WHEN TO TRUST YOUR TEACHER: UNCERTAINTY-AWARE ON-POLICY DISTILLATION FOR EMBODIED POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation mitigates compounding errors by aligning supervision with the student-induced state distribution, but its effectiveness is limited by a mismatch between this distribution and the teacher's reliable support. We study how to effectively leverage teacher policies for on-policy learning of flow-based robotic policies. We formulate OPD as policy optimization in a teacher-induced reward MDP and derive an uncertainty-penalized objective that, under an admissible uncertainty estimate, lower-bounds the return in the optimal-policy-induced MDP. Our analysis shows that optimizing this pessimistic objective yields a policy that is near-optimal among policies with bounded cumulative reward uncertainty, revealing a trade-off between the scope and tightness of the guarantee. We propose Uncertainty-Aware On-Policy Distillation (UA-OPD), a framework that accounts for potentially unreliable teacher supervision through uncertainty penalization. We instantiate UA-OPD for flow-based robotic policies with a self-consistency uncertainty estimator, combining soft penalization with uncertainty-based trajectory truncation. Experiments on LIBERO and RoboTwin 2.0 show that UA-OPD achieves a 1.5$\times$ improvement in sample efficiency and higher average peak success rates than OPD, while surpassing the teacher on most evaluated suites and tasks. Its benefits extend to out-of-distribution generalization and one-step policy distillation. These findings suggest that embodied OPD benefits from adapting the use of teacher supervision to its reliability.

1 INTRODUCTION

Recent advances in foundation models have renewed interest in distillation methods that move beyond static teacher-labeled datasets (Hinton et al., 2015; Gu et al., 2024). Across large foundation models, on-policy distillation (OPD) has been explored for multi-teacher capability amalgamation, large-to-small knowledge distillation, and sampling acceleration via progressive distillation, where the student learns from teacher feedback on its own generated outputs instead of only imitating a fixed offline corpus (Agarwal et al., 2024; Sang et al., 2026; Zhao et al., 2026). By adapting supervision to the student-induced distribution, OPD exposes the student to its own mistakes during training, narrows the gap between training-time and test-time behavior, and mitigates error accumulation in sequential generation. These properties make OPD especially appealing for sequential decision-making problems, where the student's future inputs are shaped by its previous outputs (Ross et al., 2011; Czarnecki et al., 2019).

This motivation is even stronger in embodied intelligence, where each action of a robotic policy changes the future observation distribution. However, this also introduces a critical failure mode where supervision may be unreliable when derived from states that are out-of-distribution for the teacher. Blindly distilling such out-of-support supervision can make learning inefficient and degrade the quality of adaptation, especially in long-horizon manipulation tasks where early deviations can persist and amplify. This raises the central question this work aims to answer: **how should a student balance the performance gains of learning from teacher feedback against the risk of trusting possibly unreliable teacher predictions**?

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.

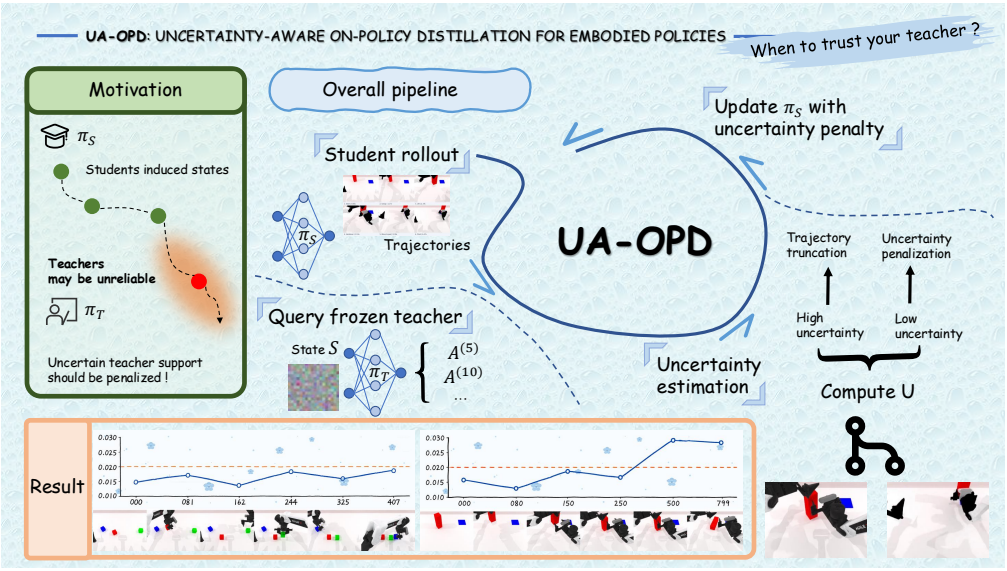


Figure 1: Overview of UA-OPD.

To approach this question, since on-policy distillation algorithms cannot improve the teacher model itself, we expect that such algorithms require careful use of teacher supervision in regions outside the teacher's reliable support. Quantifying the risk induced by sub-optimal teacher and appropriately trading off this risk with return is a crucial ingredient for building a strong OPD algorithm. To this end, we propose Uncertainty-Aware On-Policy Distillation (UA-OPD), which incorporates a reward penalty based on an estimate of the reward error. We first formulate OPD as a teacher-induced reward Markov Decision Process (MDP), where the teacher defines a surrogate reward over student-visited states and actions. Building on this, we represent potential errors in the teacher-induced reward through an uncertainty cost and use it to construct a penalized distillation objective.

We show that, under an admissible uncertainty cost, the penalized objective provides a conservative lower bound on the return in the optimal-policy-induced MDP, explicitly accounting for the risk of unreliable teacher supervision. To instantiate UA-OPD for flow-based robotic policies, we estimate teacher uncertainty from the discrepancy between teacher action chunks generated from the same initial noise under different denoising discretizations (Liu et al., 2023b; Yang et al., 2024). This self-consistency-based estimate guides soft uncertainty penalization and trajectory truncation during student training.

Experiments on LIBERO and RoboTwin 2.0 demonstrate that UA-OPD achieves a 1.5× improvement in sample efficiency and higher average peak success rates than vanilla OPD, while surpassing the teacher on most evaluated suites and tasks (Liu et al., 2023a; Chen et al., 2026). Further experiments on RoboTwin 2.0 show improved out-of-distribution generalization and benefits in one-step policy distillation. Component ablations support the complementary roles of trajectory truncation and soft uncertainty penalization.

In summary, our contributions are:

- We propose UA-OPD, an on-policy distillation framework that accounts for teacher reliability on student-induced states. We instantiate it for flow-based policies using self-consistency uncertainty to guide soft penalization and trajectory truncation.
- Under an admissible uncertainty cost, we establish a conservative lower bound on the ideal OPD return and a near-optimality guarantee relative to policies with bounded cumulative uncertainty.
- Experiments on LIBERO and RoboTwin 2.0 demonstrate improved sample efficiency, average peak success rates, and out-of-distribution generalization, with benefits extending to one-step distillation.

2 Related Work

On-Policy Distillation. Knowledge distillation is a prevalent technique for transferring learned capabilities from teacher models to student models (Hinton et al., 2015). In sequential decision-making, standard offline distillation is inherently prone to compounding errors (Ross et al., 2011). To overcome this, On-Policy Distillation (OPD) queries the teacher to supervise the student on the student's self-generated trajectories, providing corrective actions for the specific states the student actually visits (Czarnecki et al., 2019; Gu et al., 2024). Building upon these foundations, self-distillation emerges as a specialized paradigm where the model itself acts as the teacher (Zhang et al., 2019). Recent works have heavily leveraged on-policy self-distillation to compress complex reasoning capabilities and enable continual learning without relying on external or superior oracle models (Sang et al., 2026; Zhao et al., 2026; Shenfeld et al., 2026). However, in embodied intelligence, although initial efforts (Zhong et al., 2026) have introduced on-policy distillation for robotic control, the systematic utilization of these frameworks remains underexplored.

Vision-Language-Action Models. Scaling up large pretrained vision-language models (VLMs) for continuous robot control has rapidly emerged as the dominant paradigm for building generalist embodied agents (Reed et al., 2022; Driess et al., 2023). A foundational and prevalent paradigm formulates control by casting continuous actions as discrete tokens, enabling autoregressive action generation through standard Transformer architectures (Brohan et al., 2023; Zitkovich et al., 2023; O'Neill et al., 2024; Jiang et al., 2023; Kim et al., 2025b). Beyond token-based formulations, recent VLA architectures increasingly adopt expressive continuous action distributions to better capture the high-precision and multi-modal nature of robotic control (Kim et al., 2025a). This shift towards continuous actions is heavily influenced by the success of Diffusion Policy (Chi et al., 2025), which has been scaled up in foundation models (Team et al., 2024). Most recently, flow matching has been introduced as an efficient alternative for continuous action generation, with models such as (Black et al., 2025b;a) demonstrating state-of-the-art performance and scalability in dynamic manipulation tasks.

3 Preliminaries

3.1 On-Policy Distillation

We model the environment as a discounted Markov Decision Process (MDP) $\mathcal{E} = (\mathcal{S}, \mathcal{A}, P, r_{\text{env}}, \rho_0, \gamma)$ with $\gamma \in (0, 1)$, and let Π denote the student policy class. For a policy $\pi \in \Pi$, let d_P^π and ρ_P^π denote its discounted state and state-action occupancy measures, and let $J_\mathcal{E}(\pi) := \bar{\mathbb{E}}_{(s,a)\sim\rho_P^\pi}[r_{\text{env}}(s, a)]$ denote its environment return. Given a supervision policy ψ, on-policy distillation trains π on its induced state distribution by minimizing the reverse-KL objective:

$$\mathcal{L}_{\text{OPD}}^\psi(\pi) = \bar{\mathbb{E}}_{s\sim d_P^\pi}[D_{\text{KL}}(\pi(\cdot \mid s) \| \psi(\cdot \mid s))]\,. \tag{1}$$

Define the supervision-induced reward $r^\psi(s, a) := \log \psi(a \mid s)$ and the corresponding entropy-regularized MDP $\mathcal{M}^\psi = (\mathcal{S}, \mathcal{A}, P, r^\psi, \rho_0, \gamma)$. Its regularized return is:

$$\eta_{\mathcal{M}^\psi}(\pi) := \bar{\mathbb{E}}_{(s,a)\sim\rho_P^\pi}\left[r^\psi(s, a) - \log \pi(a \mid s)\right] = -\mathcal{L}_{\text{OPD}}^\psi(\pi). \tag{2}$$

Therefore, minimizing the OPD loss is equivalent to maximizing the return in $\mathcal{M}^\psi$.

3.2 Transition-Level OPD for Flow Policies

For flow-based policies, directly evaluating the action density $\pi_\theta(a|c)$ is difficult, making the action-level reverse-KL objective in Eq. (1) impractical. Following Flow-GRPO (Liu et al., 2025) and MixGRPO (Li et al., 2026), we adopt a mixed ODE-SDE formulation that replaces action likelihoods with tractable Gaussian transition likelihoods along the denoising trajectory.

Let c denote the policy condition, z the initial noise, and x_i^S the intermediate variable generated by the student trajectory at flow time τ_i, with $\Delta\tau_i = \tau_{i+1} - \tau_i$. We uniformly sample a denoising interval and evaluate both the student and teacher transitions at the same student-generated x_i^S.

When the two transitions share the same standard deviation, the transition-level reverse-KL reduces to

$$\mathcal{L}_{\mathrm{OPD}} = \mathbb{E}_{c,z,i}\left[\frac{\Delta\tau_i}{2\sigma_i^2}\left(1+\frac{\sigma_i^2(1-\tau_i)}{2\tau_i}\right)^2 \left\|v_S(x_i^S,\tau_i,c)-v_T(x_i^S,\tau_i,c)\right\|_2^2\right]. \tag{3}$$

Thus, reverse-KL OPD becomes a weighted velocity-matching objective evaluated on student-induced denoising states, providing the base distillation objective used in the following section.

4 Uncertainty-Aware On-Policy Distillation

This section develops UA-OPD as a pessimistic approach to on-policy distillation under unreliable teacher supervision. We first introduce an uncertainty cost that upper-bounds the error in the teacher-induced OPD reward, and use it to define a conservative distillation objective. We then establish an uncertainty-dependent comparator guarantee, showing that the resulting policy remains competitive with policies whose trajectories incur limited teacher uncertainty. Finally, we instantiate the framework for flow-based robotic policies by combining the mixed ODE-SDE reverse-KL objective with a lightweight discretization self-consistency score, soft uncertainty penalization, and uncertainty-based trajectory truncation.

4.1 Pessimistic OPD

We begin by establishing that OPD preserves environment optimality when the supervision policy is optimal.

Proposition 4.1 (Optimality preservation). Let $\pi^\star \in \arg\max_{\pi\in\Pi} J_{\mathcal{E}}(\pi)$. Then

$$\arg\min_{\pi\in\Pi} \mathcal{L}_{\mathrm{OPD}}^{\pi^\star}(\pi) \subseteq \arg\max_{\pi\in\Pi} J_{\mathcal{E}}(\pi). \tag{4}$$

The proof is provided in Appendix A.1. Proposition 4.1 identifies optimal-policy supervision as the ideal reference for OPD. In practice, however, supervision is provided by a teacher policy π_T, and we must account for the error introduced by replacing $\pi^\star$ with π_T. Let $\mathcal{M} := \mathcal{M}^{\pi^\star}$ and $\widehat{\mathcal{M}} := \mathcal{M}^{\pi_T}$ denote the corresponding regularized MDPs, with rewards $r(s,a) := \log \pi^\star(a|s)$ and $\hat{r}(s,a) := \log \pi_T(a|s)$, respectively. Because the two MDPs share the same dynamics and entropy regularization, their returns differ only through their rewards, and for every $\pi \in \Pi$,

$$\eta_{\widehat{\mathcal{M}}}(\pi) - \eta_{\mathcal{M}}(\pi) = \bar{\mathbb{E}}_{(s,a)\sim\rho_P^\pi}\left[\hat{r}(s,a) - r(s,a)\right]. \tag{5}$$

The right-hand side is the cumulative teacher reward error under the student-induced distribution. To control this error, we introduce a non-negative uncertainty $u : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ satisfying

$$|\hat{r}(s,a) - r(s,a)| \le u(s,a). \tag{6}$$

We then define the pessimistic reward $\widetilde{r}$ and cumulative uncertainty $\epsilon_u(\pi)$ by

$$\widetilde{r}(s,a) := \hat{r}(s,a) - u(s,a), \qquad \epsilon_u(\pi) := \bar{\mathbb{E}}_{(s,a)\sim\rho_P^\pi}\left[u(s,a)\right]. \tag{7}$$

Let $\widetilde{\mathcal{M}}$ denote the regularized MDP induced by $\widetilde{r}$. Since $|\hat{r} - r| \le u$, the pessimistic reward satisfies $r - 2u \le \widetilde{r} \le r$, which implies that, for every $\pi \in \Pi$,

$$\eta_{\mathcal{M}}(\pi) - 2\epsilon_u(\pi) \le \eta_{\widetilde{\mathcal{M}}}(\pi) \le \eta_{\mathcal{M}}(\pi). \tag{8}$$

Thus, $\eta_{\widetilde{\mathcal{M}}}(\pi)$ is a lower bound on the ideal OPD return, with a gap controlled by the cumulative uncertainty. The following theorem characterizes the policy obtained by optimizing this pessimistic objective.

Theorem 4.2 (Pessimistic OPD guarantee). Let $\widetilde{\pi} \in \Pi$ be a δ-optimal policy in $\widetilde{\mathcal{M}}$ satisfying $\eta_{\widetilde{\mathcal{M}}}(\widetilde{\pi}) \geq \sup_{\pi \in \Pi} \eta_{\widetilde{\mathcal{M}}}(\pi) - \delta$ for some $\delta \geq 0$. For any uncertainty budget $\epsilon \geq 0$, define the comparator class $\Pi_\epsilon := \{\pi \in \Pi : \epsilon_u(\pi) \leq \epsilon\}$. If Π_ϵ is nonempty, then

$$\eta_{\mathcal{M}}(\widetilde{\pi}) \geq \sup_{\pi \in \Pi_\epsilon} \eta_{\mathcal{M}}(\pi) - 2\epsilon - \delta. \tag{9}$$

The proof is provided in Appendix A.2. Theorem 4.2 guarantees that $\widetilde{\pi}$ is within $2\epsilon + \delta$ of the best policy in Π_ϵ under the ideal OPD objective. The uncertainty budget therefore controls a scope–tightness trade-off: a smaller ϵ yields a tighter guarantee over a narrower comparator class, whereas a larger ϵ expands the comparator class but loosens the guarantee.

4.2 Uncertainty Estimation for Flow Policies

The admissible uncertainty cost in Eq. (6) depends on the inaccessible optimal policy. To instantiate UA-OPD for flow-based policies, we use discretization self-consistency to estimate teacher uncertainty. We compare teacher action chunks generated from the same initial noise under different denoising budgets, measuring their sensitivity to the numerical discretization. We examine the relationship between this estimate and teacher reliability in Section 5.1.

Let $\mathcal{K} = \{k_1, \ldots, k_N\}$ be a set of $N \geq 2$ denoising budgets. For each $k_n \in \mathcal{K}$, let $A^{(k_n)}(s, z) \in \mathbb{R}^{H \times D}$ denote the action chunk generated by the frozen teacher at state s from the same initial noise $z \sim p_0$ using k_n denoising steps, where H and D denote the action horizon and dimension. All ensemble members share the same teacher parameters and initial noise and differ only in their numerical discretizations. For any $p \geq 1$, we measure the discretization-induced uncertainty of each action element by its normalized p-order dispersion around this consensus:

$$u^{(p)}_{h,d}(s, z) := \frac{\left(\frac{1}{N}\sum_{n=1}^{N} \left|A^{(k_n)}_{h,d}(s, z) - \bar{A}_{h,d}(s, z)\right|^p\right)^{\frac{1}{p}}}{\left(\frac{1}{N}\sum_{n=1}^{N} \left|A^{(k_n)}_{h,d}(s, z)\right|^p\right)^{\frac{1}{p}}}. \tag{10}$$

Here $\bar{A}_{h,d} = \frac{1}{N}\sum_{n=1}^{N} A^{(k_n)}_{h,d}$. The numerator measures disagreement among numerical solutions, while the denominator normalizes this disagreement by the magnitude of the corresponding action coordinate. The resulting score is dimensionless and comparable across action dimensions. We then average the element-wise scores over the action horizon and dimensions to obtain the uncertainty of the complete action chunk:

$$U^{(p)}(s, z) := \frac{1}{HD}\sum_{h=1}^{H}\sum_{d=1}^{D} u^{(p)}_{h,d}(s, z). \tag{11}$$

Using the same initial noise holds the stochastic input fixed across ensemble members, so their disagreement primarily reflects sensitivity to the denoising discretization rather than variation caused by independently sampled action modes. Unlike a conventional model ensemble, this estimator requires neither separately trained policies nor an auxiliary uncertainty model, since all members are generated by evaluating the same frozen teacher under different denoising budgets.

4.3 Uncertainty-Penalized Flow OPD

We now instantiate the pessimistic principle in Section 4.1 for flow-based OPD. Let $\ell_{\mathrm{OPD}}(s, z, i; \theta)$ denote the single-interval weighted velocity-matching loss in Eq. (3), evaluated at a student-induced state s, initial noise $z \sim p_0$, and denoising interval i. This loss corresponds to the reverse KL between the student and teacher transition kernels and admits an entropy-regularized policy optimization interpretation over the denoising process. Incorporating the self-consistency uncertainty $U^{(p)}(s, z)$ gives the pessimistic objective

$$\widetilde{\eta}(\pi_\theta) := \bar{\mathbb{E}}_{s \sim d_P^{\pi_\theta}} \mathbb{E}_{z \sim p_0}\left[-\mathbb{E}_{i \sim \mathrm{Unif}(\mathcal{I})}\left[\ell_{\mathrm{OPD}}(s, z, i; \theta)\right] - \lambda U^{(p)}(s, z)\right], \tag{12}$$

where $\mathcal{I}$ denotes the set of denoising intervals and $\lambda \geq 0$ controls the strength of uncertainty penalization. The distillation term encourages agreement with the teacher, while the uncertainty cost introduces a pessimistic correction for potentially unreliable supervision. Since the expectation is taken under the student-induced state distribution, the objective balances teacher matching against uncertainty accumulated along student trajectories. Setting $\lambda = 0$ recovers the unpenalized flow-based OPD objective.

If the uncertainty cost satisfies the admissibility condition in Eq. (6) for the corresponding transition-level reward error, the pessimistic lower-bound and comparator arguments in Section 4.1 apply to Eq. (12). In practice, self-consistency provides an empirical uncertainty estimate rather than a verified upper bound on the reward error. Its relationship with teacher reliability and reward discrepancy is examined in Section 5.1.

We implement UA-OPD by combining soft uncertainty penalization with hard trajectory truncation. During each student rollout, we estimate $U_t := U^{(1)}(s_t, z_t)$ using **only two** action chunks generated by the frozen teacher from the same initial noise z_t under different denoising budgets. To exclude highly uncertain supervision, we define the first threshold-crossing time and the corresponding trajectory mask as

$$T_\tau := \inf\{t : U_t > \tau\}, \qquad m_t := \mathbf{1}\{t < T_\tau\}, \tag{13}$$

with $T_\tau = \infty$ if the threshold is never crossed. Only the prefix $t < T_\tau$ is retained for training; the threshold-crossing chunk and all subsequent chunks are masked out, even if their uncertainty later falls below the threshold.

On the retained chunks, we implement soft uncertainty penalization through an uncertainty-weighted log-probability surrogate. The student is updated by minimizing

$$\mathcal{L}_{\text{UA-OPD}}(\theta) := \widehat{\mathbb{E}}_{t,i}\left[m_t\left(\ell_{\text{OPD}}(s_t, z_t, i; \theta) + \lambda\, \text{sg}[U_t] \log p_{S,t,i}(\theta)\right)\right], \tag{14}$$

where $\log p_{S,t,i}(\theta)$ denotes the student Gaussian log-probability of the sampled denoising transition at interval i of chunk t, and sg denotes stop-gradient. The empirical expectation averages over student rollout samples and uniformly sampled denoising intervals, with uncertainty scores, masks, and sampled transitions held fixed during each update. This surrogate introduces an uncertainty-dependent correction within the retained prefix, complementing hard truncation of the remaining suffix.

5 Experiments

Our experiments address five key questions: (1) Does the uncertainty estimate used by UA-OPD reflect variation in teacher reliability on student-induced states? (2) Does UA-OPD improve the sample efficiency and task performance of on-policy distillation? (3) Does UA-OPD improve out-of-distribution generalization? (4) Does UA-OPD improve one-step policy distillation? (5) How do trajectory truncation and uncertainty penalization affect performance?

Benchmarks and Evaluation. We evaluate UA-OPD on LIBERO (Liu et al., 2023a) and RoboTwin 2.0 (Chen et al., 2026). For LIBERO, we use all four task suites, covering language-conditioned manipulation with variations in spatial layouts, object types, and task goals. For RoboTwin 2.0, we consider four dual-arm manipulation tasks: *Adjust Bottle*, *Beat Block Hammer*, *Blocks Ranking RGB*, and *Handover Block*. We report policy success rates over 500 evaluation rollouts per LIBERO suite and 100 evaluation rollouts per RoboTwin 2.0 task. The teacher-takeover evaluation follows a separate protocol described in Section 5.1. Training details are provided in Appendix B.

5.1 Teacher Reliability on Student-Induced States

We examine teacher reliability on student-induced states and its relationship with the uncertainty estimate used by UA-OPD. We assess the teacher's closed-loop performance through takeover evaluations and examine its reward discrepancy relative to a near-optimal reference.

Teacher-takeover reliability. We begin each evaluation episode with the initial student policy and probabilistically transfer control to the teacher during the rollout. At takeover, we record the teacher's self-consistency uncertainty at the current state, after which the teacher controls the robot for the remainder of the same episode. We relate the final task outcome to the uncertainty recorded at takeover, assessing whether the score predicts the teacher's ability to complete the task from a student-induced state. We conduct 500 takeover trials per RoboTwin 2.0 task and 2,000 trials per LIBERO suite.

Table 1: Teacher-takeover success rates partitioned by uncertainty. Gap denotes the difference between the low- and high-uncertainty groups.

Benchmark	Task	$\mathrm{SR}_{U\leq\tau}$	$\mathrm{SR}_{U>\tau}$	Gap
LIBERO	Spatial	0.974	0.931	0.043
	Object	0.985	0.974	0.011
	Goal	0.957	0.865	0.092
	Long	0.827	0.711	0.116
RoboTwin 2.0	Adjust Bottle	0.91	0.57	0.34
	Beat Block Hammer	0.85	0.65	0.20
	Blocks Ranking RGB	0.75	0.50	0.25
	Handover Block	0.39	0.08	0.31

Table 1 groups takeover trials by the uncertainty recorded at takeover, using analysis thresholds of $\tau = 0.02$ for RoboTwin 2.0 and $\tau = 0.01$ for LIBERO. Across all evaluated suites and tasks, trials with low uncertainty at takeover yield higher success rates than those with high uncertainty, with more pronounced separation on RoboTwin 2.0. On *Adjust Bottle*, for example, the success-rate gap reaches 34 percentage points. Figure 2a further shows that takeover success decreases consistently across increasing uncertainty intervals. These results support self-consistency uncertainty as an indicator of teacher reliability on states encountered during student execution.

Reward discrepancy. We further examine whether self-consistency uncertainty reflects the reward discrepancy underlying the pessimistic objective. On RoboTwin 2.0 *Adjust Bottle*, we compare a teacher with a lower performance ceiling against a near-optimal reference teacher, which serves as an empirical proxy for the inaccessible optimal policy $\pi^\star$. For each student-induced sample, we compute the empirical reward discrepancy $|\hat{r}(s,a) - r_{\mathrm{ref}}(s,a)|$, where $\hat{r}$ and r_{ref} denote the rewards induced by the lower-performing teacher and the reference teacher, respectively. We then relate this discrepancy to the self-consistency uncertainty of the lower-performing teacher. As shown in Figure 2b, uncertainty is positively associated with reward discrepancy, with a Spearman correlation of $\rho = 0.683$. Together, these results provide empirical support for using self-consistency uncertainty to guide supervision in UA-OPD.

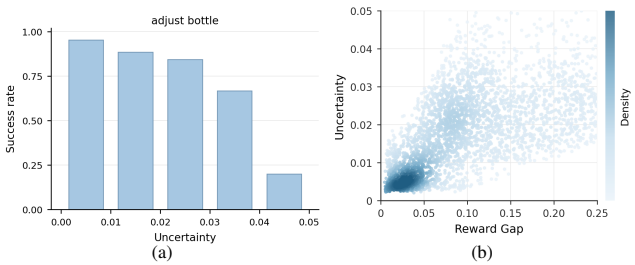


Figure 2: Self-consistency uncertainty on *Adjust Bottle*. **(a)** Teacher-takeover success rates across uncertainty intervals. **(b)** Relationship between self-consistency uncertainty and empirical reward discrepancy.

5.2 UA-OPD Enables Efficient and Effective On-Policy Distillation.

We evaluate the sample efficiency of UA-OPD by comparing its online learning curves with vanilla OPD (Agarwal et al., 2024) and PPO (Chen et al., 2025). As shown in Figure 3, UA-OPD reaches comparably high success rates with fewer training samples than vanilla OPD across LIBERO and RoboTwin 2.0. The learning curves also exhibit more stable improvement, with smaller early-stage performance drops and reduced fluctuations. On *Adjust Bottle*, for example, vanilla OPD undergoes an initial performance decline before recovering, whereas UA-OPD improves more steadily toward high success rates. PPO makes limited progress on this task, with rollouts retaining the initial policy's predominantly left-arm behavior rather than acquiring the right-arm behaviors needed for higher success.

Beyond sample efficiency, we evaluate the peak performance attained by the distilled policies. Table 2 reports the peak success rates of offline knowledge distillation (KD) (Hinton et al., 2015), OPD, and UA-OPD, with the initial student and teacher as references. Together with the learning

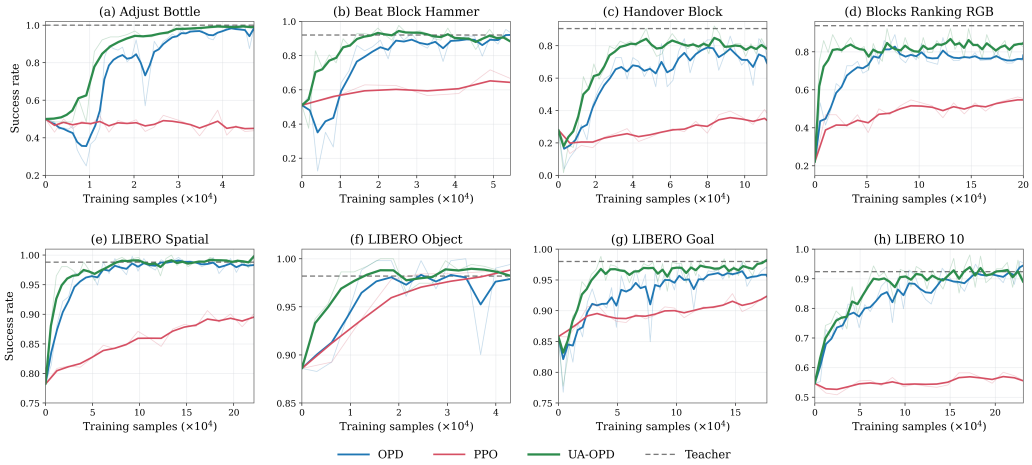


Figure 3: UA-OPD enables efficient adaptation on LIBERO and RoboTwin 2.0.

curves, these results show that UA-OPD improves sample efficiency over OPD while maintaining competitive performance across both benchmarks. Within LIBERO, the performance gain is most pronounced on LIBERO Long, the most challenging of the four evaluated suites. Since all distillation methods use the same teacher, these results support the value of incorporating an uncertainty penalty to reduce reliance on potentially unreliable teacher supervision and make more effective use of on-policy guidance. Notably, UA-OPD also surpasses the teacher on most evaluated suites and tasks, supporting the value of accounting for teacher reliability when incorporating on-policy supervision.

Table 2: UA-OPD enables effective adaptation on LIBERO and RoboTwin 2.0.

	LIBERO (%)					**RoboTwin 2.0 (%)**				
Method	S	O	G	L	**Avg.**	AB	BH	BR	HB	**Avg.**
Student	78.2	88.6	85.8	54.6	76.8	50	51	22	28	37.75
Teacher	98.8	98.2	98.0	92.4	96.9	100	92	93	90	93.75
KD	91.7	97.8	88.4	86.0	91.0	88	90	84	55	79.25
OPD	99.2	**100.0**	97.2	93.0	97.4	**100**	95	89	87	92.75
UA-OPD	**99.8**	**100.0**	**98.4**	**98.0**	**99.1**	**100**	**97**	**92**	**92**	**95.25**
Δ_T	+1.0	+1.8	+0.4	+5.6	+2.2	0	+5	−1	+2	+1.50

Bold indicates the best result among distillation methods; Δ_T denotes the difference relative to the teacher.

5.3 UA-OPD Improves Out-of-Distribution Generalization

We further evaluate whether the benefits of UA-OPD extend to environmental conditions not encountered during distillation. We distill policies with OPD and UA-OPD in the RoboTwin 2.0 *clean* setting and evaluate them in the *random* setting, which introduces background textures, task-irrelevant clutter, randomized illumination, and a tabletop-height offset. As shown in Table 3, UA-OPD consistently achieves higher success rates than vanilla OPD across all four evaluated tasks. These results suggest that accounting for teacher reliability during on-policy learning benefits not only adaptation within the distillation environment but also the transfer of learned behaviors under environmental shifts.

Table 3: UA-OPD improves out-of-distribution generalization.

Method	Adjust Bottle	Handover Block	Blocks Ranking RGB	Beat Block Hammer	Average
OPD	0.61	0.29	0.53	0.47	0.475
UA-OPD	**0.67**	**0.31**	**0.57**	**0.57**	**0.530**

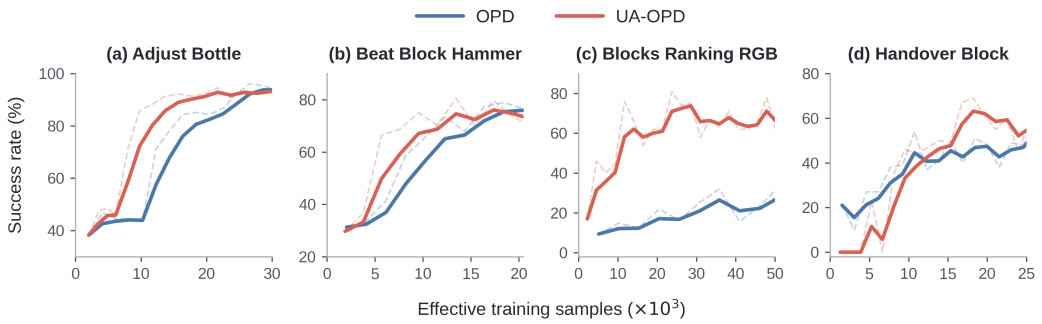


Figure 4: UA-OPD Improves One-Step Distillation.

5.4 UA-OPD Improves One-Step Distillation

Beyond policy adaptation, we evaluate UA-OPD for one-step distillation, which aims to improve inference efficiency by generating each action chunk with a single denoising step. We compare UA-OPD with OPD on four RoboTwin 2.0 tasks under the same single-step constraint. As shown in Figure 4, UA-OPD reaches comparable success rates with fewer effective training samples on *Adjust Bottle* and *Beat Block Hammer*, while attaining higher success rates on the remaining tasks. The gains are particularly pronounced on *Blocks Ranking RGB*, where UA-OPD improves rapidly and maintains a substantial performance advantage over OPD. These results suggest that accounting for teacher uncertainty can improve sample efficiency and policy performance even under a single-step denoising constraint.

5.5 Ablation Study

We examine the contributions of uncertainty-based trajectory truncation and soft uncertainty penalization on the RoboTwin 2.0 *Handover Block* task, where teacher reliability varies substantially across student-induced states. We compare four variants: OPD, UA-OPD without truncation, UA-OPD without penalization, and the full method. Each single-component variant removes either the trajectory mask or the soft penalty while retaining the other component. As shown in Figure 5, the full UA-OPD reaches high success rates with fewer training samples than vanilla OPD and either single-component variant. Without uncertainty penalization, the policy improves rapidly during early training but exhibits lower success rates over much of the subsequent learning process. Removing trajectory truncation leads to more gradual improvement, although this variant approaches the full method later in training. These observations suggest that combining trajectory truncation with soft uncertainty penalization improves adaptation efficiency while maintaining competitive policy performance.

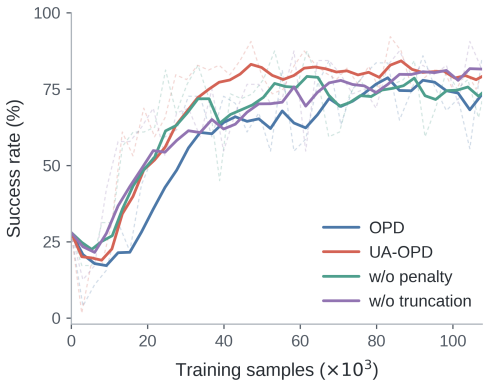


Figure 5: Ablation of trajectory truncation and uncertainty penalization on *Handover Block*.

6 Conclusion

This work studies the performance–reliability trade-off in on-policy distillation: querying the teacher on student-induced states provides useful adaptive supervision, but blindly trusting the teacher outside its reliable support can make distillation inefficient and lead to suboptimal adapted policies. UA-OPD addresses this trade-off by treating teacher uncertainty as a pessimistic correction. The resulting method provides a conservative objective for flow-based robotic policies and improves both adaptation efficiency and final performance, showing that reliable embodied OPD requires knowing when and how to trust the teacher.

AI USE STATEMENT

In this work, we used generative AI tools to assist with proof writing and to provide coding support for method implementation. We did not use these tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean or reformat datasets, support qualitative or thematic data analysis, or interpret results. Additionally, we used these tools to polish the language and improve the readability of manuscript text written by the authors, rather than to draft manuscript content.

The authors reviewed all AI-assisted content. Specifically, we checked the mathematical arguments in the proofs and reviewed and tested the AI-assisted code for correctness and consistency with the methods described in the paper. We also reviewed all language edits to ensure that they preserved the intended technical meaning. We take full responsibility for the final content of this work, including all text, claims, proofs, code, and other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies on-policy distillation for robotic manipulation and does not involve human subjects or personally identifiable information.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of our method, theoretical derivations, experimental protocols, and implementation settings in the main paper and appendix. To further facilitate reproducibility, we will release our complete training and evaluation code upon acceptance.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *12th International Conference on Learning Representations (ICLR'24), Vienna, Austria*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Conference on Robot Learning (CoRL'25), Seoul, South Korea*, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems XXI (RSS'25), Los Angeles, California, USA*, 2025b.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems XIX (RSS'23), Daegu, Republic of Korea*, 2023.

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Xiang Li, Quanlu Zhang, Zhaofei Yu, et al. π_{RL}: Online rl fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *43rd International Conference on Machine Learning (ICML'26), Seoul, South Korea*, 2026.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research (IJRR)*, 44(10-11):1684–1704, 2025.

Wojciech M Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In *The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS'19), Naha, Okinawa, Japan*, 2019.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *40th International Conference on Machine Learning (ICML'23), Honolulu, HI, USA*, 2023.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *12th International Conference on Learning Representations (ICLR'24), Vienna, Austria*, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: Robot manipulation with multimodal prompts. In *40th International Conference on Machine Learning (ICML'23), Honolulu, HI, USA*, 2023.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems XXI (RSS'25), Los Angeles, California, USA*, 2025a.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *8th Conference on Robot Learning (CoRL'24), Munich, Germany*, 2025b.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Yiming Cheng, Miles Yang, Zhao Zhong, and Liefeng Bo. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. In *The 19th European Conference on Computer Vision (ECCV'26), Malmö, Sweden*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS'23), New Orleans, Louisiana, USA*, 2023a.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di ZHANG, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. In *39th Conference on Neural Information Processing Systems (NeurIPS'25), San Diego, California, USA*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *11th International Conference on Learning Representations (ICLR'23), Kigali, Rwanda*, 2023b.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA'24), Yokohama, Japan*, 2024.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *Transactions on Machine Learning Research (TMLR'22)*, 2022.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *14th International Conference on Artificial Intelligence and Statistics (AISTATS'11), Fort Lauderdale, Florida, USA*, 2011.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*, 2026.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. In *43rd International Conference on Machine Learning (ICML'26), Seoul, South Korea*, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems XX (RSS'24), Delft, Netherlands*, 2024.

Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency flow matching: Defining straight flows with velocity consistency. *arXiv preprint arXiv:2407.02398*, 2024.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV'19), Seoul, Korea (South)*, 2019.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *43rd International Conference on Machine Learning (ICML'26), Seoul, South Korea*, 2026.

Zhide Zhong, Haodong Yan, Junfeng Li, Junjie He, Tianran Zhang, and Haoang Li. Vla-opd: Bridging offline sft and online rl for vision-language-action models via on-policy distillation. *arXiv preprint arXiv:2603.26666*, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *7th Conference on Robot Learning (CoRL'23), Atlanta, Georgia, USA*, 2023.

A PROOFS

A.1 PROOF OF PROPOSITION 4.1

Let $\bar{\pi}$ be an arbitrary global minimizer of the OPD objective:

$$\bar{\pi} \in \arg\min_{\pi \in \Pi} \mathcal{L}_{\mathrm{OPD}}^{\pi^\star}(\pi).$$

Since the KL divergence is non-negative, the objective is non-negative for every $\pi \in \Pi$:

$$\mathcal{L}_{\mathrm{OPD}}^{\pi^\star}(\pi) \geq 0.$$

Moreover, because $\pi^\star \in \Pi$, its objective value is zero:

$$\mathcal{L}_{\mathrm{OPD}}^{\pi^\star}(\pi^\star) = \bar{\mathbb{E}}_{s \sim d_P^{\pi^\star}} \left[D_{\mathrm{KL}} \left(\pi^\star(\cdot|s) \| \pi^\star(\cdot|s) \right) \right] = 0.$$

Therefore, the global minimum of $\mathcal{L}_{\mathrm{OPD}}^{\pi^\star}$ is zero, so:

$$\mathcal{L}_{\mathrm{OPD}}^{\pi^\star}(\bar{\pi}) = 0.$$

Let μ_t^π denote the state distribution induced by policy π at time t. Expanding the discounted expectation gives:

$$\mathcal{L}_{\mathrm{OPD}}^{\pi^\star}(\bar{\pi}) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{s \sim \mu_t^{\bar{\pi}}} \left[D_{\mathrm{KL}} \left(\bar{\pi}(\cdot|s) \| \pi^\star(\cdot|s) \right) \right] = 0.$$

Since every term is non-negative and $\gamma^t > 0$, the expectation is zero for every t:

$$\mathbb{E}_{s \sim \mu_t^{\bar{\pi}}} \left[D_{\mathrm{KL}} \left(\bar{\pi}(\cdot|s) \| \pi^\star(\cdot|s) \right) \right] = 0.$$

The non-negativity of the KL divergence therefore implies that the policies agree almost everywhere under each state distribution:

$$\bar{\pi}(\cdot|s) = \pi^\star(\cdot|s) \quad \text{for } \mu_t^{\bar{\pi}}\text{-almost every } s.$$

We next show by induction that $\bar{\pi}$ and $\pi^\star$ induce the same state distribution at every time step. At $t = 0$, both policies start from ρ_0:

$$\mu_0^{\bar{\pi}} = \mu_0^{\pi^\star}.$$

For the induction hypothesis, suppose that:

$$\mu_t^{\bar{\pi}} = \mu_t^{\pi^\star}.$$

For any measurable set $B \subseteq \mathcal{S}$:

$$\begin{aligned}
\mu_{t+1}^{\bar{\pi}}(B) &= \int_{\mathcal{S}} \mu_t^{\bar{\pi}}(ds) \int_{\mathcal{A}} \bar{\pi}(da|s) P(B|s,a) \\
&= \int_{\mathcal{S}} \mu_t^{\bar{\pi}}(ds) \int_{\mathcal{A}} \pi^\star(da|s) P(B|s,a) \\
&= \int_{\mathcal{S}} \mu_t^{\pi^\star}(ds) \int_{\mathcal{A}} \pi^\star(da|s) P(B|s,a) \\
&= \mu_{t+1}^{\pi^\star}(B). \qquad (15)
\end{aligned}$$

The second equality follows from the almost-everywhere agreement between $\bar{\pi}$ and $\pi^\star$ under $\mu_t^{\bar{\pi}}$, while the third follows from the induction hypothesis. Thus, the next-state distributions agree:

$$\mu_{t+1}^{\bar{\pi}} = \mu_{t+1}^{\pi^\star}.$$

By induction, the two policies induce the same state distribution at every time step. Together with their almost-everywhere agreement on these states, they also induce the same state-action distribution at every time step. Consequently, their environment returns are equal:

$$\begin{aligned} J_{\mathcal{E}}(\bar{\pi}) &= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\substack{s\sim\mu_t^{\bar{\pi}} \\ a\sim\bar{\pi}(\cdot|s)}} [r_{\text{env}}(s,a)] \\ &= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\substack{s\sim\mu_t^{\pi^\star} \\ a\sim\pi^\star(\cdot|s)}} [r_{\text{env}}(s,a)] \\ &= J_{\mathcal{E}}(\pi^\star). \end{aligned} \tag{16}$$

Since $\pi^\star$ maximizes $J_{\mathcal{E}}$ over Π, $\bar{\pi}$ is also optimal in $\mathcal{E}$. Equivalently:

$$\arg\min_{\pi\in\Pi} \mathcal{L}_{\text{OPD}}^{\pi^\star}(\pi) \subseteq \arg\max_{\pi\in\Pi} J_{\mathcal{E}}(\pi).$$

This completes the proof.

A.2 Proof of Theorem 4.2

Let $\pi \in \Pi$ be arbitrary. By Eq. (6) and the definition $\widetilde{r} = \hat{r} - u$, we have:

$$r(s,a) - 2u(s,a) \le \widetilde{r}(s,a) \le r(s,a) \tag{17}$$

Adding the common entropy regularization term and taking the discounted expectation under ρ_P^π yields Eq. (8).

Now let $\pi \in \Pi_\epsilon$. Using Eq. (8), the δ-optimality of $\widetilde{\pi}$, and $\epsilon_u(\pi) \le \epsilon$, we obtain

$$\begin{aligned} \eta_{\mathcal{M}}(\widetilde{\pi}) &\ge \eta_{\widetilde{\mathcal{M}}}(\widetilde{\pi}) \\ &\ge \sup_{\pi'\in\Pi} \eta_{\widetilde{\mathcal{M}}}(\pi') - \delta \\ &\ge \eta_{\widetilde{\mathcal{M}}}(\pi) - \delta \\ &\ge \eta_{\mathcal{M}}(\pi) - 2\epsilon_u(\pi) - \delta \\ &\ge \eta_{\mathcal{M}}(\pi) - 2\epsilon - \delta. \end{aligned} \tag{18}$$

Taking the supremum over $\pi \in \Pi_\epsilon$ proves Eq. (9).

B Implementation and Experimental Details

B.1 Policies and Initialization

Demonstration data and trainable parameters. Table 4 specifies the demonstration budgets used for student and teacher supervised fine-tuning (SFT). On RoboTwin, the Adjust Bottle student uses one demonstration and its teacher uses 50; the other three tasks use 50 demonstrations for the student and 200 for the teacher. LIBERO uses a one-shot student and a full-shot teacher, with full-shot denoting the complete available training demonstration set. During distillation, the teacher is frozen and the student updates only its action expert, with the vision-language backbone frozen. KD, OPD, and UA-OPD use the same teacher for each task.

Policy inputs and actions. Both policies use the flow-based $\pi_{0.5}$ architecture (Black et al., 2025a), conditioned on RGB images, proprioception, and a language instruction. LIBERO uses main- and wrist-camera views with a 7-dimensional action interface. RoboTwin uses the ALOHA-AgileX embodiment, three RGB views (head and two wrists), and a 14-dimensional action interface. Observation preprocessing and action normalization follow the corresponding checkpoint transforms. The executed chunk length is $H = 10$ for LIBERO and $H = 20$ for RoboTwin. Losses and uncertainty scores use only these executed actions and valid action coordinates, excluding padding.

B.2 Training Configuration

Table 4 gives the initialization data and benchmark-level training settings. The thresholds match the uncertainty partitions in Section 5.1: $\tau = 0.01$ for LIBERO and $\tau = 0.02$ for RoboTwin.

Table 4: SFT demonstration budgets and UA-OPD hyperparameters. Student and teacher columns give demonstration counts; Full denotes the full training demonstration set. Batch is the global optimization batch, and H is the executed action-chunk length.

Benchmark / task	Student	Teacher	τ	λ	Batch	H
LIBERO (all four suites)	1-shot	Full-shot	0.01	0.01	512	10
RoboTwin Adjust Bottle	1	50	0.02	0.02	512	20
RoboTwin Beat Block Hammer	50	200	0.02	0.02	512	20
RoboTwin Blocks Ranking RGB	50	200	0.02	0.02	512	20
RoboTwin Handover Block	50	200	0.02	0.02	512	20

B.3 Uncertainty and Loss Implementation

We use $p = 1$ in Eqs. (10)–(11) and teacher denoising budgets $\mathcal{K} = \{10, 5\}$. The first-crossing mask in Eq. (13) resets at each episode boundary and excludes the threshold-crossing chunk and its entire suffix. The base distillation loss and soft penalty share this mask and are averaged over retained entries. For the penalty in Eq. (14), Gaussian log-probabilities are averaged over valid action coordinates and chunk positions; uncertainty, masks, and the sampled transition in the log-density are detached.

B.4 Task and Environment Visualizations

Figures 6 and 7 illustrate the four RoboTwin tasks and the environmental shifts used for clean-to-random evaluation.

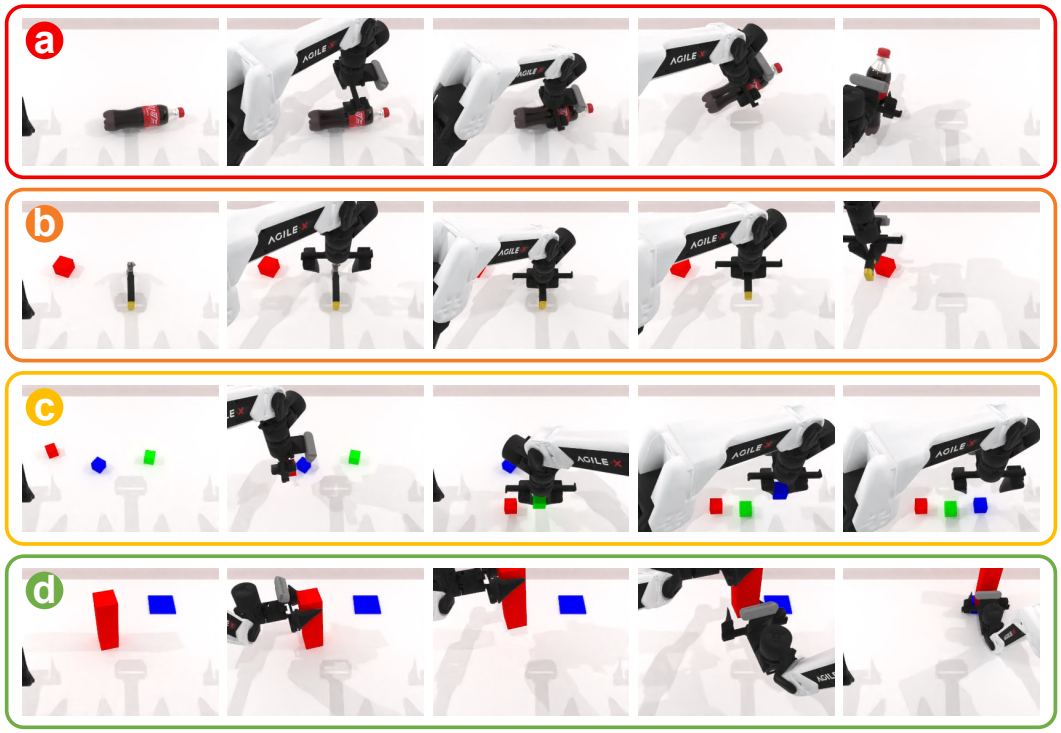


Figure 6: RoboTwin task sequences: (a) Adjust Bottle, (b) Beat Block Hammer, (c) Blocks Ranking RGB, and (d) Handover Block. Each row shows five chronologically ordered head-camera frames from one episode.

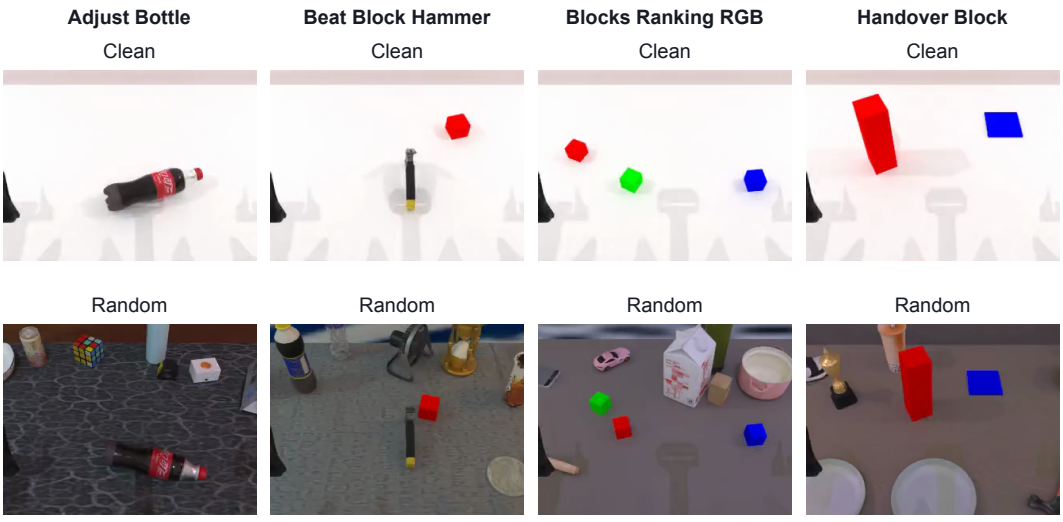


Figure 7: Clean and randomized RoboTwin environments. Each column shows one task using the head-camera view. Random scenes vary background and tabletop appearance, distractor objects, and illumination. Clean and random examples are independently sampled scenes.