; O_{\text{aux}} \mid O)$, between the demonstration action a and an auxiliary observation O_{aux}, conditioned on the recorded demonstration observation O, to quantify how much the auxiliary observation reveals about the demonstration action beyond what the original observation provides. A large CMI value indicates that the original observation is less observable, whereas a small value indicates that it already contains most of the action-relevant information exposed by the auxiliary observation. For ranking, we negate this quantity to define an observability score, so that higher scores indicate better observability.

Although this principle is not specific to visual observations, we primarily study **visual observability**. We consider two practical settings depending on the sensing available during data collection. When an additional exterior camera is available, we use it to directly probe action-relevant scene information missing from the observations available to the policy. When no additional camera is available, we instead use robot proprioception to measure action-relevant information that is available from the robot state but not captured by the visual observations. Observability failures in manipulation are often localized to short, critical interaction phases, so averaging observability over an entire trajectory can obscure the demonstrations that contain such failures. We therefore compute an observability score for each transition and summarize each demonstration by averaging its lowest-scoring (x%) of transitions. We rank demonstrations by this score, discard those with lowest observability scores, and retain the remaining data for policy training.

The main contribution of this work is a CMI-based criterion for robot data curation that assesses observability through the action-relevant information auxiliary observations reveal beyond the recorded observations being evaluated. Across two simulated and four real-robot manipulation tasks, our scores generally align with human observability judgments, with agreement depending on the auxiliary information available. The best-performing FOCUS configurations improve policy success by 5–45 percentage points across these six tasks compared with training on the full dataset. These results suggest that observability is a useful dimension of demonstration quality for imitation learning.

2 Related Work

We review how prior methods quantify different aspects of dataset quality, alongside studies of partial observability in robot learning.

Robot Data Curation. Demonstration quality substantially affects imitation learning, making data selection important for improving downstream policy performance (Mandlekar et al., 2021; Belkhale et al., 2023). Existing curation criteria assess complementary aspects of usefulness, including quality inferred from human preferences (Kuhar et al., 2023), behavioral consistency (Gandhi et al., 2023), task progress (Zhang et al., 2025; Chen et al., 2026; Yu et al., 2026), smoothness (**?**), and diversity or coverage of reusable behaviors (Sirigiri et al., 2026; Wu et al., 2026). Retrieval methods assess relevance to a target task by matching behaviors or sub-trajectories and reweighting source data for transfer (Nasiriany et al., 2022; Du et al., 2023; Lin et al., 2024; Memmel et al., 2025; Xie et al., 2025; Kumar et al., 2025). Another line of work estimates data utility for downstream policy learning. Demo-SCORE (Chen et al., 2025) uses online experience to assess demonstration reliability, while CUPID (Agia et al., 2025) estimates data influence on policy return using evaluation rollouts. Both depend on online environment interaction beyond the offline demonstrations. Offline alternatives include DataMIL (Dass et al., 2026), which predicts data utility using a loss on target-task demonstrations, and QoQ (Lee et al., 2026), which estimates each training sample's influence on validation loss. These methods assess offline data quality through its estimated effect on the policy loss. Most closely related, DemInf (Hejna et al., 2025) uses state–action mutual information to favor diverse yet predictable actions. However, uncertainty in actions conditioned on observations can arise from both missing action-relevant information and variation in demonstrator behavior, so action predictability alone does not distinguish these sources. Unlike all of these prior works, we develop a curation criterion that specifically probes information missing from the policy observations.

Partial Observability in Robot Learning. Partial observability is common in real–world robot learning, where the full environment state is typically unavailable and sensor observations may omit information needed for decision–making. Decision-making under incomplete state information is classically formulated as a partially observable Markov decision process (POMDP) (Åström, 1965; Smallwood & Sondik, 1973). Interactive teacher–student methods adapt guidance or teacher behavior when privileged expert information makes actions difficult to imitate from student observations (Warrington et al., 2021; Weihs et al., 2021; Messikommer et al., 2025), but these methods rely on online access to an expert policy during training. Active-perception methods address information limitations by learning behaviors that acquire useful observations (Xiong et al., 2025; Kerr et al., 2025; Hu et al., 2026; Wang et al., 2026), while our curation setting requires assessing the observations already recorded. Memory-based approaches in reinforcement and imitation learning aggregate past observations and actions to recover information unavailable in the current observations (Heess et al., 2015; Gangwani et al., 2020; Ni et al., 2021; Torne et al., 2025). Such representations can recover information from past experience, but ambiguities may persist when action-relevant information is absent throughout the recorded history. Moreover, history-conditioned imitation may exploit temporal correlations by inferring and repeating previous actions, yielding low demonstration prediction error but poor policy execution (Wen et al., 2020; Swamy et al., 2022). Rather than improving how policies can learn under limited observations, our work addresses observability for offline dataset by filtering out demonstrations whose selected visual inputs lack information needed to predict appropriate actions. We assess visual observability by measuring how much additional action-relevant information is revealed by recorded auxiliary information, such as robot proprioceptive states or additional camera observations, enabling offline curation without expert queries or online environment interactions.

3 Preliminaries

We consider the problem of learning a robot policy from a fixed set of human demonstrations. We first formalize imitation learning from demonstrations and trajectory-level data curation, then review conditional mutual information, which provides the information-theoretic foundation for our method.

3.1 Imitation Learning from Demonstrations

We consider a fixed demonstration dataset $\mathscr{D} = \{\tau_i\}_{i=1}^{N}$, where each trajectory $\tau_i = \{(o_{i,t}, a_{i,t})\}_{t=1}^{T_i}$ consists of observations $o_{i,t}$ and corresponding actions $a_{i,t}$ recorded during demonstration collection.

Given $\mathscr{D}$, behavioral cloning (BC) trains a policy π_θ to predict demonstration actions from the corresponding observations. For policies that predict action chunks, we write the training objective as

$$\theta^\star = \arg\min_\theta \mathbb{E}_{(o_t, a_{t:t+H-1})\sim\mathscr{D}} \left[\ell_{\mathrm{BC}}\left(\pi_\theta(\cdot \mid o_t), a_{t:t+H-1}\right)\right], \tag{1}$$

where H denotes the prediction horizon and ℓ_{BC} denotes the loss between the predicted and demonstration action chunks.

3.2 Demonstration Data Curation

Demonstration data curation selects a subset of $\mathscr{D}$ for downstream policy training according to a scoring function S that assigns each trajectory a scalar score. We consider trajectory-level filtering with a controlled retention ratio $\rho \in (0, 1]$, retaining

$$\mathscr{D}_\rho(S) = \left\{\tau_i \,:\, S(\tau_i) \geq \kappa_\rho\right\}, \tag{2}$$

where κ_ρ is the $\lceil \rho N \rceil$-th largest trajectory score, such that higher-scoring trajectories are preferentially retained. Scoring and filtering are performed using the fixed demonstration dataset and require no additional environment interaction.

4 Quantifying Observability for Demonstration Curation

In this section, we introduce a framework for quantifying observability in robot demonstrations and using it for data curation. We first define visual observability and introduce a CMI-based estimator that measures the action-relevant information revealed by auxiliary observations beyond the recorded demonstration observations. We then instantiate this formulation under two practical sensing settings and use the resulting observability estimates to score and filter demonstrations.

4.1 Estimating Observability with Auxiliary Information

Given a demonstration trajectory $\tau = \{(o_t, a_t)\}_{t=1}^{T}$, we define *visual observability* as the extent to which the recorded visual observations contain the action-relevant information needed to explain the corresponding demonstration actions. Intuitively, if the observation O contains all information relevant to predicting the demonstration action A, then providing an additional observation should not improve our knowledge of A. Our goal is to construct a trajectory score $S(\tau)$ that reflects visual observability and can be used for demonstration curation.

A natural way to quantify visual observability is through the uncertainty of the demonstration action conditioned on the recorded observation. Let A denote the demonstration action and O the recorded observation being evaluated. The conditional entropy $H(A \mid O)$ characterizes the uncertainty of the action distribution given O. However, high conditional action uncertainty can result not only from missing sensory information, but also from multimodality in demonstrator behavior or action noise. Therefore, $H(A \mid O)$ alone cannot distinguish uncertainty caused by missing sensory information.

To determine whether O already contains the information needed to predict A, we introduce an auxiliary observation O_{aux} that provides additional information about the same transition. If O is fully sufficient for predicting A, then observing O_{aux} should not further change the conditional action distribution. Equivalently, the conditional action distribution should remain unchanged,

$$p(A \mid O, O_{\mathrm{aux}}) = p(A \mid O). \tag{3}$$

When this condition does not hold, the change in the conditional action distribution reflects action-relevant information that is available in O_{aux} but missing from O.

We quantify this change using the expected KL divergence between the two conditional action distributions,

$$\mathbb{E}_{O,O_{\mathrm{aux}}}\left[D_{\mathrm{KL}}\left(p(A \mid O, O_{\mathrm{aux}}) \,\|\, p(A \mid O)\right)\right]. \tag{4}$$

This quantity is precisely the conditional mutual information (CMI)

$$I(A; O_{\text{aux}} \mid O) = H(A \mid O) - H(A \mid O, O_{\text{aux}}). \tag{5}$$

The first term measures action uncertainty given the recorded observation, while the second measures the remaining uncertainty after incorporating the auxiliary observation. A larger CMI indicates that more action-relevant information is revealed by O_{aux} beyond O, corresponding to lower observability. We therefore define the observability score as the negative CMI, $-I(A; O_{\text{aux}} \mid O)$, so that higher scores indicate better observability. This estimate is relative to the chosen auxiliary observation, and therefore does not measure information that is absent from both O and O_{aux}.

Using the equivalent KL form of CMI in Eq. equation 4, we define an observability score for each transition as

$$s_t = -D_{\text{KL}}\left(p(A \mid o_t, o_{\text{aux},t}) \,\|\, p(A \mid o_t)\right). \tag{6}$$

The expectation of s_t over the demonstration distribution is $-I(A; O_{\text{aux}} \mid O)$.

4.2 Auxiliary Information for Visual Observability

The choice of auxiliary information determines which source of missing action-relevant information can be detected. Let V denote the visual observations available to the policy and R the non-visual observations, including robot proprioception. We consider two practical instantiations depending on the sensing available in the demonstration dataset.

Adding visual information. When the demonstration dataset contains an exterior camera view E that is not included in the policy observation, we use this view as auxiliary information and define the observability score using

$$-I(A; E \mid V, R). \tag{7}$$

Conditioning on (V, R) accounts for the observations available to the policy, while E provides additional visual information about the scene. A lower score indicates that the exterior view reveals more action-relevant information missing from the policy observations, for example because of occlusion, limited field of view, or viewpoint-dependent visibility.

Removing non-visual information. When exterior views are not available in datasets, we use the non-visual robot observations already recorded in the dataset as auxiliary information and define the observability score using

$$-I(A; R \mid V). \tag{8}$$

This formulation measures how much action-relevant information R provides beyond the recorded visual observations. A lower score indicates that more information relevant to the demonstration action is available from non-visual sensing but not captured by vision, providing a practical diagnostic of visual observability without requiring an additional camera.

Algorithm 1: FOCUS: Demonstration Curation

Require: Demonstrations $\mathscr{D} = \{\tau_i\}_{i=1}^{N}$, auxiliary source, tail fraction α, retention ratio ρ, Monte Carlo sample count M
Ensure: Selected demonstrations $\mathscr{D}_\rho$
if using an auxiliary exterior camera **then**
  Set $O = (V, R)$ and $O_{\text{aux}} = E$
else
  Set $O = V$ and $O_{\text{aux}} = R$
end if
Fit $q_{\text{base}}(A \mid O)$ and $q_{\text{aug}}(A \mid O, O_{\text{aux}})$ by demonstration-action negative log-likelihood
Select checkpoints by validation negative log-likelihood using shared episode-level splits
for each trajectory $\tau_i \in \mathscr{D}$ **do**
  for each transition $t = 1, \ldots, T_i$ **do**
    Sample $\tilde{a}^{(1)}, \ldots, \tilde{a}^{(M)}$ from $q_{\text{aug}}(\cdot \mid o_{i,t}, o_{\text{aux},i,t})$
    Estimate negative conditional KL:
$$s_{i,t} \leftarrow -\frac{1}{M} \sum_{m=1}^{M} \log \frac{q_{\text{aug}}(\tilde{a}^{(m)} \mid o_{i,t}, o_{\text{aux},i,t})}{q_{\text{base}}(\tilde{a}^{(m)} \mid o_{i,t})}$$
  end for
  Let $\mathscr{T}_i^{\alpha}$ index the lowest $\lceil \alpha T_i \rceil$ scores
  $S(\tau_i) \leftarrow \frac{1}{|\mathscr{T}_i^{\alpha}|} \sum_{t \in \mathscr{T}_i^{\alpha}} s_{i,t}$
end for
$\mathscr{D}_\rho \leftarrow$ the highest-scoring ρN trajectories, rounding the retained count to an integer
return $\mathscr{D}_\rho$

4.3 Observability-Based Trajectory Filtering

The estimators above provide an observability score s_t for each transition. We filter at the trajectory level rather than removing individual transitions for two reasons. First, transition-level estimates can be noisy, making individual-transition filtering unstable. Second, a low-observability transition may result from preceding behavior that leads the system into a poorly observed state, so removing only that transition retains the behavior that produced it. We therefore use the trajectory as the unit of data curation.

A trajectory-level score must also account for the fact that observability failures can be localized to short, critical phases such as contact or insertion. A simple temporal average can dilute these failures. Since lower s_t indicates lower observability, we aggregate the lower tail of the transition scores within each trajectory using α-CVaR. We use $\alpha \in (0,1]$ to denote the fraction of lowest-scoring transitions included in the tail. Let $\mathscr{T}_i^{\alpha}$ contain the indices of the lowest $\lceil \alpha T_i \rceil$ transition scores in trajectory τ_i. We define

$$S(\tau_i) = \frac{1}{|\mathscr{T}_i^{\alpha}|} \sum_{t \in \mathscr{T}_i^{\alpha}} s_{i,t}. \tag{9}$$

For example, $\alpha = 0.2$ averages the lowest-scoring 20% of transitions. This aggregation emphasizes phases with low observability while reducing sensitivity to individual noisy estimates. Given a retention ratio ρ, we apply Eq. 2 to retain the highest-scoring trajectories for downstream behavioral cloning.

Algorithm 1 summarizes density fitting, transition scoring, and trajectory selection for either auxiliary-information source.

4.4 Implementation Details

We fit three conditional action-density models: $q_V(A \mid V)$, $q_{VR}(A \mid V,R)$, and, when an additional exterior view is available, $q_{VRE}(A \mid V,R,E)$. Each model minimizes the negative log-likelihood of recorded demonstration actions, using fixed episode-level training and validation splits shared across model variants and random seeds. Here, V contains the wrist and base exterior RGB views, R contains joint positions, joint velocities, and gripper positions, and E is the additional exterior RGB view. All inputs correspond to a single timestep, and each model predicts a Gaussian mixture over seven arm commands and one continuous gripper command. We use ResNet-18 visual encoders and an MLP density head, with a separate checkpoint selected for each model by validation negative log-likelihood.

To instantiate Eq. equation 6, the exterior-camera score uses $-D_{\mathrm{KL}}(q_{VRE} \| q_{VR})$, whereas the proprioception score uses $-D_{\mathrm{KL}}(q_{VR} \| q_V)$, with each density conditioned on its corresponding inputs at the same timestep. We estimate each divergence by sampling actions from the density with augmented observations and averaging the log-density ratio. Architecture, preprocessing, and optimization details are provided in Appendix .1.

5 Experiments

We aim to answer four questions: (1) Does our observability score accurately identify demonstrations with insufficient visual observations? (2) Does observability-based data curation improve downstream imitation learning performance? (3) What does our observability score capture? (4) How does CVaR aggregation affect observability score quality?

5.1 Experimental Setup

We first introduce the task suite and annotations, then describe the policy training, comparisons, and evaluation protocol. We denote the variant using robot proprioception as auxiliary information by FOCUS-robot and the variant using an additional exterior camera by FOCUS-extra.

Tasks and Datasets. We evaluate our method on two simulated and four real-robot manipulation tasks. The simulated tasks are "Thread Needle" and "Insert Fork", implemented in robosuite (Zhu

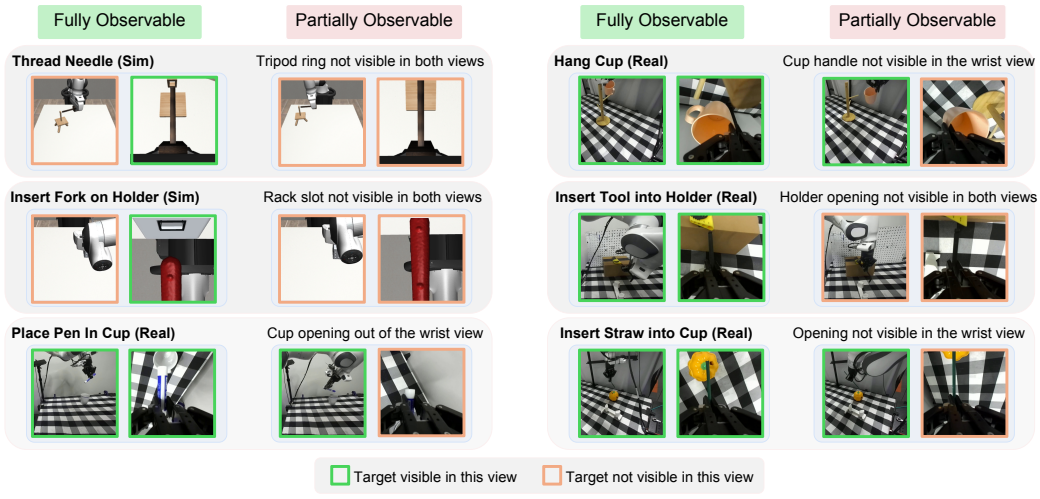


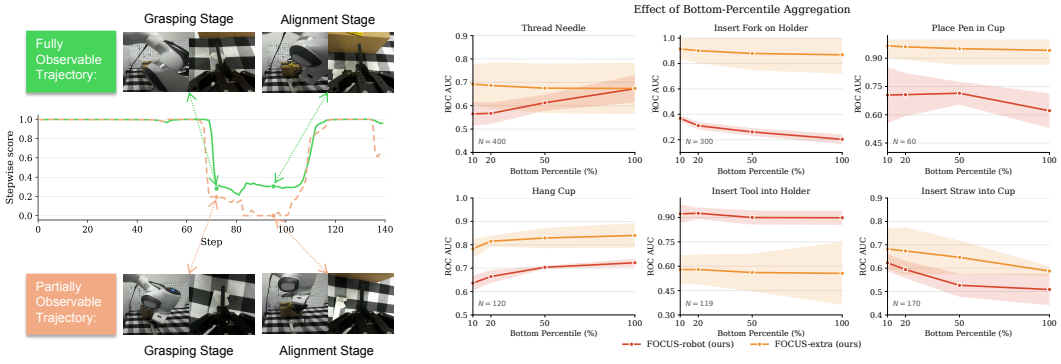
Figure 2: Tasks and observability annotations. Each row contrasts successful demonstrations whose approach configurations yield different visibility of the task-critical target at the marked interaction phase. A trajectory is fully observable when the target is clearly visible in either camera view, though the wrist view usually provides this information because the exterior camera is farther from the interaction. Green and orange outlines indicate whether the target is clearly visible or obscured in each view. Labels are used to evaluate learned observability scores, but are not inputs to our learned curation method.

et al., 2020). Thread Needle is adapted from the MimicGen Threading environment (Mandlekar et al., 2023), while Insert Fork is based on the fork insertion task from Play2Perfect (Lum et al., 2026). The real-robot tasks—"Place Pen in Cup", "Insert Straw into Cup", "Hang Cup", and "Fit Tool into Holder"—are performed on the DROID setup (Khazatsky et al., 2024) with a Franka Panda.

Observability Annotations. For each task, we collect successful demonstrations that vary in how well the task-relevant state is captured by the policy's camera observations. We manually annotate each trajectory as fully or partially observable based on whether the camera views clearly capture the information needed to infer the demonstrated action during task-critical phases. For example, partial observability may arise when the camera views do not clearly reveal a tool's alignment with a holder opening or the position of a cup handle relative to a peg. Figure 2 summarizes the observability criterion for each task and shows representative fully and partially observable demonstrations. These annotations provide an offline reference for evaluating whether our observability scores align with human judgments. We do not use them to train the scoring models or downstream policies. Detailed descriptions of the tasks, data collection, and annotation protocols are provided in Appendix .4 and Appendix .5.

Policy training. We fine-tune pretrained OpenPI $\pi_{0.5}$ policies (Intelligence et al., 2025) with LoRA adapters in the PaliGemma backbone and action expert, initializing from $\pi_{0.5}$-base. We use a high-capacity pretrained VLA to reduce potential confounding from limited model capacity in observability estimation, and train downstream policies on curated datasets to evaluate whether data selection improves policy performance. Each policy takes an exterior-camera image, a wrist-camera image, robot state, and a task-language instruction, and predicts 16-step action chunks with seven arm-joint commands and one gripper command. In DROID, demonstrations contain two exterior views and one wrist view; the policy uses one exterior view and the wrist view, leaving the second exterior view as auxiliary information for observability estimation. Detailed training settings are provided in Appendix .1.

Baselines and evaluation. At retention ratio ρ, a curation method retains a ρ fraction of a task's demonstrations for VLA fine-tuning; we evaluate $\rho \in \{40\%, 60\%, 80\%\}$. We compare observability-



(a) Timestep-level observability.

(b) Effect of CVaR aggregation.

Figure 3: **Analysis of observability scores.** (a) Scores along fully and partially observable demonstrations respond to both task-critical occlusions and less critical motion, helping explain disagreement with human trajectory-level annotations. (b) Varying the fraction of lowest-scoring timesteps used for trajectory aggregation shows that focusing on low-score phases improves ROC AUC on several tasks, while overly aggressive aggregation can emphasize incidental partial observability.

based curation against training on all demonstrations, random trajectory selection at matched retention ratios, DemInf (Hejna et al., 2025), which uses mutual information to estimate demonstration quality, and QoQ (Lee et al., 2026), which uses influence functions to estimate each demonstration's contribution to validation loss. Appendix .2 provides further baseline details. Our primary metric is closed-loop task success. We train one policy per curation condition and evaluate it with 100 rollouts on Thread Needle, 200 on Insert Fork, and 20 on each real-robot task. As a secondary offline diagnostic, we report ROC AUC against human annotations, averaging over three seeds for the MLP–GMM density models due to their sensitivity to initialization. Additional retention ratios and implementation details are provided in the appendix.

5.2 Does the Observability Score Identify Demonstrations with Insufficient Visual Observations?

We first evaluate whether our observability score correctly ranks demonstrations by how observable they are from the policy inputs (Q1). Figure 4 (top) progressively removes demonstrations according to each method and reports the average human-rated observability of the remaining dataset. A better ranking should remove poorly observable demonstrations earlier, causing the retained set to approach the human-label oracle, while random filtering provides a chance-level reference.

Across tasks, observability-based ranking generally improves the quality of the retained demonstrations. FOCUS-extra is typically more aligned with human annotations than FOCUS-robot and is generally stronger than DemInf and QoQ, which often remain closer to random filtering. This is expected because our human labels primarily reflect whether task-relevant visual information is available from the policy's camera observations. An additional camera view directly reveals information missing due to occlusion or viewpoint, whereas proprioception captures a different source of information that may affect action prediction without corresponding as directly to the visual observability labels.

Insert Tool into Holder is an exception, where FOCUS-robot outperforms FOCUS-extra. The exterior camera also does not clearly observe the hole on the holder, so it provides only limited additional information beyond the policy view. Despite this weaker auxiliary signal, FOCUS-extra remains comparable to DemInf and still outperforms random filtering. Overall, these results suggest that our score is most effective when the auxiliary observation exposes information that is genuinely missing from the policy input.

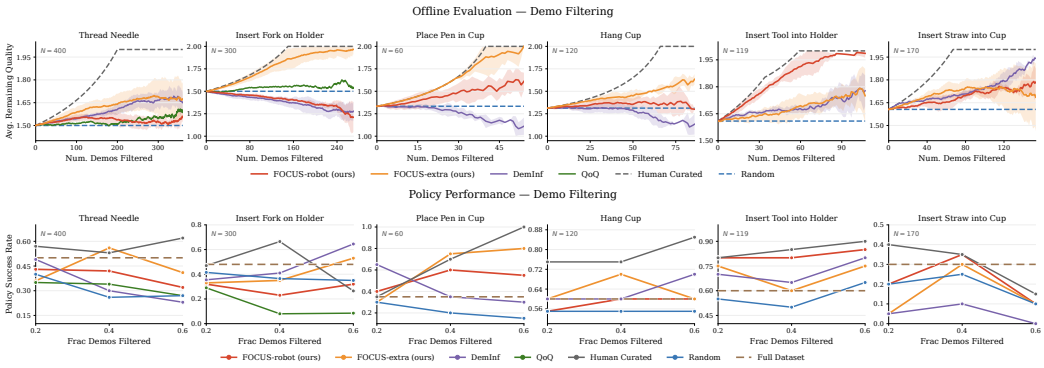


Figure 4: Offline observability ranking and downstream policy performance across six tasks. Top: mean human observability label among retained demonstrations as lower-ranked trajectories are removed. Bottom: closed-loop success after fine-tuning on selected demonstrations; the full-data line uses all demonstrations.

5.3 Does Observability-Based Data Curation Improve Downstream Imitation Learning?

We next evaluate whether selecting more observable demonstrations improves downstream imitation learning (Q2). Figure 4 (bottom) compares policies trained on demonstrations selected by FOCUS, DemInf, and random filtering with the full dataset, together with a human-curated reference. Observability-based curation improves performance on several tasks while using less data. On Place Pen in Cup, FOCUS-extra reaches 80% success using 40% of the demonstrations, compared with 35% for the full dataset and 15% for the matched random subset. On Fit Tool into Holder, FOCUS-robot reaches 85% success at the same retention ratio, compared with 60% and 65%, respectively. These matched-size comparisons show that the gains come from which demonstrations are retained, rather than simply from reducing dataset size. Since downstream performance also depends on factors beyond observability, such as data diversity and behavioral coverage, filtering solely for observability can eventually remove useful training examples. The similar degradation of the human-curated subset on Insert Straw into Cup further supports this interpretation.

5.4 What Does the Observability Score Capture?

We next qualitatively examine how the observability score evolves within a trajectory (Q3). Figure 3a compares a fully observable and a partially observable Insert Tool into Holder demonstration. The two trajectories are similar during initial free motion but diverge around grasping and alignment. Both scores decrease after grasping as the tool becomes partially occluded. During alignment, the fully observable trajectory recovers while the partially observable trajectory remains low. This example shows that the metric responds to both incidental and task-critical occlusion, whereas human annotations primarily emphasize the latter.

5.5 How Does CVaR Aggregation Affect Observability Score Quality?

Finally, we study how the CVaR aggregation percentile affects agreement with human observability labels (Q4). Figure 3b shows that focusing on lower-scoring timesteps improves ROC AUC on several tasks, especially Insert Fork, Place Pen in Cup, and Insert Straw into Cup. More aggressive aggregation can hurt on Thread Needle and Hang Cup, suggesting a tradeoff between detecting localized occlusion and emphasizing incidental low-score segments.

6 Conclusions

We introduced FOCUS, which uses auxiliary observations to assess visual observability and curate robot demonstrations. Across six tasks, its scores generally align with human judgments, and the best-performing configurations improve policy success by 5–45 percentage points over full-data

training. The approach depends on informative auxiliary observations, and aggressive filtering can reduce behavioral coverage. Combining observability with diversity and task relevance is a promising direction for future curation methods.

7 AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experiments, implement method, and support qualitative and thematic data analysis. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation, interpret results. Generating synthetic data sets and cleaning and reformating dataset are not applicable to this work. Additionally, we used generative AI tools for editing code and editing a research paper to improve readability. We have reviewed all AI-assisted work. Code generated or edited by LLMs was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

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.

Karl Johan Åström. Optimal control of markov processes with incomplete state information i. *Journal of mathematical analysis and applications*, 10:174–205, 1965.

Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Data quality in imitation learning. *Advances in neural information processing systems*, 36:80375–80395, 2023.

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

Qianzhong Chen, Justin Yu, Mac Schwager, Pieter Abbeel, Fred Shentu, and Philipp Wu. Sarm: Stage-aware reward modeling for long horizon robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 17091–17116, 2026.

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.

Kanishk Gandhi, Siddharth Karamcheti, Madeline Liao, and Dorsa Sadigh. Eliciting compatible demonstrations for multi-human imitation learning. In *Conference on Robot Learning*, pp. 1981–1991. PMLR, 2023.

Tanmay Gangwani, Joel Lehman, Qiang Liu, and Jian Peng. Learning belief representations for imitation learning in pomdps. In *uncertainty in artificial intelligence*, pp. 1061–1071. PMLR, 2020.

Nicolas Heess, Jonathan J Hunt, Timothy P Lillicrap, and David Silver. Memory-based control with recurrent neural networks. *arXiv preprint arXiv:1512.04455*, 2015.

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. *arXiv preprint arXiv:2502.08623*, 2025.

Edward Hu, Jie Wang, Xingfang Yuan, Fiona Luo, Muyao Li, Gaspard Lambrechts, Oleh Rybkin, and Dinesh Jayaraman. Real-world reinforcement learning of active perception behaviors. *Advances in Neural Information Processing Systems*, 38:117272–117308, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al.

\

pi_ {0.5} : *avision* − *language* − *actionmodelwithopen* − *worldgeneralization*. *arXiv preprint arXiv:2504.16054*, 2025.

Justin Kerr, Kush Hari, Ethan Weber, Chung Min Kim, Brent Yi, Tyler Bonnen, Ken Goldberg, and Angjoo Kanazawa. Eye, robot: Learning to look to act with a bc-rl perception-action loop. *arXiv preprint arXiv:2506.10968*, 2025.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Sachit Kuhar, Shuo Cheng, Shivang Chopra, Matthew Bronars, and Danfei Xu. Learning to discern: Imitating heterogeneous human demonstrations with preference and representation learning. In *Conference on Robot Learning*, pp. 1437–1449. PMLR, 2023.

Sateesh Kumar, Shivin Dass, Georgios Pavlakos, and Roberto Martín-Martín. Collage: Adaptive fusion-based retrieval for augmented policy learning. *arXiv preprint arXiv:2508.01131*, 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.

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.

Tyler Ga Wei Lum, Kushal Kedia, C Karen Liu, and Jeannette Bohg. Play2perfect: What matters in dexterous play pretraining for precise assembly? *arXiv preprint arXiv:2606.26428*, 2026.

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. *arXiv preprint arXiv:2108.03298*, 2021.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

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.

Nico Messikommer, Jiaxu Xing, Elie Aljalbout, and Davide Scaramuzza. Student-informed teacher training. In *International Conference on Learning Representations*, volume 2025, pp. 67308–67323, 2025.

Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. *arXiv preprint arXiv:2210.11435*, 2022.

Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent model-free rl can be a strong baseline for many pomdps. *arXiv preprint arXiv:2110.05038*, 2021.

Sreevardhan Sirigiri, Nathan Samuel de Lara, Christopher Agia, Florian Shkurti, and Fabio Ramos. Diversity you can actually measure: A fast, model-free diversity metric for robotics datasets. *arXiv preprint arXiv:2603.11634*, 2026.

Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

Gokul Swamy, Sanjiban Choudhury, J Bagnell, and Steven Wu. Sequence model imitation learning with unobserved contexts. *Advances in Neural Information Processing Systems*, 35:17665–17676, 2022.

Marcel Torne, Andy Tang, Yuejiang Liu, and Chelsea Finn. Learning long-context diffusion policies via past-token prediction. *arXiv preprint arXiv:2505.09561*, 2025.

Kuancheng Wang, Vaibhav Saxena, Shuo Cheng, Yotto Koga, and Danfei Xu. Learning to see while learning to act: Diffusion models for active perception in robot imitation. *arXiv preprint arXiv:2606.23625*, 2026.

Andrew Warrington, Jonathan W Lavington, Adam Scibior, Mark Schmidt, and Frank Wood. Robust asymmetric learning in pomdps. In *International Conference on Machine Learning*, pp. 11013–11023. PMLR, 2021.

Luca Weihs, Unnat Jain, Iou-Jen Liu, Jordi Salvador, Svetlana Lazebnik, Aniruddha Kembhavi, and Alex Schwing. Bridging the imitation gap by adaptive insubordination. *Advances in Neural Information Processing Systems*, 34:19134–19146, 2021.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. *Advances in Neural Information Processing Systems*, 33:2564–2575, 2020.

Changti Wu, Bin Yu, Zhaolong Shen, Shijie Lian, Xiaopeng Lin, Cong Huang, Zhirui Zhang, Lei Zhang, and Kai Chen. Sieve: Structure-aware data selection for imitation learning with vla models. *arXiv preprint arXiv:2607.06442*, 2026.

Amber Xie, Rahul Chand, Dorsa Sadigh, and Joey Hejna. Data retrieval with importance weights for few-shot imitation learning. *arXiv preprint arXiv:2509.01657*, 2025.

Haoyu Xiong, Xiaomeng Xu, Jimmy Wu, Yifan Hou, Jeannette Bohg, and Shuran Song. Vision in action: Learning active perception from human demonstrations. *arXiv preprint arXiv:2506.15666*, 2025.

Justin Yu, Andrew Goldberg, Kavish Kondap, Karim El-Refai, Ethan Ransing, Qianzhong Chen, Mac Schwager, Fred Shentu, Philipp Wu, and Ken Goldberg. Warp-rm: A warp-augmented relative progress reward model for data curation. *arXiv preprint arXiv:2606.28320*, 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.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

APPENDIX

.1 Experimental Details

We provide the dataset, scoring, and policy-training details underlying the experimental setup. The density-estimation and policy-training protocols use different model inputs and action targets, specified below.

Dataset populations. We evaluate observability estimates on six manipulation datasets: Thread Needle and Insert Fork in simulation, and Place Pen in Cup, Insert Straw into Cup, Hang Cup, and Fit Tool into Holder on the real robot. These datasets contain 400, 300, 60, 170, 120, and 208 demonstrations, respectively, with recorded observability annotations. The paper uses binary fully/partially observable labels. The additional-camera evaluation for Fit Tool into Holder uses the 119 demonstrations containing that view, with both conditional density models trained on the same camera-eligible population. For downstream imitation learning, we use robosuite Thread Needle

with RoboMimic-format demonstrations and locally collected data on a Franka Panda using the DROID setup (Khazatsky et al., 2024). The documented real-robot policy runs cover Fit Tool into Holder, Hang Tool, Hang Cup, Place Pen in Cup, and Insert Straw into Cup.

Scorer and policy observations. The visual observation V consists of a wrist RGB image and a base exterior RGB image. We evaluate two sensing settings from Sec. 4.2. With an additional exterior camera, we compare action densities conditioned on (V,R) and (V,R,E). Without this additional view, we compare densities conditioned on V and (V,R), using proprioception to assess visual observability. For density estimation, R includes joint positions, measured joint velocities, and gripper positions. Downstream policies receive the wrist and base exterior images, joint positions, gripper position, and a task instruction; the additional exterior view is used only for scoring.

Density-model architecture. Each RGB view is resized to 224×224, matching the downstream VLA image input size, and encoded by a ResNet-18 trained from scratch with GroupNorm and SpatialSoftmax pooling into a 64-dimensional feature. GroupNorm avoids dependence on batch statistics and the discrepancy between training-time batch statistics and evaluation-time running statistics in BatchNorm. We concatenate the visual features with the proprioceptive inputs available to the model and process them with a two-layer MLP with 512 hidden units per layer; the vision-only model receives no proprioceptive inputs. The output is a five-component Gaussian mixture with diagonal component covariances over a single eight-dimensional action: seven arm commands and one continuous gripper command. Component means use a tanh activation, and component standard deviations are parameterized as $\mathrm{softplus}(s) + 0.01$.

Density-model training. We train all conditional density models by minimizing the negative log-likelihood of recorded demonstration actions, with fixed episode-level training and validation splits shared across model variants and random seeds. Arm targets are relative joint positions for Thread Needle and Insert Fork and recorded joint velocities for the real-robot datasets. Action preprocessing is fixed across compared models, with normalization statistics computed from training episodes only. We optimize all network parameters using Adam with batch size 128 and a constant learning rate of 10^{-4} for 200 epochs, without weight decay, random cropping, or dropout. We evaluate negative log-likelihood over the complete validation split after every epoch and independently select the checkpoint with the lowest validation negative log-likelihood for each model.

Score computation. Negative conditional KL scores are estimated using 512 independent action samples from the density with augmented observations. We smooth scores over overlapping ten-step windows before averaging the lowest-scoring fraction within each episode.

Policy training and data selection. We fine-tune a separate pretrained OpenPI $\pi_{0.5}$ policy for each task and selected subset using LoRA adapters in the PaliGemma backbone and action expert, with ranks 16 and 32, respectively. Policies receive an exterior-camera image, a wrist-camera image, robot state, and a task-language instruction; they predict 16-step action chunks containing seven arm-joint commands and one gripper-position command. We initialize every policy from $\pi_{0.5}$-base except Place Pen in Cup, which is initialized from $\pi_{0.5}$-DROID. The standard recipe optimizes the flow-matching imitation objective with AdamW for 20,000 updates at batch size 32. Policies use 224×224 images. Joint-position policies predict arm targets relative to the current joint configuration; the Place Pen in Cup velocity policy uses $\pi_{0.5}$-DROID. Gripper targets remain absolute. We compare full-data training with whole-demonstration subsets selected randomly, by our scores, and by DemInf (Hejna et al., 2025). Within each comparison, initialization, policy inputs, action representation, normalization, and optimizer-update budget are held fixed. Training samples shuffled frames from retained episodes, so smaller subsets are revisited more often.

Evaluation. We assess observability ranking using ROC AUC against binary human observability labels, with higher scores corresponding to fully observable trajectories. Human labels are not used by our learned curation method; the separate human-curated Fork reference uses them to select subsets (Appendix .4). Downstream performance is measured by task success in closed-loop policy rollouts. In simulation, policies are evaluated from fresh seeded resets. We distinguish the density-fitting and validation populations when reporting ranking metrics: validation loss is used for checkpoint selection, and combined-population scores include fitting data.

.2 Curation Baselines

DemInf. DemInf (Hejna et al., 2025) learns low-dimensional representations of state-action chunks and estimates their mutual information nonparametrically. It averages transition-level estimates to obtain a trajectory-level demonstration-quality score for ranking and selection.

QoQ. QoQ (Lee et al., 2026) uses influence functions to estimate each training state-action pair's contribution to reducing validation loss. It aggregates these estimates within trajectories to rank demonstrations for selection. We apply DemInf and QoQ at the same retention ratios as our method.

.3 Retention Ratios and Matched Subsets

For a dataset containing N demonstrations, a retention ratio ρ specifies a subset of ρN trajectories, rounded to a whole number of demonstrations. We evaluate $\rho \in \{40\%, 60\%, 80\%\}$, with the full-data condition retaining all demonstrations. At each ratio, our method, DemInf, and QoQ rank the complete task dataset and retain the same number of top-ranked trajectories. The random baseline retains the same number of trajectories sampled without using any curation score. Thus, comparisons at a fixed retention ratio control for training-set size and isolate the effect of which demonstrations are retained.

.4 Simulation Task and Fork Evaluation Details

Thread Needle builds on the Threading environment in MimicGen (Mandlekar et al., 2023). We detail the Fork adaptation, data conditions, and evaluation protocol. These settings apply to the Fork policy comparison and should not be assumed for other tasks.

Task adaptation and demonstrations. Insert Fork is our robosuite adaptation of the fork-insertion task in Play2Perfect (Lum et al., 2026), using a fork from the YCB object set (**?**). We replace the original embodiment with a Franka Emika Panda and parallel-jaw gripper, adapting the grasp and contact geometry, rack aperture, and rack placement. The robot grasps the fork from a table, transports it to the rack, aligns it with the aperture, and inserts it. The dataset contains 300 successful demonstrations. We sample grasp locations from two continuous ranges, producing 150 demonstrations with the rack aperture visible in the wrist camera at a predefined pre-insertion checkpoint and 150 without that visibility. These groups differ in grasp configuration as well as wrist-camera visibility.

Reference data conditions. ALL uses all 300 demonstrations, and OBSERVABLE uses the 150 demonstrations with wrist-camera aperture visibility. Human-curated subsets use a fixed ordering with all observable demonstrations first. Retaining 40%, 60%, or 80% gives, respectively, 120 observable demonstrations; 150 observable and 30 partial demonstrations; and 150 observable and 90 partial demonstrations. Size-matched random controls use prefixes of a single permutation of all 300 demonstrations generated with seed 1 without labels. Their sizes are 120, 180, and 240, and the random subsets are nested. These label-based references are distinct from our automatic curation method.

Policy training. We fine-tune the pretrained OpenPI $\pi_{0.5}$-base checkpoint with LoRA in the PaliGemma backbone and action expert. The policy receives exterior and wrist images and robot state and predicts 16-step chunks of seven joint commands and one gripper command. Stored absolute joint targets are converted to offsets from the current observed joint configuration; the gripper target is unchanged. At inference, predicted offsets are converted back to absolute joint targets. All data conditions use identical initialization, preprocessing, and optimization: 20,000 updates, batch size 32, training seed 42, and a cosine learning-rate schedule with 1,000 warmup steps, peak 5×10^{-5}, and final value 5×10^{-6}. Fixed update budgets cause smaller subsets to be revisited more often.

Success criteria. We score a Thread Needle rollout as successful if the needle is threaded through the target opening. We score an Insert Fork rollout as successful if the fork is inserted into the rack

and its final pose lies within 1.8 cm and 20° of the target pose. All other outcomes are scored as failures.

Rollout evaluation. We evaluate the final checkpoint at 20,000 updates with a replanning horizon of 12 in randomized scenes. We use 100 rollouts for Thread Needle and 200 for Insert Fork. For Fork, every data condition uses the same 200 initial states from a separately generated reset manifest, with identical state IDs and ordering. We report the fraction of successful rollouts. Comparisons are paired by initial state, but each condition has only one training seed; rollout variability does not quantify variability across independently trained policies.

.5 Real-Robot Task Details

Success criteria. We use the Franka robot setup from DROID (Khazatsky et al., 2024). We score a Fit Tool into Holder trial as successful if the robot places the tool in the holder and it remains there without falling out. We score Place Pen in Cup as successful if the pen ends in the cup. We score Insert Straw into Cup as successful if the straw is inserted into its target container (the pumpkin) and remains in place. We score Hang Cup as successful if the cup hangs securely on the stand. All other outcomes are scored as failures.