DATA THAT SELECTS ITSELF: SCALABLE SELF-CURATION FOR ROBOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot imitation learning relies on demonstrations of varying quality, making effective data curation essential for scalable and data-efficient policy learning. However, learning a useful curation signal remains challenging when trusted reference demonstrations and policy-rollout feedback are unavailable. We present **Trajectory Progress Efficiency Consensus (TPEC)**, a self-supervised framework that learns to rank demonstrations from small, uncurated reference sets sampled directly from the candidate dataset. Using lightweight progress predictors, TPEC aggregates whole-trajectory efficiency scores into a consensus ranking for action-chunk selection, enabling data curation **without curated references or online policy rollouts**. Experiments demonstrate the effectiveness of TPEC across demonstration-quality regimes and policy scales. On RoboTwin 2.0 and in real-robot experiments, TPEC matches or exceeds full-data joint fine-tuning of the billion-parameter VLA model π_0 using only 50% of the available action chunks. Furthermore, TPEC scales to large-scale pretraining, highlighting its potential for practical robot learning pipelines. Enabled by its in-dataset selection strategy, TPEC achieves curation-preparation speedups of $2{,}257\times$ over the gradient-based CUPID and $4.53\times$ over the progress-reward-based WARP-RM in VLA data curation. Code will be public.

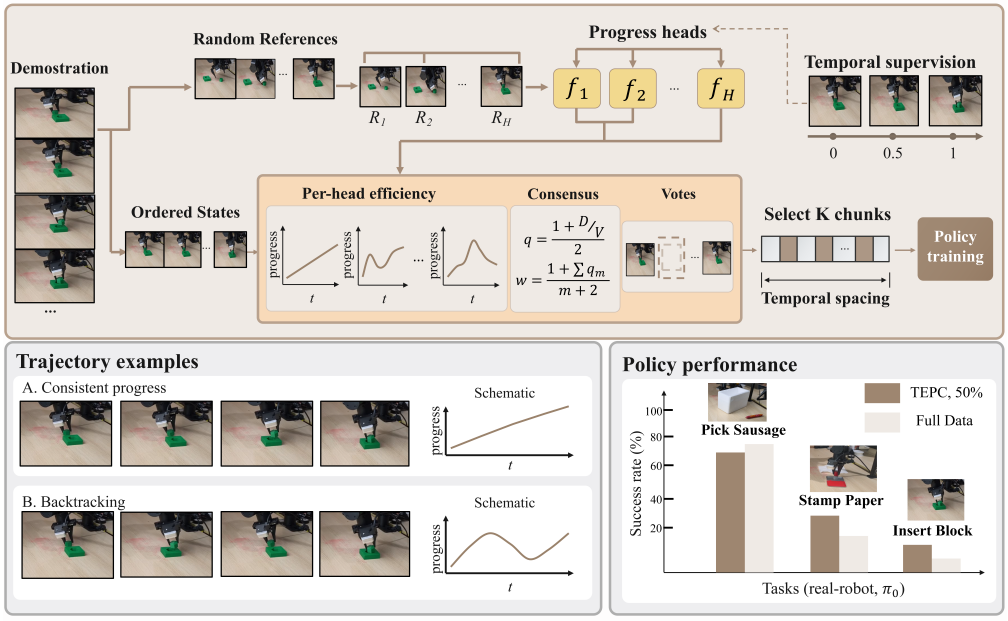

Figure 1: **Overview of TPEC.** TPEC learns progress from randomly sampled demonstrations and aggregates trajectory-level progress-efficiency scores to select action chunks for policy training. **Bottom left:** Trajectories with consistent progress and backtracking, alongside schematic progress curves. **Bottom right:** Policy success rates at different percentages of retained action chunks.

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.

1 INTRODUCTION

Robot imitation learning relies on demonstrations that vary in operator proficiency, execution quality, and task coverage (Mandlekar et al., 2022; Belkhale et al., 2023). This dependence spans action-chunking and diffusion policies (Zhao et al., 2023; Chi et al., 2023), as well as transformer-based and vision–language–action models (Brohan et al., 2023; Zitkovich et al., 2023; Ghosh et al., 2024; Kim et al., 2025; Black et al., 2025). Growing demonstration collections (Walke et al., 2023; O'Neill et al., 2024; Khazatsky et al., 2024) make practical selection signals increasingly valuable, particularly when trusted references and policy-rollout feedback are unavailable.

Data curation has been studied through mixture optimization, semantic deduplication, and learned filtering in language and vision (Xie et al., 2023; Abbas et al., 2023; Fang et al., 2024). Robotics methods use domain composition and state–action statistics (Hejna et al., 2025a;b), reference similarity (Du et al., 2023; Lin et al., 2025; Memmel et al., 2025), or policy-based utility estimates (Dass et al., 2026; Kumar et al., 2025). Constructing these signals can incur substantial preparation costs. Reference-guided retrieval and valuation rely on target-task demonstrations, with DataMIL and COLLAGE additionally requiring policy training (Dass et al., 2026; Kumar et al., 2025). DemoSCORE, CUPID, and ATHENA require policy training and evaluation rollouts (Chen et al., 2025; Agia et al., 2025; Xu et al., 2026), while CUPID further incurs the computational and memory costs of per-sample gradients and curvature approximations (Agia et al., 2025; Xu et al., 2026).

Reward and progress models offer alternative sources of supervision (Brown et al., 2019; 2020; Ma et al., 2023b;a; Zhang et al., 2025; Chen et al., 2026). SCIZOR combines self-supervised progress estimation with deduplication, while WARP-RM learns relative progress from time-warped successful demonstrations (Zhang et al., 2026; Yu et al., 2026). These progress-based approaches support curation without policy execution, although scorer preparation may still require additional supervision or reference selection: SARM uses subtask annotations, and WARP-RM's folding experiments use demonstrations preselected by duration (Chen et al., 2026; Yu et al., 2026). With uncurated references, timestamps provide inexpensive supervision but imperfect progress targets, since pauses and retries need not advance the task (Yu et al., 2026). Differences in predictor calibration and scoring trajectories used for predictor fitting further complicate ranking.

We present **Trajectory Progress Efficiency Consensus (TPEC)**, a self-supervised framework that uses the consistency of predicted progress along a trajectory to rank demonstrations. Lightweight predictors learn normalized timestamps from disjoint random reference subsets. For each predictor, whole-trajectory progress efficiency is the ratio of net predicted progress to total predicted variation. The resulting score measures directional consistency and is invariant to positive affine transformations of the predictions. TPEC aggregates scores only from predictors not trained on the evaluated trajectory and uses the resulting trajectory ranking to guide action-chunk selection under a fixed filter ratio with temporal suppression. Building on pretrained representations for data-efficient learning (Nair et al., 2023), the visual implementation combines frozen features (Oquab et al., 2024) with joint-command inputs and trains only compact prediction heads.

Our contributions are threefold:

1. **Curation from uncurated references.** We introduce a self-supervised framework that constructs curation signals from small random subsets of the candidate dataset, without quality-based preselection or online policy rollouts.

2. **Whole-trajectory efficiency consensus.** We aggregate whole-trajectory efficiency scores from lightweight progress predictors to guide filter-ratio-based action-chunk selection. Each score is invariant to positive affine transformations, and aggregation excludes in-sample predictions.

3. **Data efficiency and scalability.** On RoboTwin 2.0 and in real-robot experiments, TPEC matches or exceeds full-data joint fine-tuning of π_0 using only 50% of the available action chunks. TPEC also scales to large-scale pretraining and achieves curation-preparation speedups of $2{,}257\times$ over CUPID and $4.53\times$ over WARP-RM in VLA data curation.

2 Related Work

Data curation and reference-guided retrieval. Data quality and composition affect imitation-learning performance (Mandlekar et al., 2022; Belkhale et al., 2023). Data-mixture optimization (Xie et al., 2023; Hejna et al., 2025a), semantic deduplication (Abbas et al., 2023), and learned filtering (Fang et al., 2024) offer complementary approaches to improving training data. In robotics, DemInf scores trajectories using state–action mutual information (Hejna et al., 2025b). Behavior Retrieval, FlowRetrieval, and STRAP retrieve relevant data through state–action embeddings, motion representations, and visually aligned sub-trajectories, respectively (Du et al., 2023; Lin et al., 2025; Memmel et al., 2025). These retrieval methods use target-task demonstrations as queries. TPEC instead learns trajectory-quality scores from small, randomly sampled reference sets within the candidate pool, without quality-based reference preselection.

Policy-dependent data valuation. Influence functions, datamodels, and scalable attribution estimators relate training examples to model behavior (Koh & Liang, 2017; Ilyas et al., 2022; Park et al., 2023). In robotics, Demo-SCORE learns from successful and failed policy rollouts (Chen et al., 2025), while CUPID uses influence functions to attribute policy return to demonstrations (Agia et al., 2025). ATHENA improves computational efficiency and task balance, extending influence-based curation to billion-parameter multitask VLAs (Xu et al., 2026). DataMIL replaces rollout evaluation with target-task validation losses, estimating influence through subset-specific policy training or metagradients (Dass et al., 2026). COLLAGE weights retrieved subsets by target-demonstration likelihood under policies trained on those subsets (Kumar et al., 2025). Both therefore retain policy-training costs despite avoiding rollout-based scoring. TPEC confines scorer training to compact progress predictors, without task-policy training, rollout evaluation, or policy-based attribution, supporting curation for VLA fine-tuning and large-scale pretraining.

Reward learning and progress-based curation. T-REX and D-REX learn rewards from provided or automatically generated demonstration rankings (Brown et al., 2019; 2020). VIP, LIV, and ReWiND learn visual or language-conditioned reward signals from videos (Ma et al., 2023b;a; Zhang et al., 2025). SARM uses subtask annotations for stage-aware progress (Chen et al., 2026). SCIZOR combines self-supervised progress estimation with deduplication and aggregates local and trajectory-level information (Zhang et al., 2026), while WARP-RM learns relative progress from time-warped successful demonstrations to filter and reweight action chunks (Yu et al., 2026). TPEC combines training on disjoint, uncurated reference subsets with whole-trajectory efficiency scoring based on net predicted progress relative to total predicted variation. Its scores are invariant to positive affine transformations and aggregated only from predictors not fitted on the evaluated trajectory, an exclusion principle related to out-of-bag valuation (Kwon & Zou, 2023). The consensus trajectory scores are then assigned to action chunks for filter-ratio-based selection.

3 Background and Problem Formulation

3.1 Imitation Learning and Filter-Ratio Curation

Let $\mathcal{D} = \{\tau_i\}_{i=1}^{n}$ be a demonstration dataset, where $\tau_i = \{(x_{i,t}, a_{i,t})\}_{t=0}^{T_i-1}$ and $T_i \geq 2$. Here, $x_{i,t}$ denotes the available observation and conditioning information, and $a_{i,t}$ is the demonstrated action. Demonstrations need not be equally useful for learning, even when they reach the same task outcome (Mandlekar et al., 2022; Belkhale et al., 2023). For action-chunk policies, let $\mathcal{C}(\mathcal{D})$ contain valid training examples $c_{i,t} = (x_{i,t}, a_{i,t:t+L_c-1})$ with chunk length L_c (Zhao et al., 2023; Chi et al., 2023). Training on a nonempty subset $S \subseteq \mathcal{C}(\mathcal{D})$ optimizes the empirical imitation objective

$$\mathcal{L}_{\mathrm{IL}}(\theta; S) = \frac{1}{|S|} \sum_{c \in S} \ell_{\mathrm{IL}}(\theta; c), \tag{1}$$

where ℓ_{IL} is the loss for the chosen policy class, including regression, diffusion, or flow-matching objectives (Chi et al., 2023; Black et al., 2025). Let $\widehat{\theta}(S) = \mathcal{A}(S)$ denote the parameters returned by a fixed training procedure, including its initialization and optimization schedule. This formulation covers both pretraining and fine-tuning.

Given a selected-set size of K chunks, the ideal curation objective is

$$S^\star \in \arg\max_{\substack{S\subseteq\mathcal{C}(\mathcal{D})\\|S|=K}} J\left(\pi_{\widehat{\theta}(S)}\right), \tag{2}$$

where $J(\pi)=\mathbb{E}_{\xi\sim\pi}[R(\xi)]$ measures closed-loop performance under a fixed evaluation distribution. With a binary completion reward, J is the success rate. Curation therefore targets downstream policy utility rather than imitation loss alone (Agia et al., 2025; Dass et al., 2026). Direct subset comparison requires repeated policy training and evaluation, motivating surrogate selection signals that can be constructed at substantially lower cost. Here, J specifies the curation objective but is unavailable as feedback during selection.

3.2 Temporal Supervision with Uncurated References

We consider curation without quality labels, trusted reference demonstrations, or policy-rollout feedback. A small reference set $\mathcal{R}\subset\mathcal{D}$, with $|\mathcal{R}|=M\ll n$, is sampled uniformly without replacement and without quality-based preselection. Scorer-fitting examples come from $\mathcal{R}$, while the resulting scorer may be applied to the entire candidate pool. Reference trajectories are not assumed to be successful or optimal.

Normalized timestamps provide self-supervised targets for temporal prediction (Zhang et al., 2026; Yu et al., 2026). For $y_{i,t}=t/(T_i-1)$, the unrestricted population-optimal squared-loss predictor satisfies

$$f^\star(x)=\arg\min_{z\in\mathbb{R}}\mathbb{E}[(z-Y)^2\mid X=x]=\mathbb{E}[Y\mid X=x], \tag{3}$$

where (X,Y) follows the reference-state sampling distribution. This predictor estimates a temporal coordinate, not task completion or demonstration utility: pauses, retries, and unsuccessful attempts can advance elapsed time without advancing the task. With limited references, predictions may also reflect trajectory-specific timing rather than transferable structure. The challenge is to derive useful trajectory assessments from these imperfect temporal predictions at low preparation cost, without task-policy training or execution.

4 TPEC: Curating robot data with data itself

TPEC selects K action chunks from demonstrations whose quality is unknown, favoring trajectories that advance consistently through task stages. It first learns temporal coordinates from observations, then scores each trajectory for forward progress relative to backtracking. Multiple predictors combine their scores, and a final selection step spreads the selected chunks across time.

4.1 Learning a transferable temporal coordinate

We need a coordinate that indicates which stage an observation resembles. Normalized timestamps supply targets from zero at the start to one at the end of each reference trajectory (Zhang et al., 2026; Yu et al., 2026). A predictor learns these targets from observations alone. Assuming observable task stages share an ordering across demonstrations, returning to an earlier stage can lower the predicted coordinate even as elapsed time increases.

We partition the random reference set $\mathcal{R}$ into $H\geq 2$ nonempty disjoint subsets $\mathcal{R}_h$. Each compact predictor $f_h(x)=g_{\theta_h}(z(x))$ maps observation features $z(x)$ to a scalar coordinate. With $\mathcal{B}_h$ denoting the observation slots sampled from its subset, head h minimizes timestamp prediction error:

$$\widehat{\theta}_h\in\arg\min_{\theta_h}\frac{1}{|\mathcal{B}_h|}\sum_{(i,t)\in\mathcal{B}_h}\left(g_{\theta_h}(z(x_{i,t}))-\frac{t}{T_i-1}\right)^2. \tag{4}$$

For low-dimensional observations, $z(x)$ contains standardized states. The visual head combines one global and four regional tokens from frozen DINOv2 features (Oquab et al., 2024) with a joint and gripper state token in a two-layer Transformer. A learned readout token gathers these cues to predict the temporal coordinate. Only the heads are trained, and inputs exclude timestamps and trajectory identifiers. Appendix C.2 gives the token layout and architecture details.

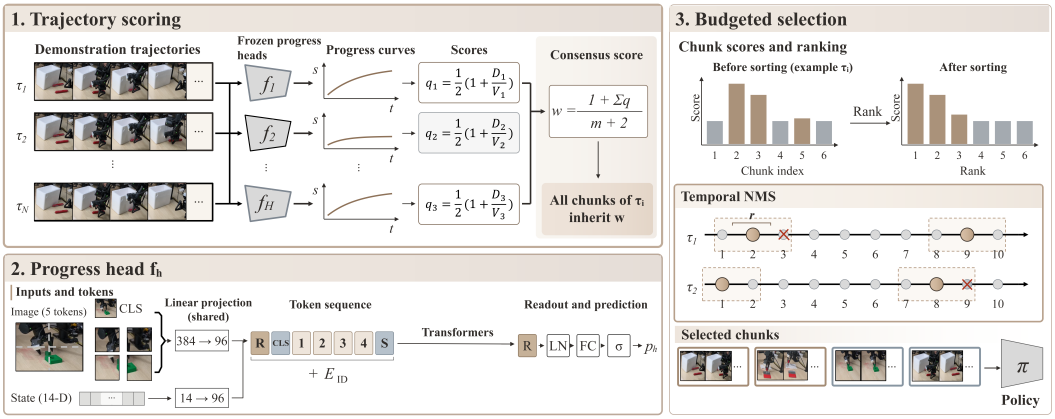

Figure 2: **TPEC architecture.** a) Trajectory scoring. Frozen progress heads map ordered observations to progress curves. Their efficiency scores are aggregated into a consensus score, excluding heads trained on the evaluated trajectory. b) Progress head. Five visual tokens, a state token, and a learned readout token pass through a two-layer Transformer. The readout token is used to predict scalar progress. c) Filter-ratio selection. Chunks inherit their trajectory's consensus score. Score ranking and temporal non-maximum suppression (NMS) select K chunks for policy training, suppressing nearby chunk starts within each trajectory.

We check transfer to the remaining references $\mathcal{R} \setminus \mathcal{R}_h$ using mean squared error (MSE), E_h. A head is retained if $E_h < E_{0,h}$, where $E_{0,h}$ is the error of always predicting 0.5. Sampling, calibration averaging, and additional timestamp-shuffle checks appear in Appendix C.

4.2 Measuring progress efficiency

Two trajectories can start and end at the same predicted stages but differ in how often they move backward. We therefore compare the endpoint change with all movement along the predicted coordinate. All heads query the same L_i states of τ_i in time order, including both endpoints, yielding $p_{h,i,j} = f_h(x_{i,t_{i,j}})$. Net advancement $D_{h,i}$, total variation $V_{h,i}$, and efficiency $q_{h,i}$ are

$$D_{h,i} = p_{h,i,L_i} - p_{h,i,1}, \qquad V_{h,i} = \sum_{j=1}^{L_i-1} |p_{h,i,j+1} - p_{h,i,j}|,$$
$$q_{h,i} = \begin{cases} \dfrac{1}{2}\left(1 + \dfrac{D_{h,i}}{V_{h,i}}\right), & V_{h,i} > 0, \\ \dfrac{1}{2}, & V_{h,i} = 0. \end{cases} \tag{5}$$

Here V counts both forward and backward movement. Equivalently, let $P = \sum_j (\Delta p_j)_+$ and $B = \sum_j (-\Delta p_j)_+$ sum positive and negative changes, with $\Delta p_j = p_{j+1} - p_j$ and $(u)_+ = \max(u, 0)$. Then $q = P/(P+B)$ for $V > 0$: efficiency is the fraction of predicted movement that goes forward. For example, $(0, 0.5, 1)$ has $q = 1$, whereas $(0, 1, 0.5, 1)$ has $q = 0.75$ despite identical endpoints. A constant prediction receives the neutral score 0.5.

Since $V = D + 2B$, more backtracking lowers q at fixed $D > 0$, while repeated predictions leave it unchanged. Positive rescaling and shifts also leave efficiency unchanged: $q(ap + b) = q(p)$ for $a > 0$. This lets the consensus compare progress efficiency across heads without first aligning the numerical scales and offsets of their predicted coordinates. Path-shape stability and sampling effects are analyzed in Appendix C.3.

4.3 Trajectory-excluded efficiency consensus

A head trained on a trajectory may reproduce its increasing timestamps without learning transferable stages. When scoring τ_i, we therefore exclude every head fitted on it and combine the remaining

judgments into a consensus (Breiman, 1996). Let $\mathcal{H}_i$ contain the m_i eligible heads:

$$\mathcal{H}_i = \{h \mid E_h < E_{0,h},\ \tau_i \notin \mathcal{R}_h\}, \qquad m_i = |\mathcal{H}_i|,$$
$$w_i = \frac{1 + \sum_{h \in \mathcal{H}_i} q_{h,i}}{m_i + 2}. \tag{6}$$

We compute efficiency separately for each head because averaging coordinate paths first can smooth away reversals. The formula for w_i averages the observed scores with two neutral votes of 0.5. Scores with fewer eligible votes are pulled more strongly toward 0.5, with $w_i = 0.5$ when none qualify. Combining heads can improve ranking when their errors are complementary. Proposition 2 and Appendix C.3 give the conditions and discuss correlated errors and unequal vote counts.

4.4 Filter-ratio selection as distribution concentration

Each chunk inherits its source trajectory's score w_i. We process chunks in descending score order, breaking ties randomly with a fixed seed. To avoid concentrating the selected set on neighboring windows, temporal non-maximum suppression (NMS) skips a chunk if its start lies within r time steps of a selected start in the same trajectory. Thus it accepts $c_{i,t}$ only if

$$|t - s| > r \qquad \text{for all already selected } c_{i,s} \in S. \tag{7}$$

Selection stops at K chunks. This spacing encourages temporal coverage while permitting chunk overlap. Radius and relaxation details appear in Appendix C. Policy training retains Eq. (1) with equal weight on each selected chunk, giving the training distribution $\widehat{P}_S = K^{-1} \sum_{c \in S} \delta_c$, where δ_c places unit mass on c.

To understand how selection changes training data, assume the selected trajectories have positive net advancement in an underlying task-progress coordinate. Their movement can be decomposed into net-forward progress and round trips, each pairing a backward movement with an equal forward movement. Let $q_i^\star$ be efficiency in this ideal coordinate. If $n_i(S)$ chunks come from trajectory i, its share of the selection is $\rho_i = n_i(S)/K$. An analytical reference distribution R_S samples within each trajectory in proportion to progress variation, then mixes trajectories using ρ_i. It decomposes as

$$R_S = (1 - \beta_S) G_S + \beta_S U_S,$$
$$\beta_S = 2\left(1 - \sum_i \rho_i q_i^\star\right). \tag{8}$$

Here G_S and U_S contain the net-forward and round-trip components, and β_S is the round-trip share. Higher ideal efficiency reduces this share. Because R_S weights progress movement whereas $\widehat{P}_S$ weights selected chunks equally, the two distributions can differ, for example when a trajectory contains long pauses.

To connect this decomposition to learning, write $g_P(\theta)$ for the average imitation-loss gradient under distribution P, which summarizes its training signal. Let $P_\star$ describe the behaviors needed for the task, including useful recovery actions. Over a parameter region Θ, let η_S be the largest gradient discrepancy between the actual selection $\widehat{P}_S$ and the reference R_S, and δ_S that between G_S and the target $P_\star$. If each chunk's gradient norm is at most C_g on Θ,

$$\sup_{\theta \in \Theta} \left\| g_{\widehat{P}_S}(\theta) - g_{P_\star}(\theta) \right\| \le \eta_S + (1 - \beta_S)\delta_S + 2C_g\beta_S. \tag{9}$$

The three terms bound the effects of sampling mismatch, target-behavior mismatch in the net-forward component, and round trips, respectively.

Proof sketch. In the ideal coordinate, a path with net advancement D_i and backward variation B_i has round-trip fraction $2B_i/(D_i + 2B_i) = 2(1 - q_i^\star)$. Mixing these components gives Eq. (8). Gradient linearity and the triangle inequality yield Eq. (9).

This gives a conditional reason to favor efficient trajectories: they reduce the round-trip term, while useful learning signals also depend on accurate scores and coverage of needed behaviors. Appendix C.4 provides the full derivation and discusses pauses and NMS.

Computational cost. Scoring uses $H \sum_i L_i$ head evaluations and sorting costs $O(N_c \log N_c)$ for $N_c = |\mathcal{C}(\mathcal{D})|$, beyond feature extraction, head fitting, and NMS. Features, heads, and scores are reusable across filter ratios. For fixed scorer inputs and architecture, preparation is independent of policy size and requires no policy gradients or rollouts.

5 Experiments

5.1 Experimental Setup

We evaluate TPEC on the RoboMimic (Mandlekar et al., 2022) and RoboTwin (Mu et al., 2025) simulation benchmarks, a 50-task multitask setting with 2,500 demonstrations fine-tuned with π_0 (Black et al., 2025), and three real-robot manipulation tasks. Single-task experiments use Diffusion Policy students (Chi et al., 2023); the multitask evaluation covers clean and domain-randomized conditions. We compare against Random, Random-NMS, CUPID, SS, DemInf, SCIZOR, WARP, ATHENA, and finally Full-data controls. Implementation, filter-ratio schedules, training protocols, and evaluation details are provided in Appendices A and B.

5.2 Simulation Benchmarks: RoboMimic and RoboTwin

We evaluate TPEC with task-specific students under matched architectures, selected-window counts, and training schedules. Figure 3 summarizes RoboMimic and RoboTwin on a common filter-ratio axis, where each percentage denotes the filter ratio, defined as 100% minus the retained percentage.

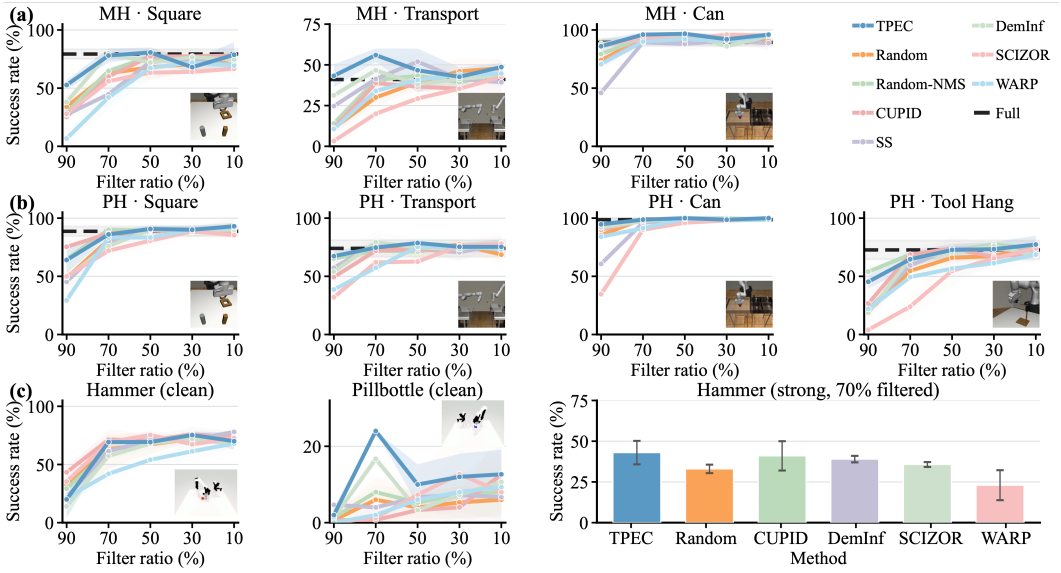

Figure 3: **Single-task curation across simulation benchmarks.** (a) RoboMimic MH tasks, (b) RoboMimic PH tasks, and (c) RoboTwin Beat Block Hammer task with clean data, Move Pillbottle Pad with clean data, and Beat Block Hammer with strongly perturbed data

The clearest gains occur at aggressive filtering. At a 90% filter ratio on RoboMimic MH, TPEC improves over Random by 18.7, 32.0, and 12.0 points on Square, Transport, and Can; the average gain across PH tasks is 15.0 points. Methods converge as retained subsets overlap, while Random-NMS remains competitive when temporal deduplication is already an effective inductive bias. On RoboTwin, TPEC reaches 69.3% on Hammer and 24.0% on Pillbottle at a 70% filter ratio, exceeding Random by 6.0 and 18.0 points. Under strong Hammer perturbations, it reaches 43.0% versus 33.0% for Random. The same score therefore transfers across observation modalities and demonstration-quality partitions, with its largest value appearing when the training set is most selective.

5.3 Multitask VLA Fine-tuning on 2,500 Demonstrations

We jointly fine-tune π_0 policies on 2,500 demonstrations spanning 50 tasks. Figure 4 summarizes the clean, domain-randomized, and averaged evaluations. The 50% filter-ratio setting retains half of the action chunks and reaches 48.56% clean and 17.15% DR success, exceeding Full by 5.44 and 2.05 points and Random by 4.46 and 0.93 points. Its mean across the two conditions is 32.86%, versus 29.11% for Full and 30.33% for ATHENA. At a 30% filter ratio, TPEC reaches 48.54% clean and 18.17% DR success; at 70%, it preserves a clean-scene advantage while sacrificing DR performance. The results show that an internally learned ranking can improve joint VLA training while removing half of the candidate action chunks.

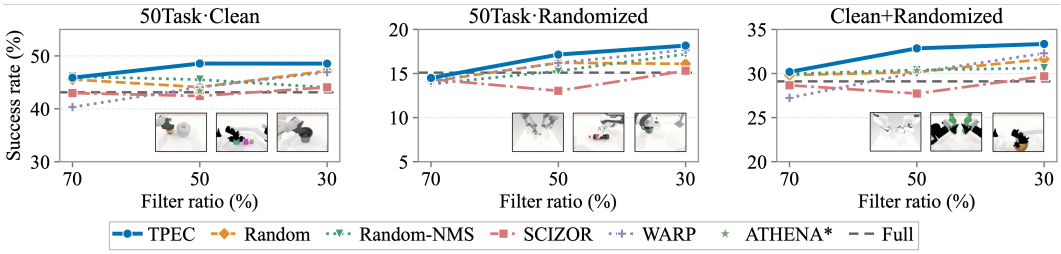

Figure 4: **Multitask VLA fine-tuning with selected data.** Success rates for π_0 on 50 tasks under clean, domain-randomized, and averaged evaluation

5.4 Real-robot Evaluation

Real-robot evaluation. To further evaluate TPEC, we conduct experiments across three real-robot desktop manipulation tasks using 300 demonstrations collected at 25 Hz over 2.9 hours, with 100 demonstrations per task. The suite spans a simple sausage-picking task, a long-horizon paper-stamping task, and a precision block-insertion task. We then compare π_0 trained on the 50% filter-ratio subset selected by TPEC with π_0 trained on all demonstrations.

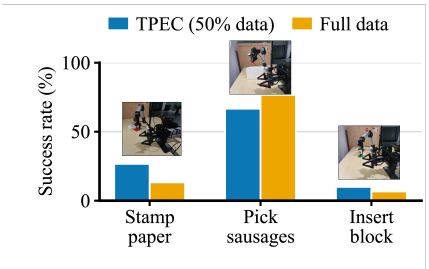

Figure 5: **Real-robot success across three tasks.**

As shown in Figure 5, TPEC50 reaches 26.7%, 66.7%, and 10.0% success rate on Stamp paper, Pick sausages, and Insert block, respectively, while Full π_0 reaches 13.3%, 76.7%, and 6.7%. The mean success is 34.5% for TPEC50 and 32.2% for Full π_0; TPEC50 wins on two tasks and remains close on the third. Thus, the progress-efficiency signal preserves useful action chunks under real-world execution noise while halving the training data.

From fine-tuning to pretraining. Beyond the multitask fine-tuning experiment, TPEC also performs well as a lightweight curation step for large-scale pretraining. This indicates that the in-dataset progress-efficiency signal transfers beyond a fixed task pool and can be reused in practical robot-learning data pipelines.

5.5 Ablations and Computational Cost

Whole-trajectory scoring and consensus make distinct contributions. With student settings and selected-window counts fixed, TPEC exceeds one-vote and net-progress variants across all ten Square/Transport comparisons, by 5.08 and 10.07 points on average (Figure 6, left). Its 24.41-point gain over 16-step local efficiency supports whole-trajectory scoring, while reduced-vote variants isolate the role of consensus. Shortest-trajectory selection remains competitive because duration captures useful structure, although it does not model direction.

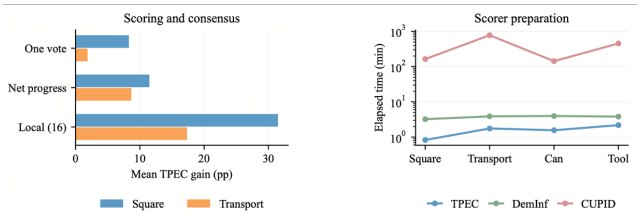

Figure 6: **Effective and inexpensive curation.**

Useful references need not be preselected for quality. Reference-panel experiments show that mixed- and low-quality internal references can both support useful policies; no source dominates across all repeats. Fresh-scene evaluation favors TPEC over one vote on Square and Transport, while shortest-trajectory selection remains strong.

Scoring adds a small preparation overhead. TPEC requires 0.83–2.19 minutes of elapsed preparation per PH task, compared with 142.79–776.78 for CUPID and 3.23–3.97 for DemInf (Figure 6, bottom). Scores and scorer artifacts are reused across five filter ratios and three student seeds. In the Can batch measurement, preparation accounts for approximately 1% of the TPEC pipeline's elapsed time. This low measured overhead complements the policy-independent scoring design. Timing includes queueing; measurement boundaries and resource conventions are specified in Appendix A.

6 Conclusion

We introduced TPEC, a self-supervised framework for curating robot demonstrations using only a small, randomly sampled reference set from the candidate data. By training compact progress predictors on disjoint reference subsets, measuring directional progress efficiency with a calibration-invariant score, and excluding in-sample predictions, TPEC constructs a selection signal without quality annotations, curated references, or online policy rollouts. Across RoboMimic, RoboTwin, and the reported multitask π_0 setting, the results show that mixed-quality demonstrations can provide a useful signal for selecting their own training subset. Performance varies across tasks and filter ratios, and strong heuristics such as shortest-trajectory selection remain competitive in some conditions. Overall, these findings position uncurated internal references as a practical alternative to selection methods that require a clean reference set or policy-derived feedback.

7 Limitation

First, our evaluation scope does not establish robustness to hardware noise, safety-critical failures, or shifts in camera, embodiment, and operator behavior. Second, normalized time is only a proxy for task progress. Retries, pauses, cyclic motions, and successful trajectories with different strategies can violate its assumptions, while frozen visual features and joint commands may omit task-relevant signals such as force or tactile feedback. Third, TPEC does not eliminate model or filter-ratio selection. Its advantage varies with the task, reference composition, and filter ratio. Training multiple progress heads also introduces preparation cost, even though that cost is small in the measured settings.

REFERENCES

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. SemDeDup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023. URL https://arxiv.org/abs/2303.09540.

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.

Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Data Quality in Imitation Learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 80375–80395. Curran Associates, Inc., 2023. doi: 10.52202/075280-3524. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/fe692980c5d9732cf153ce27947653a7-Abstract-Conference.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.010. URL https://roboticsproceedings.org/rss21/p010.html.

Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996. doi: 10.1007/BF00058655. URL https://link.springer.com/article/10.1007/BF00058655.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, et al. RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://roboticsproceedings.org/rss19/p025.html.

Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating Beyond Suboptimal Demonstrations via Inverse Reinforcement Learning from Observations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 783–792. PMLR, 2019. URL https://proceedings.mlr.press/v97/brown19a.html.

Daniel S. Brown, Wonjoon Goo, and Scott Niekum. Better-than-Demonstrator Imitation Learning via Automatically-Ranked Demonstrations. In *Proceedings of the Conference on Robot Learning*, volume 100, pp. 330–359. PMLR, 2020. URL https://proceedings.mlr.press/v100/brown20a.html.

Annie S. Chen, Alec M. Lessing, Yuejiang Liu, and Chelsea Finn. Curating Demonstrations using Online Experience. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.071. URL https://roboticsproceedings.org/rss21/p071.html.

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. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1c92edb990a05f2269f0cc3afbb4c952-Paper-Conference.pdf.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, et al. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://roboticsproceedings.org/rss19/p026.html.

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. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/033d9e8dbbbc0ad90e59222cf1db0fc2-Paper-Conference.pdf.

Maximilian Du, Suraj Nair, Dorsa Sadigh, and Chelsea Finn. Behavior Retrieval: Few-shot imitation learning by querying unlabeled datasets. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.011. URL https://roboticsproceedings.org/rss19/p011.html.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data Filtering Networks. In *International Conference on Learning Representations*, volume 2024, pp. 36221–36237, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/9bf0810a4a1597a36d27ceea58667d92-Paper-Conference.pdf.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, et al. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://roboticsproceedings.org/rss20/p090.html.

Joey Hejna, Chethan Anand Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. ReMix: Optimizing Data Mixtures for Large Scale Imitation Learning. In *Proceedings of The 8th Conference on Robot Learning*, volume 270, pp. 145–164. PMLR, 2025a. URL https://proceedings.mlr.press/v270/hejna25a.html.

Joey Hejna, Suvir Mirchandani, Ashwin Balakrishna, Annie Xie, Ayzaan Wahid, Jonathan Tompson, et al. Robot Data Curation with Mutual Information Estimators. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.023. URL https://roboticsproceedings.org/rss21/p023.html.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 9525–9587. PMLR, 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, et al. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.120. URL https://roboticsproceedings.org/rss20/p120.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, et al. OpenVLA: An Open-Source Vision-Language-Action Model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270, pp. 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1885–1894. PMLR, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Sateesh Kumar, Shivin Dass, Georgios Pavlakos, and Roberto Martín-Martín. COLLAGE: Adaptive fusion-based retrieval for augmented policy learning. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4607–4624. PMLR, 2025. URL https://proceedings.mlr.press/v305/kumar25a.html.

Yongchan Kwon and James Zou. Data-OOB: Out-of-bag Estimate as a Simple and Efficient Data Value. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 18135–18152. PMLR, 2023. URL https://proceedings.mlr.press/v202/kwon23e.html.

Li-Heng Lin, Yuchen Cui, Amber Xie, Tianyu Hua, and Dorsa Sadigh. FlowRetrieval: Flow-guided data retrieval for few-shot imitation learning. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4084–4099. PMLR, 2025. URL https://proceedings.mlr.press/v270/lin25a.html.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-image representations and rewards for robotic control. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 23301–23320. PMLR, 2023a. URL https://proceedings.mlr.press/v202/ma23b.html.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=YJ7o2wetJ2.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, et al. What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164, pp. 1678–1690. PMLR, 2022. URL https://proceedings.mlr.press/v164/mandlekar22a.html.

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. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a06e129e01e0d2ef853e9ff67b911360-Paper-Conference.pdf.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, et al. RoboTwin: Dual-Arm Robot Benchmark with Generative Digital Twins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27649–27660, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Mu_RoboTwin_Dual-Arm_Robot_Benchmark_with_Generative_Digital_Twins_CVPR_2025_paper.html.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A Universal Visual Representation for Robot Manipulation. In *Proceedings of The 6th Conference on Robot Learning*, volume 205, pp. 892–909. PMLR, 2023. URL https://proceedings.mlr.press/v205/nair23a.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, et al. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Gupta, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://doi.org/10.1109/ICRA57147.2024.10611477.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 27074–27113. PMLR, 2023. URL https://proceedings.mlr.press/v202/park23c.html.

Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, et al. BridgeData V2: A Dataset for Robot Learning at Scale. In *Proceedings of The 7th Conference on Robot Learning*, volume 229, pp. 1723–1736. PMLR, 2023. URL https://proceedings.mlr.press/v229/walke23a.html.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, et al. DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 36, pp. 69798–69818. Curran Associates, Inc., 2023. doi: 10.52202/075280-3059. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/dcba6be91359358c2355cd920da3fcbd-Abstract-Conference.html.

Tao Xu, Jiaxin Wang, Runhao Zhang, Jiayi Guan, Xianchao Zeng, Weixi Song, Xinyu Zhou, Zhetao Chen, Guang Chen, and Yong-Lu Li. ATHENA: Accelerated Multi-Task Heterogeneous Influence Functions for Robot Data Curation, 2026. URL https://arxiv.org/abs/2606.16208.

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. URL https://arxiv.org/abs/2606.28320.

Jiahui Zhang, Yusen Luo, Abrar Anwar, Sumedh Anand Sontakke, Joseph J Lim, Jesse Thomason, et al. ReWiND: Language-Guided Rewards Teach Robot Policies without New Demonstrations. In *Proceedings of The 9th Conference on Robot Learning*, volume 305, pp. 460–488. PMLR, 2025. URL https://proceedings.mlr.press/v305/zhang25a.html.

Yu Zhang, Yuqi Xie, Huihan Liu, Rutav Shah, Michael Wan, Linxi Fan, et al. SCIZOR: A Self-Supervised Approach to Data Curation for Large-Scale Imitation Learning. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026. URL https://arxiv.org/abs/2505.22626.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.016. URL https://roboticsproceedings.org/rss19/p016.html.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A RoboMimic Protocols and Complete Results

A.1 Datasets and budget conventions

Each MH task uses 288 candidate demonstrations and 12 for validation. The candidate pool combines the original 192 training demonstrations with 96 additional demonstrations. All 96 additions are candidates, not test data. Budgets still use the original training-window count as their denominator, despite selection drawing from the expanded pool.

PH uses 180 training trajectories and 20 validation trajectories. Its candidate-window counts are 25,905, 82,719, 19,623, and 85,076 for Square, Transport, Can, and Tool Hang, respectively. Each PH budget retains the stated fraction K/N of candidate windows.

In the source tables, $b = 100K/N_{\text{ref}}$ denotes the nominal retained percentage. Figures instead show the complementary filter ratio, $100 - b$, or the percentage removed. Thus source columns 10/30/50/70/90 correspond to plotted filter ratios 90/70/50/30/10. For MH, both conventions use the original training-window denominator rather than the expanded candidate-window count.

Table 1: MH candidate windows, original budget denominator, and selected window counts.

Task	N	Denominator	K_{10}	K_{30}	K_{50}
Square	75,129	49,996	5,000	14,999	24,998
Transport	185,942	125,324	12,532	37,597	62,662
Can	58,136	39,283	3,928	11,785	19,642

A.2 Training and evaluation protocols

MH. MH TPEC uses low-dimensional inputs and 48 internal references arranged in eight groups of six trajectories. Each head has two 128-dimensional GELU hidden layers and a sigmoid output. Reference candidates receive seven out-of-sample votes, while all other candidates receive eight. Trajectory scores are broadcast to their windows, and the student is trained with the standard Diffusion Policy loss.

Single-task evaluation. Each final EMA policy is tested for 50 episodes per seed, using seeds 24/25/26. Error bars show sample standard deviations. The main MH scenes were used during development. Fresh-scene results are reported separately below.

Two reported means do not match the stated granularity of three seeds with 50 episodes per seed: Square PH TPEC at 90% (93.00 ± 3.00) and Can MH Random-NMS at 90% (95.67 ± 4.16). The tables retain both source values. Their stated granularity does not support inferring per-seed counts or statistical significance.

PH. PH uses 4,051/2,951/2,551/2,111/1,751 epochs on Square and Can, and 3,051/2,501/2,301/1,261/1,001 on Transport and Tool Hang, as the budget increases. Methods are matched within each budget. Full follows the 90% schedule but retains more windows, so its update count differs. Comparisons across budgets and against Full therefore use different numbers of training updates.

A.3 Method versions in the budget tables

The tables shorten method names by omitting the NMS suffix, except for Random-NMS. The shortened names do not imply that temporal suppression is absent from those methods.

For Square and Transport, the 10/30/50% columns use the original main-comparison versions. TPEC, CUPID, SS, and DemInf use radius one. Random-NMS follows its historical schedule. At 90%, Random-NMS uses radius one, while DemInf uses relative actions, fixed 1,024-sample KSG estimation, and clipping before aggregation. The unified tables retain these budget-specific versions.

Can uses version r2 for TPEC, DemInf, SCIZOR, and WARP, and version r3 for CUPID, SS, and Random-NMS. DemInf retains its older 8/2/1 radius schedule. At high budgets, suppression can relax to radius zero to fill the exact K.

SCIZOR-DINOv2 and WARP-RM-DINOv2 are fixed-budget adaptations whose modalities, reference sizes, and scorer capacities do not fully match TPEC. The SCIZOR adaptation omits some original trajectory-score mixing settings. The WARP adaptation selects windows without the full WARP-BC continuous loss weighting. Their results compare these pipelines as implemented, rather than isolating the scoring formulas.

A.4 MH: UNIFIED BUDGET COMPARISONS

The MH tables cover nominal budgets 10–90% for each task. At 70%, Square and Transport use seeds 24/25/26, with 50 rollouts per seed. Transport Full averages 22/50 and 19/50 successes across two seeds (41.0%), so it is shown without a three-seed error band. Later-budget results stay in separate cells, and independent training runs are not pooled.

Table 2: Square MH success rate (%). Percentages denote nominal budgets, not candidate-pool retention. Full: 79.3 ± 4.2. The 70% results use three training seeds (24/25/26) with 50 rollouts per seed. The 90% column uses the updated baseline versions specified in the protocol.

Method	10%	30%	50%	70%	90%
TPEC	52.7 ± 4.2	78.0 ± 6.0	80.7 ± 4.2	68.00 ± 6.00	78.67 ± 11.02
Random	34.0 ± 9.2	62.7 ± 3.1	68.0 ± 10.6	70.67 ± 6.43	75.33 ± 5.77
Random-NMS	29.3 ± 6.4	64.7 ± 3.1	77.3 ± 5.0	72.00 ± 11.14	70.67 ± 3.06
CUPID-NMS	25.3 ± 3.1	60.0 ± 0.0	77.3 ± 6.1	77.33 ± 6.11	78.00 ± 4.00
SS-NMS	27.3 ± 1.2	44.7 ± 10.3	72.7 ± 4.2	75.33 ± 3.06	69.33 ± 6.43
DemInf-NMS	38.0 ± 5.3	80.0 ± 4.0	72.0 ± 2.0	76.67 ± 3.06	74.67 ± 7.57
SCIZOR-DINOv2	28.0 ± 7.2	56.0 ± 2.0	63.3 ± 8.3	64.00 ± 5.29	66.67 ± 4.16
WARP-RM-DINOv2	6.7 ± 3.1	42.0 ± 10.4	67.3 ± 9.2	72.00 ± 3.46	69.33 ± 2.31

Table 3: Transport MH success rate (%). Percentages denote nominal budgets, not candidate-pool retention. Full: 41.0%, a two-seed mean (22/50 and 19/50). The 70% results use three training seeds (24/25/26) with 50 rollouts per seed. The 90% column uses the updated baseline versions specified in the protocol.

Method	10%	30%	50%	70%	90%
TPEC	43.3 ± 7.0	56.0 ± 4.0	46.7 ± 13.3	42.67 ± 5.77	48.67 ± 2.31
Random	11.3 ± 3.1	30.0 ± 5.3	39.3 ± 4.6	46.00 ± 8.00	48.00 ± 5.29
Random-NMS	14.0 ± 2.0	40.7 ± 3.1	43.3 ± 3.1	39.33 ± 4.16	44.67 ± 5.03
CUPID-NMS	11.3 ± 6.4	38.7 ± 3.1	36.7 ± 3.1	35.33 ± 2.31	42.67 ± 9.45
SS-NMS	24.7 ± 4.2	41.3 ± 11.0	52.0 ± 5.3	41.33 ± 3.06	39.33 ± 8.33
DemInf-NMS	31.3 ± 1.2	46.7 ± 8.3	38.0 ± 3.5	41.33 ± 5.77	43.33 ± 5.03
SCIZOR-DINOv2	3.3 ± 3.1	20.0 ± 2.0	29.3 ± 6.1	35.33 ± 8.08	42.67 ± 12.06
WARP-RM-DINOv2	10.7 ± 1.2	34.0 ± 10.6	40.7 ± 2.3	44.00 ± 2.00	44.67 ± 8.08

Table 4: Can MH success rate (%). All available budgets are shown together; percentages denote nominal budgets, not candidate-pool retention. Full: 89.33 ± 3.06.

Method	10%	30%	50%	70%	90%
TPEC	86.0 ± 6.0	96.0 ± 2.0	96.7 ± 1.2	92.00 ± 4.00	96.00 ± 0.00
Random	74.0 ± 2.0	92.7 ± 1.2	91.3 ± 3.1	94.00 ± 2.00	90.67 ± 8.08
Random-NMS	79.3 ± 9.0	93.3 ± 3.1	93.3 ± 2.3	93.33 ± 1.15	95.67 ± 4.16
CUPID	71.3 ± 7.0	92.0 ± 0.0	96.7 ± 1.2	92.67 ± 1.15	94.67 ± 1.15
SS	46.0 ± 2.0	89.3 ± 1.2	88.0 ± 5.3	90.67 ± 5.03	88.67 ± 4.16
DemInf	88.0 ± 3.5	92.0 ± 2.0	96.7 ± 2.3	86.00 ± 3.46	94.67 ± 2.31
SCIZOR-DINOv2	86.0 ± 5.3	93.3 ± 5.0	92.0 ± 6.0	96.00 ± 5.29	95.33 ± 4.16
WARP-RM-DINOv2	70.7 ± 3.1	90.0 ± 5.3	92.0 ± 3.5	90.67 ± 1.15	94.67 ± 3.06

A.5 PH: FIVE-BUDGET COMPARISONS

The PH tables report five budgets and a Full-data control for each task.

Table 5: Square PH success rate (%). Budgets are fractions of candidate windows. Full: 88.67 ± 4.16.

Method	10%	30%	50%	70%	90%
TPEC	64.00 ± 11.14	86.00 ± 2.00	90.67 ± 3.06	90.00 ± 3.46	93.00 ± 3.00
Random	48.67 ± 2.31	81.33 ± 1.15	90.67 ± 3.06	89.33 ± 3.06	93.33 ± 6.43
Random-NMS	64.67 ± 6.43	90.00 ± 3.46	89.33 ± 1.15	88.00 ± 4.00	90.00 ± 2.00
CUPID	75.33 ± 4.16	86.67 ± 4.62	89.33 ± 3.06	88.67 ± 5.03	87.33 ± 2.31
SS	45.33 ± 4.16	80.00 ± 4.00	90.67 ± 4.16	90.00 ± 0.00	90.67 ± 3.06
DemInf	49.33 ± 10.07	76.00 ± 6.93	90.00 ± 2.00	90.00 ± 5.29	90.00 ± 2.00
SCIZOR-DINOv2	50.00 ± 5.29	72.00 ± 3.46	80.67 ± 7.02	88.67 ± 4.16	85.33 ± 3.06
WARP-RM-DINOv2	29.33 ± 6.11	83.33 ± 2.31	83.33 ± 4.62	90.00 ± 3.46	92.00 ± 7.21

Table 6: Transport PH success rate (%). Budgets are fractions of candidate windows. Full: 74.00 ± 7.21.

Method	10%	30%	50%	70%	90%
TPEC	67.33 ± 4.16	74.67 ± 3.06	78.67 ± 1.15	75.33 ± 3.06	75.33 ± 6.43
Random	57.33 ± 7.02	78.00 ± 2.00	72.00 ± 7.21	77.33 ± 4.16	68.67 ± 7.02
Random-NMS	65.33 ± 8.08	72.00 ± 5.29	71.33 ± 4.62	76.00 ± 3.46	78.00 ± 7.21
CUPID	49.33 ± 8.08	72.00 ± 8.72	73.33 ± 3.06	70.67 ± 7.57	77.33 ± 4.16
SS	57.33 ± 10.07	78.00 ± 5.29	75.33 ± 5.03	70.67 ± 7.02	78.00 ± 3.46
DemInf	54.00 ± 3.46	79.33 ± 10.26	74.67 ± 7.57	73.33 ± 9.02	78.67 ± 2.31
SCIZOR-DINOv2	32.00 ± 5.29	62.00 ± 0.00	62.67 ± 3.06	76.00 ± 0.00	78.00 ± 12.17
WARP-RM-DINOv2	38.67 ± 8.33	57.33 ± 2.31	75.33 ± 9.45	75.33 ± 7.57	73.33 ± 4.16

Table 7: Can PH success rate (%). Budgets are fractions of candidate windows. Full: 98.67 ± 1.15.

Method	10%	30%	50%	70%	90%
TPEC	94.67 ± 3.06	98.67 ± 2.31	100.00 ± 0.00	98.67 ± 1.15	100.00 ± 0.00
Random	85.33 ± 3.06	99.33 ± 1.15	98.67 ± 1.15	99.33 ± 1.15	99.33 ± 1.15
Random-NMS	92.00 ± 2.00	98.67 ± 2.31	98.67 ± 1.15	98.00 ± 2.00	98.67 ± 2.31
CUPID	90.00 ± 3.46	98.00 ± 2.00	99.33 ± 1.15	100.00 ± 0.00	98.67 ± 1.15
SS	60.67 ± 4.16	96.00 ± 0.00	98.67 ± 1.15	98.67 ± 1.15	99.33 ± 1.15
DemInf	92.00 ± 3.46	97.33 ± 3.06	100.00 ± 0.00	100.00 ± 0.00	98.67 ± 1.15
SCIZOR-DINOv2	34.67 ± 7.02	90.00 ± 2.00	96.00 ± 0.00	98.00 ± 3.46	99.33 ± 1.15
WARP-RM-DINOv2	84.00 ± 2.00	91.33 ± 5.03	99.33 ± 1.15	98.67 ± 1.15	98.67 ± 1.15

Table 8: Tool Hang PH success rate (%). Budgets are fractions of candidate windows. Full: 72.67 ± 8.08.

Method	10%	30%	50%	70%	90%
TPEC	45.33 ± 5.77	64.67 ± 7.02	72.67 ± 5.03	73.33 ± 6.11	77.33 ± 7.57
Random	20.00 ± 2.00	54.67 ± 8.08	66.00 ± 5.29	67.33 ± 3.06	76.67 ± 4.16
Random-NMS	54.00 ± 4.00	68.67 ± 4.16	72.00 ± 5.29	77.33 ± 2.31	74.00 ± 5.29
CUPID	26.67 ± 11.72	68.67 ± 3.06	74.00 ± 8.00	66.67 ± 4.62	70.00 ± 5.29
SS	21.33 ± 8.08	59.33 ± 5.03	74.00 ± 11.14	73.33 ± 3.06	72.00 ± 5.29
DemInf	18.67 ± 3.06	65.33 ± 4.16	70.67 ± 4.16	76.00 ± 8.72	68.00 ± 8.00
SCIZOR-DINOv2	4.00 ± 2.00	24.00 ± 10.58	54.67 ± 12.22	65.33 ± 4.62	74.00 ± 9.17
WARP-RM-DINOv2	22.00 ± 5.29	49.33 ± 1.15	56.67 ± 7.57	61.33 ± 5.03	68.67 ± 5.03

Tool Hang Full reaches $72.67 \pm 8.08\%$, based on seed counts of 32/50, 37/50, and 40/50.

A.6 Component ablations across budgets

Within each comparison, the ablation tables hold the selected window count, student settings, and suppression protocol fixed. Reduced-vote variants reuse the eight-head cache. Self/OOB comparisons change both the contributing head and whether in-sample predictions are included.

Can's ablation uses student-seed tie-breaking and scenes that differ from the main comparison, which uses fixed selection seed 424242. Its full-TPEC row therefore does not replace the main-table result. At budget 50, additional single-policy counts are self7: 45/50 and 48/50, and one vote: 48/50. These counts are separate from the three-seed summary cells.

Table 9: Square MH component ablation (success %). All five nominal budgets are presented together. The full-TPEC row belongs to this ablation protocol.

Method	10%	30%	50%	70%	90%
Full TPEC	52.7 ± 4.2	78.0 ± 6.0	80.7 ± 4.2	78.00 ± 6.00	78.67 ± 11.02
Seven votes (OOB)	47.3 ± 7.0	74.7 ± 7.6	78.7 ± 2.3	77.33 ± 9.45	77.33 ± 3.06
Seven votes (self)	52.0 ± 3.5	74.7 ± 2.3	74.0 ± 8.0	77.33 ± 9.45	70.67 ± 7.02
One vote	38.0 ± 5.3	65.3 ± 3.1	71.3 ± 3.1	76.00 ± 13.11	76.00 ± 0.00
Two votes	40.0 ± 4.0	72.0 ± 5.3	70.7 ± 11.7	74.67 ± 8.33	78.00 ± 5.29
Four votes	38.7 ± 13.3	73.3 ± 1.2	76.0 ± 2.0	75.33 ± 3.06	79.33 ± 6.11
Shuffled time	40.7 ± 6.4	62.0 ± 3.5	69.3 ± 8.1	79.33 ± 8.33	78.00 ± 5.29
Shortest trajectory	60.7 ± 11.4	70.0 ± 0.0	78.0 ± 5.3	75.33 ± 3.06	76.67 ± 2.31
Net progress	28.0 ± 4.0	59.3 ± 3.1	78.7 ± 7.6	74.67 ± 9.45	70.00 ± 12.17
Time MSE	34.0 ± 2.0	70.0 ± 5.3	80.7 ± 4.2	76.00 ± 4.00	78.67 ± 9.87
Local efficiency (16)	2.0 ± 0.0	15.3 ± 6.4	50.0 ± 6.0	69.33 ± 12.70	74.00 ± 5.29

Table 10: Transport MH component ablation (success %). All five nominal budgets are presented together. The full-TPEC row belongs to this ablation protocol.

Method	10%	30%	50%	70%	90%
Full TPEC	43.3 ± 7.0	56.0 ± 4.0	46.7 ± 13.3	52.67 ± 5.77	50.00 ± 0.00
Seven votes (OOB)	57.3 ± 2.3	53.3 ± 4.2	40.0 ± 5.3	39.33 ± 7.02	44.00 ± 6.00
Seven votes (self)	52.0 ± 2.0	54.0 ± 7.2	49.3 ± 4.2	40.67 ± 4.16	50.00 ± 6.00
One vote	42.7 ± 7.6	53.3 ± 8.1	45.3 ± 3.1	51.33 ± 7.57	46.67 ± 1.15
Two votes	56.0 ± 5.3	50.7 ± 10.3	40.0 ± 7.2	48.67 ± 9.87	44.67 ± 4.62
Four votes	55.3 ± 5.0	52.7 ± 4.2	48.7 ± 2.3	46.67 ± 5.77	48.00 ± 7.21
Shuffled time	30.0 ± 4.0	36.0 ± 3.5	43.3 ± 4.6	44.67 ± 2.31	42.00 ± 5.29
Shortest trajectory	60.0 ± 8.7	56.0 ± 7.2	42.0 ± 7.2	46.67 ± 8.33	52.67 ± 8.08
Net progress	34.0 ± 7.2	42.7 ± 1.2	39.3 ± 4.2	43.33 ± 14.05	46.00 ± 3.46
Time MSE	45.3 ± 6.1	36.0 ± 2.0	51.3 ± 6.1	44.67 ± 1.15	48.00 ± 7.21
Local efficiency (16)	0.0 ± 0.0	32.7 ± 4.6	42.0 ± 2.0	44.00 ± 6.00	43.33 ± 1.15

Table 11: Reference-source sensitivity at nominal MH budget 30 (success %). Each source has three reference-panel repeats, distinct from student seeds; each cell summarizes three students.

Reference panel	Square	Transport
panel_high_r0	67.3 ± 11.7	47.3 ± 11.4
panel_mixed_r0	75.3 ± 3.1	57.3 ± 5.0
panel_low_r0	68.7 ± 11.5	52.7 ± 5.8
panel_high_r1	75.3 ± 1.2	62.7 ± 7.0
panel_mixed_r1	65.3 ± 5.0	47.3 ± 9.0
panel_low_r1	64.7 ± 3.1	47.3 ± 6.4
panel_high_r2	68.7 ± 4.2	55.3 ± 3.1
panel_mixed_r2	68.0 ± 2.0	58.0 ± 7.2
panel_low_r2	67.3 ± 5.8	47.3 ± 8.3

A.7 Fresh-scene evaluation

We evaluated existing final EMA checkpoints at nominal budget 30 on the same 50 new scenes, using independent random-number streams for each scene and disabling TF32. Both the scenes and execution settings differ from the original evaluation. Paired 95% intervals use 5,000 crossed resamples of three training seeds and 50 scenes, with no multiple-comparison correction. For full TPEC minus one vote, the intervals are $[-10.0, 24.0]$ on Square and $[-12.0, 16.67]$ on Transport. For the corresponding short-trajectory comparisons, they are $[-26.67, 8.0]$ and $[-16.0, 14.67]$.

Table 12: Fresh-scene evaluation at MH $b = 30$. Success rate (%, mean $\pm$ sample standard deviation computed from the three reported seed rates). These are new evaluations of existing checkpoints.

Method	Square	Transport
TPEC	71.33 ± 9.45	51.33 ± 3.06
One vote	64.67 ± 3.06	49.33 ± 3.06
Shortest trajectory	80.67 ± 4.62	52.00 ± 7.21
Net progress	65.33 ± 5.77	45.33 ± 5.03
Local efficiency (16)	21.33 ± 5.03	34.00 ± 6.93
Shuffled time	72.67 ± 8.08	37.33 ± 10.26
SCIZOR-DINOv2	57.33 ± 3.06	19.33 ± 9.24
WARP-RM-DINOv2	46.00 ± 11.14	44.00 ± 8.00

Across six additional high-mixture pools, consensus wins twice, ties once, and loses three times against one vote. Every paired interval includes zero. Each 144-trajectory pool preserves the original smallest absolute window budget, rather than a 10% retention fraction of the new pool.

A.8 Preparation measurements

Elapsed wall-clock time includes queueing. Summed process time adds concurrent processes. It does not represent GPU-hours. Neither measure includes student training or final evaluation. Dataset auditing, cache construction, and environment setup were not fully timed separately. SCIZOR and WARP pipeline totals include shared rendering and DINO features, so those costs should be counted once when summing experiments. Preparation is reused across five budgets and three seeds.

Table 13: PH preparation minutes, with elapsed and summed process time shown separately. These measures exclude student training and evaluation and are not GPU-hours.

Method	Square	Transport	Can	Tool Hang
Elapsed preparation time (min)				
TPEC	0.83	1.76	1.57	2.19
CUPID	163.55	776.78	142.79	452.74
SS	148.15	385.59	106.54	327.30
DemInf	3.23	3.89	3.97	3.82
SCIZOR	14.52	32.56	14.25	23.28
WARP-RM	68.22	92.32	72.93	78.22
Sum of preparation-process durations (min)				
TPEC	5.03	5.43	5.60	5.92
CUPID	233.59	1024.89	167.10	670.34
SS	218.65	639.80	145.80	544.82
DemInf	12.04	12.95	12.74	13.12
SCIZOR	49.30	171.90	51.27	109.76
WARP-RM	103.04	230.21	109.93	163.26

On the same physical two-GPU setup, the nine-student Can batch times (preparation/total, in minutes) are 1.04/104.74 for TPEC (r2), 99.57/202.05 for CUPID (r3), and 0.12/103.15 for Random (r3). These are batch timings, not per-student latency or an equal-compute comparison.

B RoboTwin and Multitask Protocols

B.1 Visual manipulation

The clean experiments use approximately 500 candidate demonstrations, 600 training epochs, and batch size 128. Budgets refer to candidate windows. Source-table percentages denote retained data; the figures use the complementary filter ratio (100% minus the retained percentage). Clean evaluates 50 episodes per seed (24/25/26); the strong condition evaluates 100 per seed (0/1/2). TPEC uses the visual–joint configuration. Full obtains $74.7 \pm 5.8\%$ on Hammer and $7.3 \pm 5.0\%$ on Pillbottle, but uses more training updates.

The clean-task tables and curves report three-seed means and sample standard deviations across 10–90% retained data (90–10% filter ratios). At a 10% filter ratio, TPEC uses the TPEC-State48 runs.

The three success counts per method are provided in Table 16. DemInf denotes the recorded pipeline at each retained percentage: its 10% entry was originally named DemInf-DINOv2, 30/50/70% were named DemInf, and the 90% entry uses DemInf-image-NMS. CUPID clean references overlap 44/50 Hammer and 49/50 Pillbottle test scenes; SS clean references have no overlap. Both CUPID and SS have 100/100 overlap in the original strong evaluation. We report these overlap statistics so that the development-scene comparisons can be interpreted with the corresponding reference protocol.

Table 14: Hammer clean success (%). Cells report three-seed means and sample s.d.; columns denote retained percentages. Development-scene results; CUPID reference overlap is disclosed in the protocol.

Method	10%	30%	50%	70%	90%
TPEC	20.00 ± 15.87	69.3 ± 4.6	69.3 ± 2.3	75.3 ± 3.1	70.00 ± 7.21
Random	32.67 ± 11.55	63.3 ± 17.0	67.3 ± 8.3	72.0 ± 2.0	76.67 ± 3.06
Random-NMS	29.33 ± 5.03	62.0 ± 2.0	69.3 ± 2.3	72.0 ± 11.1	74.00 ± 3.46
CUPID	43.33 ± 8.08	71.3 ± 4.6	70.7 ± 6.4	76.0 ± 4.0	73.33 ± 6.11
SS	22.00 ± 4.00	61.3 ± 7.0	71.3 ± 8.1	71.3 ± 2.3	78.00 ± 2.00
DemInf	14.00 ± 6.00	57.3 ± 10.3	68.0 ± 8.0	72.0 ± 3.5	68.00 ± 10.39
SCIZOR-DINOv2	35.33 ± 14.05	69.3 ± 6.1	75.3 ± 3.1	67.3 ± 1.2	72.67 ± 3.06
WARP-RM-DINOv2	21.33 ± 3.06	42.0 ± 6.9	54.0 ± 2.0	61.3 ± 7.6	68.00 ± 3.46

Table 15: Pillbottle clean success (%). Cells report three-seed means and sample s.d.; columns denote retained percentages. Development-scene results; CUPID reference overlap is disclosed in the protocol.

Method	10%	30%	50%	70%	90%
TPEC	2.00 ± 2.00	24.0 ± 0.0	10.0 ± 5.3	12.0 ± 6.0	12.67 ± 6.43
Random	0.67 ± 1.15	6.0 ± 6.0	4.0 ± 3.5	5.3 ± 4.2	6.00 ± 4.00
Random-NMS	1.33 ± 1.15	8.0 ± 2.0	5.3 ± 3.1	7.3 ± 4.6	8.00 ± 7.21
CUPID	0.67 ± 1.15	0.7 ± 1.2	3.3 ± 3.1	4.0 ± 4.0	12.67 ± 10.26
SS	4.67 ± 1.15	4.0 ± 3.5	6.7 ± 2.3	7.3 ± 6.1	6.67 ± 5.03
DemInf	1.33 ± 1.15	16.7 ± 1.2	3.3 ± 2.3	7.3 ± 4.6	10.67 ± 9.87
SCIZOR-DINOv2	0.00 ± 0.00	1.3 ± 1.2	7.3 ± 6.1	12.7 ± 11.5	8.00 ± 8.72
WARP-RM-DINOv2	0.00 ± 0.00	2.0 ± 2.0	6.0 ± 0.0	8.0 ± 10.6	9.33 ± 9.45

Table 16: Clean RoboTwin results at a 10% filter ratio (90% retained data). Each entry lists successes in three runs of 50 episodes. TPEC denotes TPEC-State48; CUPID, DemInf, and SS denote CUPID-NMS, DemInf-image-NMS, and Success-similarity-NMS, respectively.

Method	Hammer	Pillbottle
TPEC	31 / 36 / 38	10 / 4 / 5
Random	38 / 40 / 37	5 / 1 / 3
Random-NMS	38 / 38 / 35	8 / 3 / 1
CUPID	34 / 36 / 40	12 / 2 / 5
SS	38 / 40 / 39	6 / 3 / 1
DemInf	28 / 37 / 37	11 / 3 / 2
SCIZOR-DINOv2	35 / 36 / 38	9 / 1 / 2
WARP-RM-DINOv2	33 / 36 / 33	10 / 1 / 3

Table 17: Hammer strong at 30% retention (success %, three seeds). CUPID and SS use references overlapping all 100 development scenes.

Method	Success
TPEC-Visual96	43.0 ± 7.2
Random	33.0 ± 2.6
CUPID	41.0 ± 9.0
SS	55.7 ± 16.0
DemInf	39.0 ± 2.0
SCIZOR-DINOv2	35.7 ± 1.5
WARP-RM-DINOv2	23.0 ± 9.2

Table 18: Clean Hammer on fresh scenes (success %). CUPID and SS are evaluated on the fresh-scene protocol; the table is reported separately from the development-scene comparison.

Method	30%	50%	70%
CUPID	73.3 ± 5.0	80.7 ± 3.1	84.7 ± 2.3
SS	65.3 ± 7.6	77.3 ± 8.1	81.3 ± 4.2

On fresh strong scenes, SS scores $54.3 \pm 9.0\%$ and CUPID $38.7 \pm 4.0\%$. This fresh-scene comparison concerns CUPID and SS; it is separate from the development-scene comparison in the main text.

B.2 Multitask fine-tuning

The π_0 experiment fine-tunes jointly on 50 tasks using 2,500 demonstrations. Table 19 reports five methods across 30/50/70% budgets, ATHENA at 50%, and the Full control. Clean denotes ordinary scenes and DR denotes randomized scenes; DR is not assumed equivalent to the single-task strong configuration. SCIZOR and WARP retain their reported names, with their implementation variants specified in the corresponding task tables. We report aggregate success as point estimates, without between-seed uncertainty intervals or significance tests. Retention percentages are the experiment's budget labels and are not converted into demonstration counts. The Full-data comparison measures data efficiency and is not interpreted as an equal-training-compute comparison.

Table 19: π_0 fine-tuning on 50 tasks using 2,500 demonstrations. Reported success (%) is Clean / Domain Randomized. Full: 43.12 / 15.10.

Method	30%	50%	70%
TPEC	45.88 / **14.50**	**48.56** / 17.15	**48.54 / 18.17**
Random	45.54 / 14.10	44.10 / 16.22	47.18 / 16.10
Random-NMS	**46.10** / 13.82	45.52 / 15.28	44.04 / 17.20
SCIZOR	42.98 / 14.36	42.42 / 13.04	44.06 / 15.32
WARP	40.32 / 14.16	44.08 / 16.20	46.92 / 17.68
ATHENA*		43.36 / **17.30**	

C TPEC implementation details and proofs

C.1 Reference sampling and predictor checks

The reference set $\mathcal{R}$ denotes the union of all fitting subsets, so its size M is the total reference budget. In the low-dimensional Square/Transport configuration, $M = 48$ and $H = 8$, giving six fitting trajectories per head. Each reference trajectory registers 128 state slots. For a trajectory with at least 128 states, we retain the endpoints and sample 126 distinct interior indices without replacement. For a shorter trajectory, all states are retained and repeated slots fill the remaining positions. Slots are retained during fitting and calibration but deduplicated for trajectory-efficiency queries.

The low-dimensional head has two 128-unit GELU hidden layers followed by a scalar sigmoid output. It is fitted using AdamW for 10,000 steps, with batch size 256, learning rate 3×10^{-4}, and weight decay 10^{-6}. Observations are standardized using candidate-pool means and standard deviations, with a 10^{-5} floor on each standard deviation. These are the low-dimensional configuration values; visual inputs and reference budgets follow their respective experiment configurations.

For head h, calibration uses $\mathcal{R} \setminus \mathcal{R}_h$. MSE is averaged over registered slots within a trajectory and then equally across calibration trajectories. The constant-0.5 comparator is evaluated using exactly the same slots and averaging. The low-dimensional protocol also trains a paired control with the same architecture, initialization, fitting states, and optimization, but with timestamp labels shuffled once within each fitting trajectory. The normal head's mean calibration efficiency must exceed both 0.5 and that of its paired control. All eight heads must pass before policy training in this configuration. Shuffled controls do not vote in the consensus. This protocol check is distinct from the algebraic fallback $w_i = 0.5$ at $m_i = 0$.

Non-reference trajectories are queried at at most 128 distinct states in the low-dimensional configuration, retaining endpoints and sampling interior positions without replacement. Reference trajecto-

ries reuse their registered positions after deduplication. All heads use the same query positions for a given trajectory. The implementation computes differences and ratios in float64 from cached float32 predictions. It uses the exact zero-variation convention in Eq. (5), without adding a denominator epsilon.

The low-budget configuration fixes temporal suppression radius $r = 1$ and requires exactly K accepted chunks without relaxation. Some high-budget configurations allow relaxation to $r = 0$. Budget nesting applies to a common fixed-radius acceptance sequence. It does not follow across configurations with different radii.

Preparation cost. With U optimization steps and batch size B_{fit} per head, fitting processes HUB_{fit} head-training examples. Candidate scoring requires $H\sum_i L_i$ compact-head evaluations, and sorting N_c candidate chunks costs $O(N_c \log N_c)$. Feature extraction and temporal suppression add their own costs. These counts describe scorer operations and do not include downstream policy training.

C.2 Visual progress predictor

Visual features. The standardized 1920-dimensional visual representation comprises five 384-dimensional tokens: one global CLS token and four regional tokens corresponding to a 2×2 spatial grid.

Token construction. Within each predictor, all five visual tokens share a learned $384 \to 96$ linear projection. A separate $14 \to 96$ linear layer maps the joint and gripper state to one state token. Adding a learned 96-dimensional progress readout token gives seven tokens in total. We add a learned identity embedding to each token, distinguishing the readout, global image, four spatial regions, and robot state. These embeddings identify the source of information within the current observation and contain no trajectory-time encoding.

Transformer and output. The tokens pass through two independently initialized pre-norm Transformer layers, each with model width 96, four attention heads, a feed-forward hidden width of 192, GELU activation, and dropout 0.1. Pre-norm applies layer normalization before each attention or feed-forward sublayer. The final readout-token representation passes through layer normalization, a $96 \to 1$ linear layer, and a sigmoid to produce the scalar temporal coordinate. The visual backbone stays frozen; the projections, readout token, identity embeddings, Transformer, and output layers are trained as part of the progress predictor.

C.3 Proofs for efficiency and aggregation

Proposition 1 (Calibration invariance and path stability) *For $a > 0$ and $b \in \mathbb{R}$, $q(ap + b) = q(p)$. For a comparison path s with $V_s > 0$ at the same query points, let $p = as + b + e$ and $E = \sum_j |e_{j+1} - e_j|$. Then*

$$|q(p) - q(s)| \le \min\left\{1, \frac{E}{aV_s}\right\}. \tag{10}$$

Directional decomposition. Writing $\Delta p_j = (\Delta p_j)_+ - (-\Delta p_j)_+$ gives $D = P - B$ and $V = P + B$. Hence $q = (1 + D/V)/2 = P/(P + B)$ for $V > 0$. At fixed $D > 0$, $P = D + B$ and

$$\frac{\partial}{\partial B}\frac{D+B}{D+2B} = -\frac{D}{(D+2B)^2} < 0.$$

This establishes the backtracking interpretation at fixed positive net advancement. Inserting a repeated prediction adds a zero increment and changes neither D nor V.

Proof of Proposition 1. Under $p' = ap + b$, $a > 0$, both D and V are multiplied by a. The score is unchanged, including the constant-path case.

For the perturbation statement, let $\widetilde{s} = as + b$ and $W = V_{\widetilde{s}} = aV_s > 0$. Telescoping and the reverse triangle inequality give

$$|D_p - D_{\widetilde{s}}| \le E, \qquad |V_p - W| \le E.$$

If $V_p > 0$, rewrite the difference using W as the common normalizing reference:

$$|q(p) - q(\widetilde{s})| = \frac{1}{2W}\left|D_p - D_{\widetilde{s}} + \frac{D_p}{V_p}(W - V_p)\right| \leq \frac{|D_p - D_{\widetilde{s}}| + |W - V_p|}{2W} \leq \frac{E}{W},$$

where $|D_p| \leq V_p$. If $V_p = 0$, the reverse triangle inequality implies $E \geq W$, and $|q(p) - q(\widetilde{s})| \leq 1/2 \leq E/W$. Finally, both scores lie in $[0, 1]$, and affine invariance gives $q(\widetilde{s}) = q(s)$, establishing the stated minimum with one.

For two trajectories with comparison paths s_A, s_B and score errors bounded by ϵ_A, ϵ_B, respectively,

$$q(p_A) - q(p_B) \geq q(s_A) - q(s_B) - \epsilon_A - \epsilon_B.$$

Thus a comparison-score gap exceeding $\epsilon_A + \epsilon_B$ preserves the ordering. If all eligible heads share the same comparison path for a trajectory, the error of their mean is bounded by the mean of their individual error bounds. This extension keeps the eligible sets and vote counts fixed.

Sampling resolution. Let p_C be an endpoint-preserving subsequence of a finer query path p_F. Each coarse increment is a sum of consecutive fine increments, so the triangle inequality yields $V_C \leq V_F$, while $D_C = D_F = D$. When both variations are positive,

$$q(p_C) - q(p_F) = \frac{D}{2}\left(\frac{1}{V_C} - \frac{1}{V_F}\right).$$

For $D > 0$, coarsening can increase measured efficiency by hiding reversals. The score properties in the main text apply to the specified query paths. A fixed query count does not by itself provide equal temporal resolution across different trajectory lengths.

Temporal coordinates and task outcomes. The population timestamp predictor estimates $\mathbb{E}[Y \mid X]$, as established in Section 3.2. Identifying this function with a physical task-progress coordinate requires assumptions on observability and timing. Perfect timestamp predictions would produce $q = 1$ for every trajectory, so timestamp accuracy alone cannot establish a useful ordering. Likewise, a small monotone predicted change can attain $q = 1$ without completing a task. These examples explain why the ranking result states its mean-margin condition explicitly and why downstream performance is evaluated separately.

Cross-entropy consensus. Scores are aggregated after evaluating each path. For example, $(0, 1, \frac{1}{2})$ and $(\frac{1}{2}, 0, 1)$ both have efficiency $2/3$, but their mean path $(\frac{1}{4}, \frac{1}{2}, \frac{3}{4})$ has efficiency 1. Averaging paths can therefore erase evidence of reversals before scoring.

For $q \in [0, 1]$ and $u \in (0, 1)$, define $\mathrm{CE}(q, u) = -q \log u - (1-q)\log(1-u)$. With $s = \sum_{h \in \mathcal{H}_i} q_{h,i}$ and $m = m_i$, consider

$$F(u) = \sum_{h \in \mathcal{H}_i} \mathrm{CE}(q_{h,i}, u) + 2\,\mathrm{CE}(1/2, u) = -(s+1)\log u - (m - s + 1)\log(1-u).$$

Both coefficients are positive, so F is strictly convex. Its unique stationary point is

$$F'(u) = \frac{(m+2)u - (s+1)}{u(1-u)} = 0, \qquad u = \frac{s+1}{m+2} = w_i.$$

The neutral terms are regularization terms, not observed task outcomes. At fixed eligibility, replacing ℓ scores in $[0, 1]$ changes their sum by at most ℓ, giving $|w_i' - w_i| \leq \ell/(m_i + 2)$. Changes to calibration outcomes or eligibility require separate analysis.

Proposition 2 (Conditional ranking reliability) *For trajectories A, B sharing a fixed set of $m \geq 1$ eligible heads, let $Z_h = q_h(A) - q_h(B)$. Conditional on this eligibility pattern, assume that reference construction and fitting induce $\mathbb{E}[m^{-1}\sum_h Z_h] = \gamma > 0$, $\mathrm{Var}(Z_h) \leq \sigma^2$, and $\mathrm{Cov}(Z_h, Z_k) \leq \rho\sigma^2$ for $h \neq k$, where $0 \leq \rho \leq 1$. Then*

$$\Pr(w_A \leq w_B) \leq \frac{v_m}{v_m + \gamma^2}, \qquad v_m = \sigma^2\left(\rho + \frac{1-\rho}{m}\right). \tag{11}$$

Proof of Proposition 2. All moments in this proof are conditional on the fixed eligibility pattern in the proposition. Let $\bar{Z} = m^{-1} \sum_h Z_h$. Then

$$\begin{aligned}\mathrm{Var}(\bar{Z}) &= \frac{1}{m^2}\left(\sum_h \mathrm{Var}(Z_h) + 2\sum_{h<k} \mathrm{Cov}(Z_h, Z_k)\right) \\ &\le \frac{m\sigma^2 + m(m-1)\rho\sigma^2}{m^2} = v_m.\end{aligned}$$

The common vote count implies $w_A - w_B = m\bar{Z}/(m+2)$. Cantelli's inequality, with $\mathbb{E}[\bar{Z}] = \gamma > 0$, gives

$$\Pr(w_A \le w_B) = \Pr(\bar{Z} \le 0) \le \frac{\mathrm{Var}(\bar{Z})}{\mathrm{Var}(\bar{Z}) + \gamma^2} \le \frac{v_m}{v_m + \gamma^2}.$$

No independence assumption is needed. Shared fitting distributions, input normalization, representations, and calibration can produce correlated errors. The proposition is a sufficient-condition analysis and does not establish that the stated margin and covariance bounds hold for every dataset or every reference budget.

At fixed γ, σ^2, ρ, the bound decreases with m when $\rho < 1$. Shrinkage preserves ordering when vote counts agree. With unequal vote counts, $w_i = 0.5 + [m_i/(m_i+2)](\overline{q}_i - 0.5)$ applies different shrinkage factors, and the proposition cannot be used without accounting for the different eligible sets.

Conditional enrichment. Draw a chunk uniformly from the candidate pool. Let G denote a higher-quality source class used only for analysis, and let A denote acceptance by the complete ranking-and-suppression rule. Write $\Pr(G) = \eta$, $\Pr(A \mid G) = \alpha$, and $\Pr(A \mid G^c) = \beta$, with $0 < \eta < 1$. For a nonempty selected set, the actual selection rule induces $\Pr(A) = \eta\alpha + (1-\eta)\beta$. Bayes' rule gives $\Pr(G \mid A) = \eta\alpha/[\eta\alpha + (1-\eta)\beta]$. Subtracting η gives

$$\Pr(G \mid A) - \eta = \frac{\eta(1-\eta)(\alpha - \beta)}{\eta\alpha + (1-\eta)\beta}. \tag{12}$$

Thus selection increases the share of this class when $\alpha > \beta$ at the chosen budget. These analytical labels are not inputs to TPEC. This statement includes temporal suppression and tie-breaking in the event A; it does not replace the implemented selector by an ideal score threshold. The condition $\alpha > \beta$ must hold at the actual budget. It is not implied by affine invariance alone, and enrichment by itself is not a policy-return bound: downstream utility also depends on state coverage and policy fitting.

C.4 Progress-guided distribution concentration

Analytical progress model. For each trajectory contributing to the analysis, assume a piecewise-linear latent scalar progress path $s_i : [0,1] \to \mathbb{R}$, with endpoints $a_i = s_i(0) < b_i = s_i(1)$. Positive net advancement is an additional structural assumption beyond the uncurated-reference setting. Endpoints may differ across trajectories, and positive advancement need not mean task completion. Let P_i and B_i be forward and backward variation, so $D_i = b_i - a_i > 0$, $V_i = P_i + B_i = D_i + 2B_i$, and $q_i^\star = P_i/V_i$. A sampled path is interpreted through piecewise-linear interpolation; comparison with a finer physical path requires a separate query-resolution error.

For almost every level $u \in \mathbb{R}$, let $n_{i,+}(u)$ and $n_{i,-}(u)$ count upward and downward crossings. Upward crossings exceed downward crossings by one between the endpoints and are equal elsewhere:

$$n_{i,+}(u) - n_{i,-}(u) = \mathbf{1}_{(a_i,b_i)}(u), \qquad \int_{\mathbb{R}} n_{i,+}(u)\,du = P_i, \qquad \int_{\mathbb{R}} n_{i,-}(u)\,du = B_i. \tag{13}$$

Define the probability measure over progress and direction $d \in \{+,-\}$ by

$$\mu_i(du, dd) = \frac{n_{i,+}(u)\delta_+(dd) + n_{i,-}(u)\delta_-(dd)}{V_i}\,du.$$

Its net-forward component is $\mu_{i,0}(du, dd) = D_i^{-1}\mathbf{1}_{(a_i,b_i)}(u)\,du\,\delta_+(dd)$. When $B_i > 0$, define

$$\mu_{i,\mathrm{rt}}(du, dd) = \frac{n_{i,-}(u)}{2B_i}\,du\,[\delta_+(dd) + \delta_-(dd)].$$

Equation (13) gives the exact identity

$$\mu_i = (2q_i^\star - 1)\mu_{i,0} + 2(1 - q_i^\star)\mu_{i,\mathrm{rt}}. \tag{14}$$

If $B_i = 0$, the second coefficient is zero and its probability measure can be chosen arbitrarily. The backward share alone is $B_i/V_i = 1-q_i^\star$; the round-trip share $2(1-q_i^\star)$ also includes an equal amount of forward variation. These are masses in progress-variation sampling, not fractions of elapsed time or failed demonstrations.

From progress crossings to chunks. Fix a probability kernel $\mathcal{K}_i(dc \mid u, d)$ supported on actual chunks of trajectory i, associating progress crossings with their observation–action training examples. It can average over different visits to a level and include chunk context. Use this same kernel for all components of Eq. (14) and define

$$R_i = \mu_i\mathcal{K}_i, \qquad G_i = \mu_{i,0}\mathcal{K}_i, \qquad U_i = \mu_{i,\mathrm{rt}}\mathcal{K}_i.$$

Linearity of integration gives the trajectory-level decomposition

$$R_i = (2q_i^\star - 1)G_i + 2(1 - q_i^\star)U_i. \tag{15}$$

The kernels serve only to define an analytical reference distribution; TPEC does not fit them. Different trajectories can have different G_i, reflecting their progress intervals, initial conditions, action modes, or contact configurations. A short monotone partial attempt can have $q_i^\star = 1$ while omitting most target behaviors. The net-forward component is a measure decomposition and need not be a physically executable trajectory after deleting excursions. The stated mixture is applied to selected trajectories with $D_i > 0$. Retained trajectories with zero or negative net progress require separate accounting rather than extending these coefficients to a negative net-forward mass.

For the actual selected set, use $\rho_i = n_i(S)/K$ and $\beta_i = 2(1 - q_i^\star)$. Then $\beta_S = \sum_i \rho_i\beta_i$. For $0 < \beta_S < 1$, define

$$G_S = \frac{\sum_i \rho_i(1 - \beta_i)G_i}{1 - \beta_S}, \qquad U_S = \frac{\sum_i \rho_i\beta_i U_i}{\beta_S}.$$

Summing $R_S = \sum_i \rho_i R_i$ proves Eq. (8); a component with zero mass can again be chosen arbitrarily. Actual chunk counts are required: an unweighted average across selected trajectories generally gives a different mixture. R_S remains a progress-weighted reference distribution, while $\widehat{P}_S$ is uniform over the selected chunk indices. Temporal suppression, selection within trajectories, window overlap, and timing can separate these distributions.

Training-gradient discrepancy. Let $\ell_\theta(c)$ denote the imitation loss after averaging its internal training randomness, and write $g_\theta(c) = \nabla_\theta\ell_\theta(c)$ and $g_P(\theta) = \mathbb{E}_{c\sim P}g_\theta(c)$. On a parameter region Θ, assume these gradients are integrable and $\|g_\theta(c)\| \le C_g$ for all distributions under consideration. Interchanging differentiation and expectation, when needed, is also assumed valid. Define

$$\eta_S = \sup_{\theta\in\Theta} \|g_{\widehat{P}_S}(\theta) - g_{R_S}(\theta)\|, \tag{16}$$

$$\delta_S = \sup_{\theta\in\Theta} \|g_{G_S}(\theta) - g_{P_\star}(\theta)\|. \tag{17}$$

Here $P_\star$ represents behaviors needed for the target task, including required recovery modes. By Eq. (8),

$$g_{R_S} - g_{P_\star} = (1 - \beta_S)(g_{G_S} - g_{P_\star}) + \beta_S(g_{U_S} - g_{P_\star}).$$

Adding $g_{\widehat{P}_S} - g_{R_S}$ and applying the triangle inequality and the bound $\|g_{U_S} - g_{P_\star}\| \le 2C_g$ proves Eq. (9). This is a conditional bound on expected training signals. Bounded gradients are an assumption on Θ, not a global property of arbitrary policy networks. The bound does not establish that η_S or δ_S is small, nor does a smaller upper bound establish a smaller realized discrepancy or higher closed-loop return.

Temporal NMS and coverage. All chunks from a trajectory share its consensus score, so score ranking alone does not distinguish neighboring windows from temporally separated ones. Temporal NMS imposes spacing between selected starts. When nearby chunks carry similar supervision, this can allocate a fixed budget across a broader range of trajectory segments and mitigate coverage loss from local redundancy. This complements efficiency-based allocation across trajectories. In Eq. (9),

the within-trajectory sampling effect relates most directly to η_S, which compares actual selected chunks with the progress-weighted reference distribution. Suppression can also change the selected trajectory weights ρ_i, thereby affecting G_S and δ_S. The effect on these discrepancies depends on how temporal separation reflects task-relevant differences: distant chunks can remain behaviorally redundant, while a brief event can carry distinct supervision. Thus NMS encourages temporal coverage, with its impact on target-behavior coverage depending on the data and suppression radius.

Connection to observed scores. For $m_i > 0$, define the unshrunk eligible-head mean $\overline{q}_i = m_i^{-1} \sum_{h \in \mathcal{H}_i} q_{h,i}$. If $|\overline{q}_i - q_i^\star| \le \epsilon_i$, then

$$\beta_S \le \min \left\{ 1, 2 \left(1 - \sum_i \rho_i \overline{q}_i + \sum_i \rho_i \epsilon_i \right) \right\}. \tag{18}$$

The errors include both predictor shape error and differences between queried and analytical paths. For a trajectory with no eligible head, the neutral analytical proxy $\overline{q}_i = 0.5$ and $\epsilon_i = 0.5$ give a valid but uninformative extension. The consensus w_i is not substituted directly for $q_i^\star$: its pseudo-count shrinkage depends on m_i. The selection weights ρ_i already incorporate the actual w_i ranking, random tie-breaking, and suppression. No global optimality of greedy suppression is needed for these identities. Relative concentration compared with another selection requires a larger actual chunk-weighted ideal efficiency, or a sufficiently large observed margin after accounting for score errors.

Pauses, coverage, and interpretation. Inserting repeated values into a fixed progress path leaves $q_i^\star$ unchanged, although it can make pauses dominate time-based sampling. Consequently, high efficiency alone cannot bound η_S. As a simple illustration, for equal-duration steps with progress increments in $\{-h, 0, h\}$, $h > 0$, let s be the fraction of zero-progress steps. The fractions of steps assigned to net-forward, round-trip, and pause components are $(1-s)(2q_i^\star - 1)$, $2(1-s)(1-q_i^\star)$, and s, respectively. Unequal movement speeds and chunk windows require further accounting. Recovery actions can also be useful training examples; omitting target-relevant behaviors increases the coverage discrepancy rather than constituting an automatic improvement.

The concentration interpretation therefore concerns a possible reduction of task-relevant training-signal mismatch through selection. It neither requires synthesizing new chunks nor changes the downstream imitation objective. Its applicability is tested by measuring score fidelity, the composition and coverage of the actual selected data, and policy outcomes; efficiency scores alone do not verify all three.

D Disclosure of AI Assistance

In accordance with ICLR's policy on AI assistance, we disclose the use of generative AI tools for manuscript writing and polishing, literature retrieval and discovery, and research ideation and execution. These tools assisted with improving grammar, wording, and clarity; identifying potentially relevant publications and related work; and exploring and refining research ideas and approaches to research tasks. The authors reviewed and revised the AI-assisted text, checked references against the original sources, and critically evaluated AI-generated suggestions before incorporating them into the work. All AI-assisted contributions were subject to human review and verification. The authors made the final decisions regarding the research and its presentation and take full responsibility for the accuracy, originality, and integrity of the manuscript, including its claims, citations, results, and associated research artifacts.