TILDE: TEACHER-IN-THE-LOOP DISTILLATION FROM EMBODIED FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

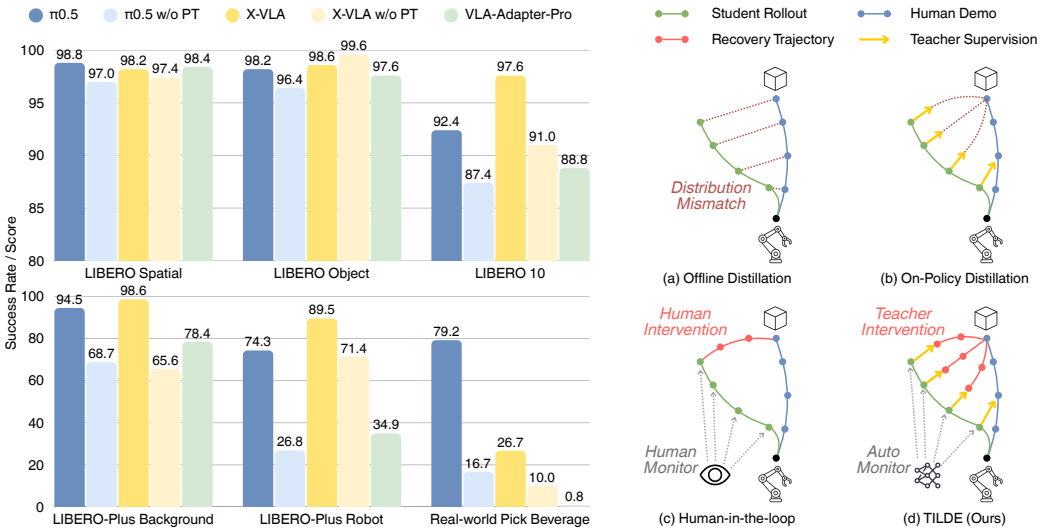

Figure 1: **Left:** Policies with ($\pi_{0.5}$ & X-VLA) and without large-scale action pretraining ("w/o PT" & VLA-Adapter-Pro) perform similarly on standard benchmarks (LIBERO-Spatial & Object), while larger gaps emerge on long-horizon tasks (LIBERO-10), under perturbations (LIBERO-Plus), and on the real-world Pick Beverage task. These results motivate transferring capabilities from pretrained EFMs to student policies without repeating action pretraining. **Right:** (a) Offline distillation is restricted to the demonstration distribution, leading to distribution mismatch during student execution. (b) On-Policy Distillation extends supervision to student-induced states but does not execute teacher corrections to observe the subsequent recovery states. (c) Human-in-the-loop methods collect recovery trajectories through human monitoring and intervention, but their coverage is limited by the bandwidth of human supervision. (d) TILDE jointly learns from demonstration, student-induced rollout, and failure recovery distributions, with AutoDAgger automating progress monitoring and teacher intervention for efficient data collection.

ABSTRACT

Large-scale action pretraining equips Embodied Foundation Models (EFMs) with strong manipulation capabilities, with particular strengths in generalization and closed-loop robustness, but reproducing such pretraining requires substantial robot data and computation. To transfer these capabilities from pretrained EFMs to policies without action pretraining, we introduce TILDE (**T**eacher-**I**n-the-**L**oop **D**istillation from **E**FMs), which synergizes on-policy distillation at student-induced states with learning from teacher-executed recovery trajectories. We further propose AutoDAgger, an autonomous data aggregation pipeline in which a robotic reward model monitors task progress and triggers teacher intervention, thereby collecting recovery experience without human monitoring or manual control. AutoDAgger accelerates recovery-trajectory collection by more than $2\times$ compared with the human-in-the-loop method. By unifying learning from demonstrations, on-policy teacher supervision, and teacher's corrective behavior, TILDE improves task performance by up to $4.1\times$ over behavior cloning across three real-world tasks and supports effective cross-architecture distillation.

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

Embodied Foundation Models (EFMs), including Vision-Language-Action (VLA) models (Zhang et al., 2025a; Shao et al., 2025; Ma et al., 2024; Sapkota et al., 2025), have demonstrated remarkable manipulation capabilities and generalization. However, these capabilities often rely on action pretraining on large-scale robot datasets (Bu et al., 2025; Padalkar et al., 2024; Walke et al., 2023; Khazatsky et al., 2024), making it costly to endow new policy architectures, particularly efficient models, with comparable capabilities (Yu et al., 2025; Guan et al., 2025). Reproducing foundation-scale pretraining requires access to sufficiently diverse, high-quality robot data and substantial computational resources, which is prohibitively expensive in many research and deployment settings.

Policies without large-scale action pretraining (Lin et al., 2025; Wang et al., 2026a; Community, 2026) can still achieve competitive performance on standard simulation benchmarks such as LIBERO-Spatial and LIBERO-Object (Liu et al., 2023). However, our pilot study suggests that this apparent parity does not consistently extend to more challenging settings. As shown in Figure 1, substantially larger gaps emerge on long-horizon tasks in LIBERO-10, under visual and robot-initial-state perturbations (Fei et al., 2025), and in real-world tasks. These observations are consistent with action pretraining providing benefits beyond nominal action prediction, particularly improved closed-loop robustness under challenging and out-of-distribution conditions. This motivates a natural question: *Can such capabilities be transferred from a pretrained EFM to a student policy without repeating expensive action pretraining?*

Policy distillation offers an intuitive route by transferring teacher knowledge in a task-specific domain (Rusu et al., 2015; Dong et al., 2025; Zhang et al., 2025b). However, for closed-loop policies, effective distillation depends not only on *what* actions are supervised, but also on *where* supervision is provided. When distillation is restricted to a fixed demonstration dataset (Figure 1(a)), supervision remains concentrated on the nominal distribution. During closed-loop deployment, compounding student errors and environmental stochasticity can drive the policy into out-of-distribution states. This distribution mismatch between training and inference leaves the student insufficiently prepared to handle execution deviations (Yan et al., 2026; Hu et al., 2026).

Interactive imitation learning mitigates this mismatch by querying expert supervision at states induced by the learner (Ross et al., 2011). In On-Policy Distillation (OPD) (Agarwal et al., 2024; 2023), the student interacts with the environment while the teacher provides action targets at student-visited states (Figure 1(b)). This expands supervision to student-induced rollout states, but the queried teacher actions are not executed (Zhong et al., 2026; Wan et al., 2026). Consequently, OPD does not observe the subsequent states that would arise following the teacher's correction. Human-in-the-loop approaches such as HG-DAgger (Kelly et al., 2019) (Figure 1(c)) can collect recovery trajectories but require continuous human monitoring and teleoperated takeover (Murray et al., 2026; Xu et al., 2025), severely limiting data scalability.

To address these limitations, we introduce TILDE (**T**eacher-**I**n-the-**L**oop **D**istillation from **E**FMs), a policy distillation framework that transfers teacher knowledge over three complementary data distributions: human demonstration, student-induced rollouts, and recovery trajectories (Figure 1(d)). Central to TILDE is AutoDAgger, an autonomous teacher-in-the-loop data aggregation pipeline. An off-the-shelf robotic reward model monitors student execution and triggers intervention, after which the EFM teacher directly takes control and executes a recovery trajectory toward task completion. Automated progress monitoring removes the need for expensive human oversight, while teacher policy takeover avoids the reaction and teleoperation overhead of human intervention, substantially enhancing recovery-data collection throughput.

Given the aggregated data, TILDE applies supervision according to the sample source. Offline demonstrations and teacher recovery trajectories provide both behavior-cloning targets from executed actions and teacher distillation targets, while student rollout states use on-policy teacher supervision only. Importantly, all distillation is performed directly in the common action space and requires no alignment of internal representations, allowing the framework to accommodate heterogeneous teacher and student architectures.

Our contributions are summarized as follows:

1. We formulate policy distillation from a distributional perspective and propose the TILDE framework, which jointly transfers teacher knowledge over offline demonstrations, student-induced rollouts, and teacher-generated recovery experience.
2. We introduce AutoDAgger, an autonomous teacher-in-the-loop data aggregation pipeline that expands the training distribution through automatic task monitoring and teacher takeover, enabling efficient recovery-data collection while bypassing expensive human involvement.
3. Extensive experiments on real-world manipulation tasks demonstrate that TILDE improves student performance by up to $4.1\times$ over behavior cloning, consistently outperforms offline and on-policy distillation, and enables effective cross-architecture capability transfer.

2 Related Work

Embodied Foundation Models. Recent advances in large-scale robot learning have given rise to Embodied Foundation Models (EFMs) that integrate visual perception, language understanding, and low-level control, showing impressive performance across a broad spectrum of dexterous tasks and improved generalization to diverse environments (Zhang et al., 2025a; Shao et al., 2025; Ma et al., 2024; Xu et al., 2024b). Vision-Language-Action (VLA) models directly map multimodal observations and language instructions to robot actions (Intelligence et al., 2025a; 2026; Bjorck et al., 2025; Zheng et al., 2026). An emerging alternative is the World Action Model (WAM), which jointly models future world states and robot actions, rather than relying solely on reactive observation-to-action prediction (Ye et al., 2026; Li et al., 2026; Yuan et al., 2026; Wang et al., 2026b). Despite their strong capabilities, training such foundation models from scratch requires substantial data and computational resources. Our work instead studies how the capabilities of a pretrained EFM can be transferred to a student policy without repeating foundation-scale pretraining.

Policy Distillation. Knowledge Distillation (KD) transfers predictions or representations from a teacher to a smaller student and is widely used for model compression and capability transfer (Hinton et al., 2015; Buciluǎ et al., 2006). Policy distillation extends this paradigm to robot learning (Rusu et al., 2015; Torne et al., 2024; Dong et al., 2025; Shao et al., 2026; Zhang et al., 2025b), with recent advances leveraging foundation models as teachers (Jülg et al., 2025). Existing methods perform distillation through model-level alignment (Jeon et al., 2026; Ye et al., 2025) or teacher-generated data and supervision (Shi et al., 2026; Xiao et al., 2026; Xu et al., 2024a). On-Policy Distillation (OPD) further queries teacher supervision on student-induced states rather than relying solely on a fixed offline dataset (Agarwal et al., 2024; 2023), providing corrective signals where the student tends to encounter uncertainty or make mistakes (Zhong et al., 2026; Wan et al., 2026). Our method extends OPD by executing teacher takeover after intervention, thereby collecting the subsequent recovery trajectory and providing direct supervision for returning from off-nominal states to successful task execution.

Interactive Imitation Learning. Imitation learning suffers from distribution shift when the learned policy encounters states absent from expert demonstrations. Interactive imitation learning mitigates this issue by querying an expert on learner-induced states and iteratively aggregating the resulting supervision (Ross et al., 2011). Subsequent methods reduce expert queries using predicted novice–expert discrepancy (Hoque et al., 2021b), policy uncertainty (Menda et al., 2019; Kelly et al., 2019; Zurbrügg et al., 2026), or estimates of state novelty and task risk (Cronrath et al., 2018; Hoque et al., 2021a). However, these approaches generally depend on policy-specific techniques or continuous human oversight, and still request human teleoperation for corrective takeover on physical robots (Murray et al., 2026; Xu et al., 2025). In contrast, we propose a novel data aggregation framework that automatically detects intervention needs and executes corrective actions, substantially reducing the human-effort bottleneck in interactive data collection.

3 Methodology

3.1 Preliminaries

Task Formulation. We consider standard language-conditioned robotic manipulation with action-chunking policies (Chi et al., 2023). Given an observation state s comprising RGB images and robot

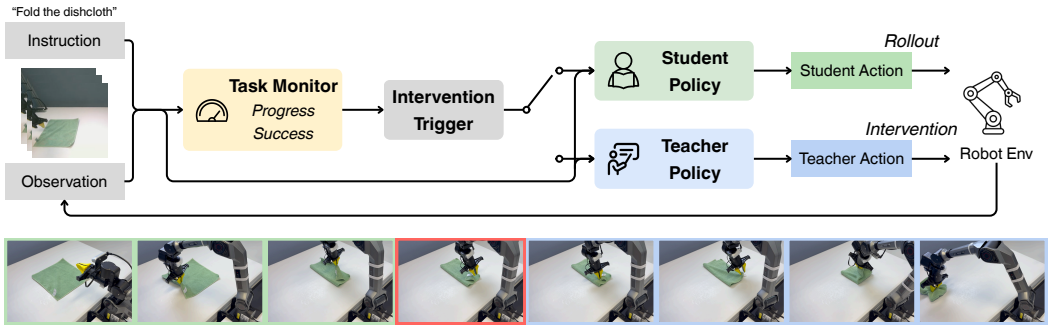

Figure 2: **AutoDAgger pipeline.** A reward-model-based task monitor tracks progress and success during student execution and triggers intervention upon sustained stagnation. The EFM teacher then takes over from the current state and executes a recovery trajectory. AutoDAgger records the pre-intervention student rollout and successful teacher recovery without human monitoring or takeover. **Bottom:** A real-world Fold Dishcloth example; green/blue denote student/teacher execution and red marks the intervention point.

proprioception, together with a language instruction $\boldsymbol{l}$, a policy π predicts an action chunk:

$$\pi(\boldsymbol{s}, \boldsymbol{l}) = \boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_H], \tag{1}$$

where H denotes the policy's prediction horizon and $\boldsymbol{a}_h$ is the h-th action in the chunk. During closed-loop execution, the policy is repeatedly queried using updated observations.

DAgger. Dataset Aggregation (DAgger) (Ross et al., 2011) is an interactive imitation learning algorithm that addresses the distribution shift between offline demonstrations $\mathcal{D}_{\text{demo}}$ and learner rollouts $\mathcal{D}_{\text{roll}}$ by collecting expert action labels at learner-induced states. In the representative human-in-the-loop variants, the human expert monitors the execution progress and takes control when the learner's behavior requires correction by providing recovery actions $\boldsymbol{a}^\star$ at the encountered observation $\boldsymbol{s}$. The resulting labeled samples $(\boldsymbol{s}, \boldsymbol{l}, \boldsymbol{a}^\star)$ are added to an aggregated dataset $\mathcal{D}_{\text{rec}}$, which is used to update the learner through behavior cloning.

3.2 Overview

TILDE transfers task-relevant capabilities from a capable and robust teacher policy π_T to a student policy π_S. With a base human demonstration dataset $\mathcal{D}_{\text{demo}}$, the teacher policy is finetuned from the pretrained EFM on the dataset to adapt to specific embodiment and target task, and remains frozen in the framework. The student policy is initialized from a pretrained backbone (*e.g.*, VLM in a VLA model) but receives no robot action pretraining, and is trained on $\mathcal{D}_{\text{demo}}$ using standard behavior cloning. TILDE consists of two stages: (1) In the AutoDAgger data collection, the student executes the task while a task monitor estimates progress and triggers teacher intervention when needed. The teacher takes over to provide recovery actions, which are combined with the student rollouts and added to the aggregated dataset. (2) During action distillation, the student learns from teacher supervision on student-visited states, together with corrective trajectories and existing demonstrations, combining knowledge distillation with imitation learning of recorded expert behaviors.

3.3 AutoDAgger

To reduce human supervision and collect informative correction data more efficiently, we propose AutoDAgger, an autonomous teacher-in-the-loop pipeline that monitors student execution, detects task-level failures, triggers teacher takeover, and retains successfully recovered trajectories for subsequent policy improvement.

Task Monitor. We employ a general-purpose robotic reward model R as a policy-independent task monitor. At each monitoring step t, we sample $K_t = \min(t, N_{\text{frame}})$ observations from the visual history, denoted by $\boldsymbol{o}_{1:K_t}$, and combine them with the language instruction $\boldsymbol{l}$ to predict progress and

success-probability sequences:

$$(\hat{\boldsymbol{p}}_{1:K_t}, \hat{\boldsymbol{q}}_{1:K_t}) = R\left(\boldsymbol{o}_{1:K_t}, \boldsymbol{l}\right). \tag{2}$$

We use the latest predictions and clip them to the valid range:

$$p_t = \mathrm{clip}(\hat{\boldsymbol{p}}_{1:K_t}[-1], 0, 1), \quad q_t = \mathrm{clip}(\hat{\boldsymbol{q}}_{1:K_t}[-1], 0, 1). \tag{3}$$

Here, p_t and q_t denote current task progress and success probability, respectively.

Intervention Trigger. Based on the task-monitor predictions, we trigger teacher intervention when student execution exhibits sustained progress stagnation while the task is not yet successful. To mitigate the effect of small fluctuations in predicted progress, we maintain a reference progress value $\bar{p}_t$. We register meaningful progress only when the current estimate exceeds the previous reference by a margin ϵ_p:

$$\bar{p}_t = \begin{cases} p_t, & p_t > \bar{p}_{t-1} + \epsilon_p, \\ \bar{p}_{t-1}, & \text{otherwise.} \end{cases} \tag{4}$$

We further maintain a timestamp τ_{valid} recording the most recent evidence that task execution remains viable. It is reset to the current time τ_t whenever meaningful progress is observed or the success probability exceeds a threshold θ_s:

$$\tau_{\text{valid}} \leftarrow \tau_t \quad \text{if} \quad (p_t > \bar{p}_{t-1} + \epsilon_p) \ \vee \ (q_t > \theta_s). \tag{5}$$

The resulting stagnation duration is

$$\Delta_t = \tau_t - \tau_{\text{valid}}. \tag{6}$$

Resetting τ_{valid} when task success is predicted prevents a completed execution, whose progress is no longer expected to increase, from being incorrectly identified as stalled. If the success probability subsequently falls below the threshold and no further progress is observed, the stagnation duration begins accumulating from the most recent reset time.

An intervention is triggered when progress has remained stagnant for at least T_{fail} seconds and the latest success probability does not exceed θ_s:

$$F_t = \mathbb{I}[(\Delta_t \geq T_{\text{fail}}) \wedge (q_t \leq \theta_s)]. \tag{7}$$

Requiring both conditions makes the trigger robust to transient fluctuations in either progress or success predictions.

Teacher Takeover and Recovery Verification. When $F_t = 1$, AutoDAgger immediately stops student execution and transfers control to the teacher policy. An analogous task monitor continues to evaluate the teacher during recovery. A recovery is considered successful only when $q_t > \theta_s$ is maintained continuously for a predefined duration T_{succ}. Any prediction at or below the threshold resets the success timer. If the teacher fails to recover the task, the trajectory is discarded. Otherwise, the combined student rollout and teacher recovery trajectory is retained and aggregated for subsequent policy training.

3.4 Distillation Objective

Given the trajectories collected by AutoDAgger, TILDE trains the student from three complementary sources of experience: offline demonstrations $\mathcal{D}_{\text{demo}}$, states visited during student rollouts $\mathcal{D}_{\text{roll}}$, and teacher-executed recovery trajectories $\mathcal{D}_{\text{rec}}$. For an observation-instruction pair $(\boldsymbol{s}, \boldsymbol{l})$, we perform distillation directly in the action space using

$$\mathcal{L}_{\text{KD}}(\boldsymbol{s}, \boldsymbol{l}) = \ell_S\left(\boldsymbol{s}, \boldsymbol{l}; \mathrm{sg}[\boldsymbol{A}_T]\right), \tag{8}$$

where $\mathrm{sg}[\cdot]$ stops gradients through the frozen teacher and ℓ_S denotes the original action-generation objective used by the student, instantiated with the teacher-predicted action chunk $\boldsymbol{A}_T = \pi_T(\boldsymbol{s}, \boldsymbol{l})$ as the target. Since supervision is imposed only on action outputs, TILDE does not require the teacher and student to share similar architectures or intermediate representations.

For samples from either offline demonstrations or recovery trajectories, the action actually executed in the environment is available. Let $\boldsymbol{A}_E$ denote this recorded action chunk. We therefore combine behavior cloning from the executed action with distillation from the teacher prediction:

$$\mathcal{L}_{\text{exec}}(\boldsymbol{s}, \boldsymbol{l}, \boldsymbol{A}_E) = \ell_S(\boldsymbol{s}, \boldsymbol{l}; \boldsymbol{A}_E) + \lambda_{\text{KD}} \mathcal{L}_{\text{KD}}(\boldsymbol{s}, \boldsymbol{l}), \quad (\boldsymbol{s}, \boldsymbol{l}, \boldsymbol{A}_E) \in \mathcal{D}_{\text{demo}} \cup \mathcal{D}_{\text{rec}}, \tag{9}$$

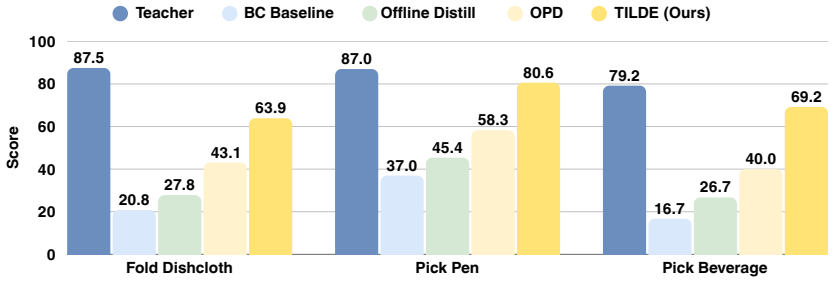

Figure 3: **Comparison of real-world task performance.** TILDE is compared with behavior cloning (BC), offline distillation, and on-policy distillation (OPD), with the teacher policy serving as a reference. TILDE outperforms all student baselines on all three tasks.

where λ_{KD} is the weight of distillation loss. In contrast, samples from $\mathcal{D}_{\mathrm{roll}}$ contain states visited by the student, thus, we apply only the distillation objective:

$$\mathcal{L}_{\mathrm{roll}}(\boldsymbol{s}, \boldsymbol{l}) = \lambda_{\mathrm{KD}} \mathcal{L}_{\mathrm{KD}}(\boldsymbol{s}, \boldsymbol{l}), \quad (\boldsymbol{s}, \boldsymbol{l}) \in \mathcal{D}_{\mathrm{roll}}. \tag{10}$$

Overall, the training loss is selected according to the source of each sample. This formulation exposes the student to three complementary data distributions. Offline demonstrations provide supervision on in-domain expert behavior, student rollouts provide teacher corrections at student-induced states, and recovery trajectories supply the subsequent observations induced by executing the teacher's corrections.

4 Experiments

4.1 Experimental Setup

Tasks and Evaluation. We conduct real-world experiments with 7-DoF AgileX PiperX robot arms on three manipulation tasks: (1) "Fold Dishcloth": folding a dishcloth in half twice and placing it at a target location toward the front left, testing deformable object manipulation; (2) "Pick Pen": picking up one to three pens and placing them in a cup, testing precise manipulation across object locations; and (3) "Pick Beverage": picking up a beverage and placing it in a paper box, focusing on generalization across objects and locations. We collect 200 human demonstration episodes for Fold Dishcloth and 100 for each of the other two tasks.

We evaluate each policy on the same predefined test cases, with fixed object types, positions, and orientations: 12 trials for Fold Dishcloth, 27 for Pick Pen, and 30 for Pick Beverage. We define fine-grained evaluation metrics to enable a more precise assessment of policy performance, with the detailed scoring criteria provided in the Appendix A.2. Scores are averaged across sub-steps and evaluation trials, while the success rate (SR), which measures complete task success, is reported as a complementary metric.

Implementation Details. We obtain the teacher by finetuning the pretrained $\pi_{0.5}$ VLA model (Intelligence et al., 2025b) on the human demonstrations for each task. Unless otherwise specified, the student employs the same architecture as $\pi_{0.5}$ but receives no action pretraining: its VLM backbone is initialized from PaliGemma (Beyer et al., 2024), while the action expert is trained from scratch. The task monitor in AutoDAgger leverages the off-the-shelf robotic reward model Robometer (Liang et al., 2026) without task-specific finetuning.

Baselines and Training Protocol. We compare TILDE with three training paradigms: standard behavior cloning (BC) (Bain & Sammut, 1995; Pomerleau, 1988), offline distillation, and On-Policy Distillation (OPD). BC learns from recorded human actions, offline distillation additionally uses teacher action targets on demonstration states, and OPD further incorporates teacher supervision at student-rollout states. TILDE unifies learning from three complementary data distributions. Unless otherwise stated, comparisons use a shared pool of collected AutoDAgger data, containing 30 successfully recovered and retained episodes per task.

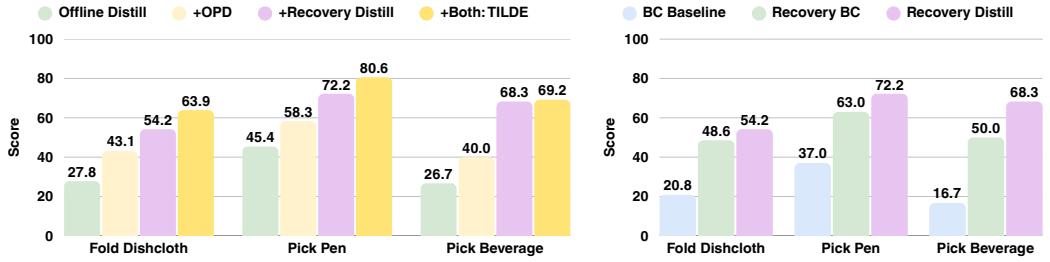

Figure 4: **Ablation study of training data and supervision.** **Left:** Starting from offline distillation, we add teacher supervision at student-rollout data (+OPD), supervision on teacher recovery trajectories (+Recovery Distill), or both (TILDE). Combining both sources gives the highest scores on all three tasks. **Right:** Recovery BC augments demonstration BC with imitation of recorded recovery actions. Recovery Distill additionally incorporates distillation on both demonstration and recovery samples, further improving performance.

Table 1: **Standardized failure recovery evaluation.** "Recovery SR" is measured from the same set of 12 predefined error states on Fold Dishcloth.

Method	Recovery SR
Teacher	9/12
BC Baseline	4/12
Offline Distill	5/12
OPD	5/12
TILDE (Ours)	**8/12**

Table 2: **Scaling with AutoDAgger data on Fold Dishcloth.** Zero AutoDAgger episodes corresponds to offline distillation.

#Episodes	Score	SR (%)
0	27.8	25.0
30	63.9	58.3
60	65.3	66.7
90	77.8	75.0

4.2 Main Results

Figure 3 shows that offline distillation yields only modest improvements over BC, as teacher supervision remains confined to the demonstration distribution. OPD yields further gains by mitigating distribution mismatch through adding teacher supervision at student-rollout states. TILDE achieves scores of 63.9, 80.6, and 69.2 on Fold Dishcloth, Pick Pen, and Pick Beverage, respectively, outperforming all evaluated baselines on each task. These scores are approximately $3.07\times$, $2.18\times$, and $4.14\times$ those of BC baseline, substantially narrowing the observed student–teacher performance gap. These consistent improvements support the value of learning from recovery data in addition to teacher action targets at student-induced states.

4.3 Analysis

Ablation Study. We first examine the contributions of student-rollout supervision and teacher-executed recovery trajectories using offline distillation as the common baseline. As shown in Figure 4 (left), adding either source improves performance on all three tasks, while combining both produces the highest scores. These results indicate that each source improves performance when the other is already present, supporting their complementary roles in transferring teacher behavior.

Figure 4 (right) further analyzes the supervision used on demonstrations and recovery trajectories. Recovery BC adds imitation of the teacher's recorded actions to conventional BC and improves performance on all three tasks. Recovery Distill additionally applies knowledge distillation to both demonstration and recovery samples, achieving further advantages.

Recovery from Failures. To directly assess whether recovery trajectories from teacher policy improve closed-loop correction behavior, we construct a standardized recovery-state evaluation with a common set of 12 distinct error states on Fold Dishcloth. Each state deviates from nominal execution and requires corrective behavior before the task can be completed. As shown in Table 1, the BC baseline succeeds from only 4 out of 12 states, while offline distillation and OPD improve this

Table 3: **Analysis of the effectiveness of additional human demonstrations and AutoDAgger data.** "Distill" indicates the use of teacher action targets, "#Episodes" denotes the total number of training episodes, and parenthesized values indicate absolute score improvements over Human Demo under the same distillation setting. With the same number of 30 additional episodes, AutoDAgger yields substantially larger performance gains.

Data Source	Distill	Fold Dishcloth		Pick Pen		Pick Beverage	
		#Episodes	Score	#Episodes	Score	#Episodes	Score
Human Demo	✗	200	20.8	100	37.0	100	16.7
+Extra Demo	✗	230	27.8 (+ 7.0)	130	45.4 (+ 8.4)	130	44.2 (+ 27.5)
+AutoDAgger	✗	230	**48.6** (+ 27.8)	130	**63.0** (+ 26.0)	130	**50.0** (+ 33.3)
Human Demo	✓	200	27.8	100	45.4	100	26.7
+Extra Demo	✓	230	29.2 (+ 1.4)	130	54.6 (+ 9.2)	130	48.3 (+ 21.6)
+AutoDAgger	✓	230	**63.9** (+ 36.1)	130	**80.6** (+ 35.2)	130	**69.2** (+ 42.5)

Table 4: **Comparison of HG-DAgger and AutoDAgger in policy performance and recovery-data collection.** "Rec. Time" denotes the time required by the human operator or teacher policy to complete a recovery after takeover, while "Rec. SR" denotes the fraction of interventions that successfully recover and complete the task. Both metrics are evaluated over 30 extra trials. Parenthesized values indicate the recovery-time speedup of AutoDAgger over HG-DAgger.

Method	Distill	Fold Dishcloth			Pick Pen			Pick Beverage		
		Score	Rec. Time	Rec. SR(%)	Score	Rec. Time	Rec. SR(%)	Score	Rec. Time	Rec. SR(%)
HG-DAgger	✗	34.7	54.4s	76.7	75.9	29.4s	76.7	60.0	32.2s	90.0
	✓	65.3			80.6			63.3		
AutoDAgger	✗	48.6	21.5s (2.53×)	73.3	63.0	13.2s (2.23×)	66.7	50.0	12.4s (2.60×)	90.0
	✓	63.9			80.6			69.2		

result slightly to 5/12. TILDE succeeds in 8 cases, compared with 9/12 for the pretrained teacher. These results support the value of learning from the subsequent states generated by executing teacher corrections, where TILDE transfers substantially stronger recovery capability.

Scaling with AutoDAgger Data. Table 2 studies how performance scales with the amount of AutoDAgger data. Starting from a score of 27.8 with offline distillation on the human demonstration dataset, the student improves to 63.9, 65.3, and 77.8 after aggregating 30, 60, and 90 AutoDAgger episodes, respectively. The corresponding success rates exhibit a similar upward trend, indicating consistent improvement. Notably, with 90 AutoDAgger episodes, the student reaches a score of 77.8, narrowing the gap to the teacher score of 87.5. These results suggest that TILDE could benefit further from continued scaling of interactive recovery data.

Quality and Efficiency of AutoDAgger Data. Table 3 compares augmenting the original dataset with the same number of additional human teleoperation episodes or AutoDAgger episodes per task. Under BC, AutoDAgger data yield scores of 48.6, 63.0, 50.0 on Fold Dishcloth, Pick Pen, and Pick Beverage, compared with 27.8, 45.4, and 44.2 using additional human demonstrations. With distillation enabled, the AutoDAgger configuration achieves scores of 63.9, 80.6, and 69.2 across the three tasks, substantially outperforming the additional-demonstration configuration. These results suggest that AutoDAgger data provide more effective supervision than conventional expert demonstrations, which typically consist of trajectories that directly lead to task success. In contrast, AutoDAgger provides corrective supervision at student-induced states that are rarely encountered during standard demonstration collection, while also enabling on-policy distillation to more effectively leverage the teacher's knowledge.

We further compare AutoDAgger with HG-DAgger (Kelly et al., 2019), a representative human-in-the-loop data aggregation method, in terms of downstream policy performance, data collection efficiency, and intervention success. As shown in Table 4, AutoDAgger substantially reduces the time required to collect recovery trajectories across all three tasks. This improvement primarily stems from direct takeover by the teacher policy. Once an intervention is triggered, the teacher

Table 5: **Cross-architecture distillation on LIBERO benchmark with SmolVLA as the student.** TILDE outperforms the baseline and offline distillation on all four suites.

Method	Spatial	Object	Goal	10	Avg.
SmolVLA	81.4	94.4	84.2	64.6	81.2
Offline Distill	85.2	96.4	88.4	63.8	83.5
TILDE (Ours)	**88.0**	**96.6**	**94.0**	**79.6**	**89.6**

Table 6: **Rollout outcome prediction on LIBERO benchmark.** Precision, recall, and F1 score evaluate binary success/failure predictions against ground-truth task outcomes.

Metric	Spatial	Object	Goal	10
Precision	0.907	0.988	0.930	0.908
Recall	0.934	1.000	0.930	0.841
F1	0.920	0.994	0.930	0.873

can immediately generate and execute corrective actions from the current state, whereas human teleoperation incurs additional reaction and control overhead.

Despite replacing human intervention with automated teacher takeover, AutoDAgger achieves recovery success rates comparable to those of HG-DAgger. Recovery is not always successful for either method, as some failure states are inherently difficult to recover from. Importantly, HG-DAgger requires a human operator to continuously monitor execution and manually take over when intervention is needed, whereas AutoDAgger automates both processes. By eliminating human involvement during data collection while substantially increasing collection throughput, AutoDAgger provides a more scalable alternative to human-in-the-loop data aggregation.

Faster recovery offers an opportunity to close the small performance gap through additional autonomous data collection. On Fold Dishcloth, increasing the AutoDAgger dataset to 60 episodes raises the score to 65.3 (Table 2), matching HG-DAgger with 30 episodes. Combined with the 2.53$\times$ recovery-time speedup, this result suggests that AutoDAgger can use its collection efficiency to improve downstream performance within a comparable time budget.

Cross-architecture Distillation. We investigate whether TILDE can transfer capabilities across heterogeneous model architectures. Since our approach performs distillation purely through action-space alignment rather than intermediate feature alignment, the teacher and student can employ different architectures and action-generation mechanisms. We use SmolVLA (Shukor et al., 2025) without action pretraining as an efficient student policy and distill knowledge from the $\pi_{0.5}$ teacher on the LIBERO benchmark. As shown in Table 5, TILDE consistently improves the student and outperforms conventional offline distillation. On the long-horizon LIBERO-10 suite, success increases from 64.6% to 79.6%, consistent with our motivation to transfer capabilities needed in more challenging settings. These results validate the broader applicability of our framework and suggest that pretrained EFMs can serve as reusable teachers for new policy architectures.

Validation of Task Monitor. Since AutoDAgger triggers interventions based on task execution estimates from a task monitor implemented with the Robometer reward model, we further evaluate its ability to distinguish successful and failed rollouts on the LIBERO benchmark. We collect 200 SmolVLA rollout episodes on each suite and compare the model's success predictions with the ground-truth outcomes provided by the simulator. As shown in Table 6, Robometer achieves strong performance across the four suites, although recall decreases on the more challenging LIBERO-10 tasks. These results support its use as an off-the-shelf, policy-independent task monitor for AutoDAgger without task-specific finetuning.

5 Conclusion

In this work, we study how capabilities acquired through large-scale action pretraining can be transferred from pretrained EFMs to new student policies without repeating foundation-scale pretraining. TILDE combines teacher supervision at student-induced states with teacher-executed recovery trajectories, covering both corrective actions and the subsequent recovery-state distribution. AutoDAgger enables this process without human monitoring or takeover by using a task monitor to trigger teacher intervention. Across three real-world manipulation tasks, TILDE consistently improves over behavior cloning, offline distillation, and conventional on-policy distillation, while AutoDAgger collects recovery data more than 2$\times$ faster than HG-DAgger. Cross-architecture experiments with SmolVLA further validate capability transfer to heterogeneous student policies.

AI USE STATEMENT

In this work, we used generative AI tools for design or provide feedback on research methodology or experiments, and implement methods. We have not used generative AI tools for generate synthetic datasets, clean and reformat dataset, support qualitative and thematic data analysis, and interpret results; and help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation are not applicable to this work. Additionally, we used generative AI tools for suggest experimental parameters, create or edit software code, summarize or analyse existing literature, edit a research paper to improve readability, and identify relevant literature. We have reviewed all AI-assisted work. All AI-assisted suggestions related to methodology and experiments were critically evaluated by the authors and were adopted only after independent assessment and empirical validation. AI-generated code was manually reviewed, integrated, and tested by the authors to verify its correctness and consistency with the intended methodology. Literature reviews and suggested references were cross-checked against the original publications. LLM-generated contents in manuscript were limited to improving clarity, readability, and presentation; all technical statements, experimental results, interpretations, and conclusions were reviewed and finalized by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our research adheres to high ethical standards in robot learning and embodied AI, ensuring transparency, reproducibility, and fairness. This work does not involve human subjects, personally identifiable information, or sensitive data. While increasing autonomy in robot data collection and policy improvement may introduce safety risks if deployed in unconstrained environments, our experiments are conducted under controlled settings with appropriate safety supervision. We recommend that such methods be equipped with suitable safety mechanisms and carefully validated before deployment in real-world environments.

REPRODUCIBILITY STATEMENT

The paper reports all information including hardware platforms, real-world tasks, simulation benchmarks, evaluation metrics, implementation details, and hyperparameter settings in Section 4.1 and Appendix A. We plan to release code and deployment instructions upon publication.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. Gkd: Generalized knowledge distillation for auto-regressive sequence models. *arXiv preprint arXiv:2306.13649*, 2023.

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 *ICLR*, 2024.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine intelligence*, pp. 103–129, 1995.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: an open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *IROS*, 2025.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *SIGKDD*, 2006.

Remi Cadene, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, Mustafa Shukor, Jess Moss, Alexander Soare, Dana Aubakirova, Quentin Lhoest, Quentin Gallouédec, and Thomas Wolf. Lerobot: An open-source library for end-to-end robot learning. In *ICLR*, 2026.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Constantin Cronrath, Emilio Jorge, John Moberg, Mats Jirstrand, and Bengt Lennartson. Bagger: A bayesian algorithm for safe and query-efficient imitation learning. In *Machine Learning in Robot Motion Planning–IROS 2018 Workshop*, 2018.

Shaoqi Dong, Chaoyou Fu, Haihan Gao, Yi-Fan Zhang, Chi Yan, Chu Wu, Xiaoyu Liu, Yunhang Shen, Jing Huo, Deqiang Jiang, et al. Vita-vla: Efficiently teaching vision-language models to act via action expert distillation. *arXiv preprint arXiv:2510.09607*, 2025.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile ALOHA: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In *CoRL*, 2024.

Weifan Guan, Qinghao Hu, Aosheng Li, and Jian Cheng. Efficient vision-language-action models for embodied manipulation: A systematic survey. *arXiv preprint arXiv:2510.17111*, 2025.

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

Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S Brown, and Ken Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *arXiv preprint arXiv:2109.08273*, 2021a.

Ryan Hoque, Ashwin Balakrishna, Carl Putterman, Michael Luo, Daniel S Brown, Daniel Seita, Brijen Thananjeyan, Ellen Novoseller, and Ken Goldberg. Lazydagger: Reducing context switching in interactive imitation learning. In *2021 IEEE 17th international conference on automation science and engineering (case)*, pp. 502–509. IEEE, 2021b.

Zheyuan Hu, Robyn Wu, Naveen Enock, Jasmine Li, Riya Kadakia, Zackory Erickson, and Aviral Kumar. Rac: Robot learning for long-horizon tasks by scaling recovery and correction. *IEEE Transactions on Robotics*, 2026.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra

Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Boseong Jeon, Yunho Choi, and Taehan Kim. Shallow-π: Knowledge distillation for flow-based vlas. *arXiv preprint arXiv:2601.20262*, 2026.

Tobias Jülg, Wolfram Burgard, and Florian Walter. Refined policy distillation: From vla generalists to rl experts. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11677–11684. IEEE, 2025.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083. IEEE, 2019.

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

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *CoRL*, 2024.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. In *RSS*, 2026.

Anthony Liang, Yigit Korkmaz, Jiahui Zhang, Minyoung Hwang, Abrar Anwar, Sidhant Kaushik, Aditya Shah, Alex S Huang, Luke Zettlemoyer, Dieter Fox, et al. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons. *arXiv preprint arXiv:2603.02115*, 2026.

Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, et al. Evo-1: Lightweight vision-language-action model with preserved semantic alignment. *arXiv preprint arXiv:2511.04555*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, pp. 44776–44791, 2023.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*, 2024.

Kunal Menda, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Ensembledagger: A bayesian approach to safe imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5041–5048. IEEE, 2019.

Michael Murray, Daphne Chen, Simran Bagaria, Dean Fortier, Tess Hellebrekers, Galen Mullins, Harshavardhan Gajarla, Oier Mees, Maya Cakmak, and Andrey Kolobov. Flowdagger: Human-in-the-loop adaptation of generative robot policies in latent space. *arXiv preprint arXiv:2607.08877*, 2026.

Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *ICRA*, 2024.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *NeurIPS*, 1988.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011.

Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

Ranjan Sapkota, Yang Cao, Konstantinos I Roumeliotis, and Manoj Karkee. Vision-language-action (vla) models: Concepts, progress, applications and challenges. *arXiv preprint arXiv:2505.04769*, 2025.

Maanping Shao, Feihong Zhang, Gu Zhang, Baiye Cheng, Zhengrong Xue, and Huazhe Xu. X-distill: Cross-architecture vision distillation for visuomotor learning. *IEEE Robotics and Automation Letters*, 2026.

Rui Shao, Wei Li, Lingsen Zhang, Renshan Zhang, Zhiyang Liu, Ran Chen, and Liqiang Nie. Large vlm-based vision-language-action models for robotic manipulation: A survey. *arXiv preprint arXiv:2508.13073*, 2025.

Jin Shi, Brady Zhang, and Yishun Lu. Offline semantic guidance for efficient vision-language-action policy distillation. *arXiv preprint arXiv:2605.16241*, 2026.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. In *RSS*, 2024.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. BridgeData v2: A dataset for robot learning at scale. In *CoRL*, 2023.

Zhenglin Wan, Jingxuan Wu, Xingrui Yu, Chubin Zhang, Mingcong Lei, Bo An, Ivor W Tsang, and Yang You. Adversarial dual on-policy distillation from expressive teacher. *arXiv preprint arXiv:2605.27095*, 2026.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. In *AAAI*, 2026a.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, et al. Openwam: An open, modular exploration towards systematic world-action model pretraining. *arXiv preprint arXiv:2609.07398*, 2026b.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *CVPR*, 2024.

Wenli Xiao, Haotian Lin, Andy Peng, Haoru Xue, Tairan He, Zhengyi Luo, Yuqi Xie, Fengyuan Hu, Jim Fan, Guanya Shi, et al. Self-improving vision-language-action models with data generation via residual rl. In *International Conference on Learning Representations*, volume 2026, pp. 13209–13236, 2026.

Charles Xu, Qiyang Li, Jianlan Luo, and Sergey Levine. Rldg: Robotic generalist policy distillation via reinforcement learning. *arXiv preprint arXiv:2412.09858*, 2024a.

Xiaomeng Xu, Yifan Hou, Zeyi Liu, and Shuran Song. Compliant residual dagger: Improving real-world contact-rich manipulation with human corrections. In *NeurIPS*, 2025.

Zhiyuan Xu, Kun Wu, Junjie Wen, Jinming Li, Ning Liu, Zhengping Che, and Jian Tang. A survey on robotics with foundation models: toward embodied ai. *arXiv preprint arXiv:2402.02385*, 2024b.

Rui Yan, Zaitian Gongye, Lars Paulsen, Xuxin Cheng, and Xiaolong Wang. Human-robot copilot for data-efficient imitation learning. *arXiv preprint arXiv:2604.03613*, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Wencheng Ye, Tianshi Wang, Lei Zhu, Fengling Li, Guoli Yang, and Hengtao Shen. Actdistill: General action-guided self-derived distillation for efficient vision-language-action models. *arXiv preprint arXiv:2511.18082*, 2025.

Zhaoshu Yu, Bo Wang, Pengpeng Zeng, Haonan Zhang, Ji Zhang, Lianli Gao, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. A survey on efficient vision-language-action models. *arXiv preprint arXiv:2510.24795*, 2025.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Dapeng Zhang, Jing Sun, Chenghui Hu, Xiaoyan Wu, Zhenlong Yuan, Rui Zhou, Fei Shen, and Qingguo Zhou. Pure vision language action (vla) models: A comprehensive survey. *arXiv preprint arXiv:2509.19012*, 2025a.

Jiazhi Zhang, Yuhu Cheng, CL Philip Chen, Hengrui Zhang, and Xuesong Wang. Diffusion policy distillation for offline reinforcement learning. *Neural Networks*, 190:107694, 2025b.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *RSS*, 2023.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *ICLR*, 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.

René Zurbrügg, Tifanny Portela, Arjun Bhardwaj, Aravind Elanjimattathil Vijayan, Maximum Wilder-Smith, and Marco Hutter. Vr-dagger: Immersive vr for dexterous data collection and uncertainty-guided on-policy correction. *arXiv preprint arXiv:2605.27114*, 2026.

APPENDIX

We provide the following contents in the appendix:

- Section A provides implementation details of AutoDAgger, real-world experiments, simulation benchmarks, training configurations, and robot deployment.
- Section B presents additional experimental results, including failure-detection evaluation, statistics, and visualizations of the AutoDAgger data-collection process.
- Section C discusses limitations and potential directions for future work.

A ADDITIONAL IMPLEMENTATION DETAILS

A.1 AUTODAGGER

Task Monitor and Intervention Trigger. We implement task monitoring asynchronously alongside policy inference. Each front-camera observation received by the monitor is appended to the visual history $\boldsymbol{o}_{1:t} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t]$. To preserve temporal coverage while bounding inference cost, we uniformly sample at most N_{frame} observations:

$$\widetilde{\boldsymbol{o}}_{1:K_t} = \text{UniformSample}\left(\boldsymbol{o}_{1:t}, K_t\right), \quad K_t = \min(t, N_{\text{frame}}), \tag{11}$$

where $\widetilde{\boldsymbol{o}}_{1:K_t}$ denotes the sampled visual history provided to Robometer. Every τ_{mon} seconds, the sampled history and task instruction are submitted to Robometer without blocking policy inference. If a new request arrives while a previous one is still pending, the pending request is replaced by the latest one to avoid evaluating stale robot states. Each prediction is logged together with its timestamp and inference latency for runtime visualization and offline analysis.

The AutoDAgger controller continuously polls the monitor associated with the active policy. At the beginning of each episode, it resets the observation history, prediction records, progress reference, temporal state, and pending monitoring requests. We initialize the progress reference $\bar{p}_0 = 0$ and the valid-progress timestamp τ_{valid} using the episode start time. The student monitor is then activated, while the teacher monitor remains idle. Once student failure is detected, the controller immediately stops the robot, pauses student inference, clears the action buffer and observation window to invalidate any pending student actions, switches control to the teacher, and initializes a fresh monitoring state for recovery. Robot execution resumes only after a teacher action chunk has been obtained, ensuring that no residual student actions are executed after takeover.

During teacher execution, the controller continues polling the teacher monitor until either recovery is confirmed or teacher failure is detected. If recovery fails, the trajectory is discarded. Otherwise, the student rollout and teacher recovery trajectory are combined and stored for subsequent policy training. Episodes completed entirely by the student are not retained, since AutoDAgger focuses on recovery trajectories.

Hyperparameter Setup. We use $N_{\text{frame}} = 8$, $\tau_{\text{mon}} = 1$, $\epsilon_p = 0.05$, and $\theta_s = 0.5$ across all experiments. N_{frame} balances temporal context against monitoring cost, while τ_{mon} determines the monitor update frequency. The progress margin ϵ_p prevents small prediction jitter from being interpreted as meaningful task advancement, and θ_s converts the predicted success probability into a binary success signal. These parameters are kept fixed across all tasks and policies.

In addition, AutoDAgger uses task- and policy-specific temporal thresholds to determine execution failure and successful completion. The failure timeout T_{fail} specifies how long a policy may execute without sufficient evidence of progress or success before being classified as failed. The success-confirmation duration T_{succ} requires the predicted success probability to remain above θ_s continuously before execution is considered successful. Formally, success is confirmed at step t only if

$$q_u > \theta_s, \quad \forall u \text{ such that } \tau_u \in [\tau_t - T_{\text{succ}},\, \tau_t]. \tag{12}$$

This temporal consistency criterion prevents transient prediction spikes from prematurely terminating an episode. The thresholds used during autonomous data collection are summarized in Table 7, with all values reported in seconds.

Table 7: Task- and policy-specific temporal thresholds used during AutoDAgger data collection. T_{fail} determines the failure-detection timeout, while T_{succ} determines the duration of continuous success predictions required to confirm task success.

Task	T_{fail} (s)		T_{succ} (s)	
	Student	Teacher	Student	Teacher
Fold Dishcloth	6	14	5	5
Pick Pen	3	8	4	4
Pick Beverage	2	6	4	4



Figure 5: AgileX PiperX robotic arms.

We use different temporal thresholds for the student and teacher because their execution characteristics differ. A shorter T_{fail} is used for the student to detect failures early and trigger timely intervention, reducing the chance of further deviation from recoverable states. The teacher is assigned a longer T_{fail} because it generally has stronger task competence and must recover from student-induced off-nominal states before completing the task. We also vary the thresholds across tasks to account for differences in execution dynamics and expected duration. For example, folding a dishcloth involves contact-rich manipulation and may contain temporary pauses during valid execution, whereas picking tasks exhibit more temporally compact and visually apparent progress.

A.2 Real-world Experiments

Hardware Setup. We conduct our real-robot experiments on an AgileX PiperX dual-arm robot platform, as illustrated in Figure 5, following the leader-follower teleoperation design of ALOHA-style systems (Zhao et al., 2023; Fu et al., 2024). The platform consists of four 7-DoF (6 joints + 1 gripper) PiperX arms: two leader arms operated by a human demonstrator and two corresponding follower arms that reproduce the leader commands during data collection and execute policy-generated actions during autonomous inference. The visual sensing system comprises one front-facing Intel RealSense D455 camera and two Intel RealSense D435 wrist cameras, with one wrist camera mounted on each follower arm.

Tasks. We evaluate our method on three real-robot manipulation tasks: "Fold Dishcloth", "Pick Pen", and "Pick Beverage". These tasks cover deformable-object manipulation, precision grasping, and object relocation. Illustrations of the three tasks are provided in Figure 6.

The language instructions for the tasks are:

- **Fold Dishcloth:** "Fold the dishcloth in half twice, then place it in the position slightly to the front and left."
- **Pick Pen:** "Pick up the scattered pens one by one and place them in the yellow cup."
- **Pick Beverage:** "Pick up the beverage and put it in the paper box."

Evaluation Protocol. Each method is evaluated using a fixed number of real-robot rollouts per task: 12 trials for Fold Dishcloth, 27 trials for Pick Pen, and 30 trials for Pick Beverage. We use predefined test configurations, including fixed object categories and controlled variations in object

position and orientation, and keep these configurations as consistent as practically possible across all evaluated methods. A rollout is considered successful only if the complete task is accomplished within the maximum episode length without human intervention.

In addition to binary task success, we report a fine-grained success score to capture both partial task completion and execution quality under a limited real-robot evaluation budget. Each task is decomposed into semantically meaningful stages. Each stage receives a score of 0, 0.5, or 1, corresponding to failure, partial completion or successful completion after repeated attempts, and successful completion in a single attempt, including immediate recovery from minor execution issues, respectively. The task-level score is computed by averaging over all stages and evaluation rollouts, and then multiplied by 100 for better display. The scoring criteria are as follows:

1. Fold Dishcloth
    - Stage 1: Folding the dishcloth in half from left to right
        - 0: The robot fails to complete the fold after multiple attempts.
        - 0.5: The robot completes the fold after several attempts or demonstrates partial recovery from an execution error during the process.
        - 1: The robot completes the fold in a single attempt, including cases in which it immediately recovers from minor issues such as a flipped corner.
    - Stage 2: Folding the dishcloth in half a second time, from bottom to top
        - 0: The robot fails to complete the second fold after multiple attempts.
        - 0.5: The robot completes the second fold after several attempts or demonstrates partial recovery from an execution error during the process.
        - 1: The robot completes the second fold in a single attempt, including cases in which it immediately recovers from minor issues such as a flipped corner.
    - Stage 3: Picking up the folded dishcloth and placing it in the front-left target region
        - 0: The robot fails to pick up and place the folded dishcloth after multiple attempts.
        - 0.5: The robot completes the pick-and-place operation after several attempts or demonstrates partial recovery from an execution error during the process.
        - 1: The robot completes the pick-and-place operation in a single attempt.
2. Pick Pen
    - Stage 1: Grasping the pen
        - 0: The robot fails to grasp the pen.
        - 0.5: The robot grasps the pen after repeated attempts or recovers from an unsuccessful grasp.
        - 1: The robot grasps the pen correctly on the first attempt.
    - Stage 2: Placing the pen into the cup
        - 0: The pen is not placed into the cup or is dropped outside it.
        - 0.5: The pen is placed into the cup after a recovery attempt or with substantial contact against the cup.
        - 1: The robot directly places the pen into the cup without requiring recovery.
    - For trials containing multiple pens, the two-stage scoring procedure is applied independently to each pen.
3. Pick Beverage
    - Stage 1: Grasping the beverage
        - 0: The robot fails to grasp the beverage.
        - 0.5: The robot grasps it after repeated attempts or recovers from an unsuccessful grasp.
        - 1: The robot grasps it correctly on the first attempt.
    - Stage 2: Placing the beverage into the box
        - 0: The beverage is not placed into the box or is dropped outside it.
        - 0.5: The beverage is placed into the box after a recovery attempt or with an undesirable collision.
        - 1: The robot directly places the beverage into the box without requiring recovery.

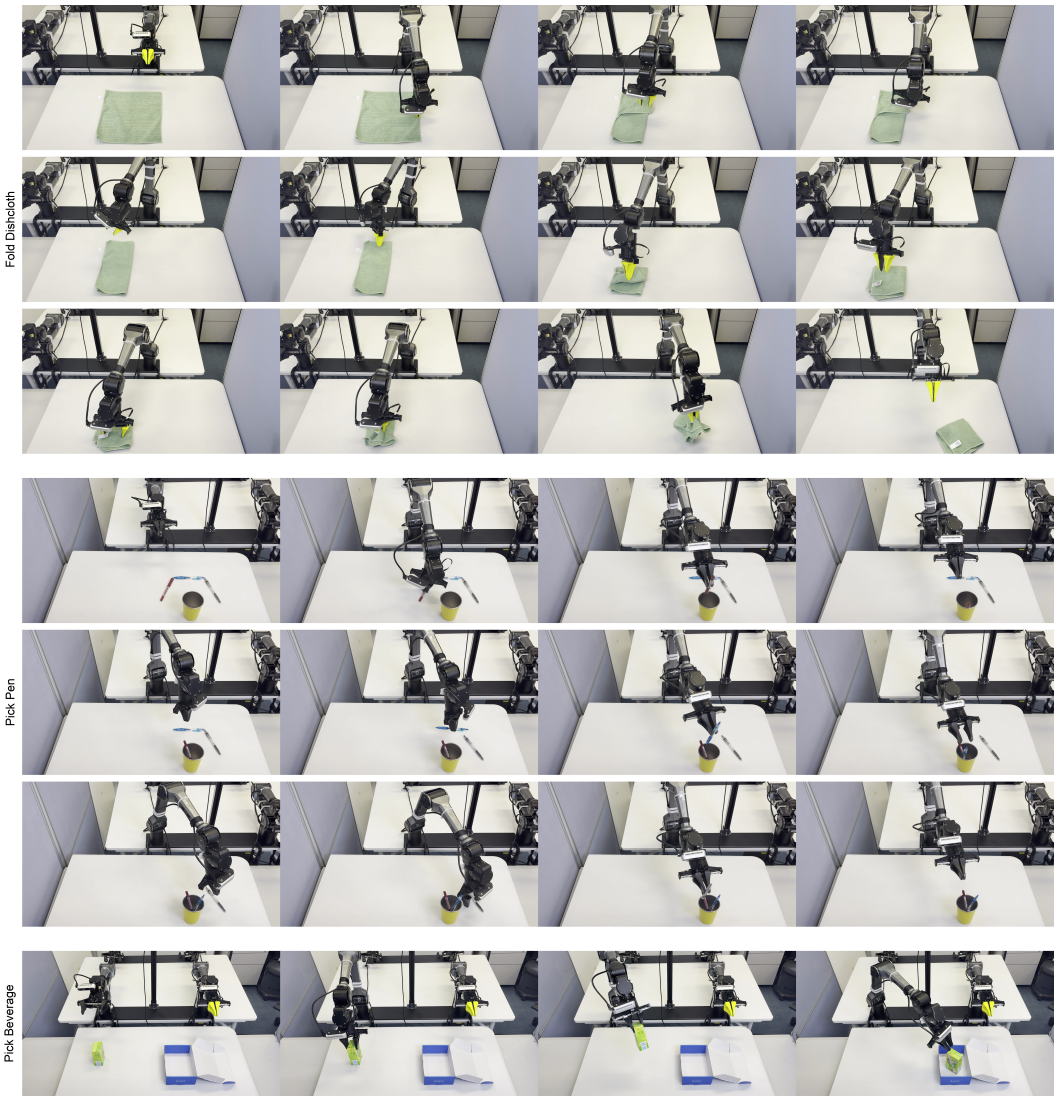

Figure 6: Visualization of three real-world tasks.

A.3 Simulation Benchmarks

LIBERO. The LIBERO benchmark (Liu et al., 2023) has four task suites—Spatial, Object, Goal, and 10 (Long)—each emphasizing different manipulation capabilities. We train a single policy jointly across all four suites. For $\pi_{0.5}$, VLA-Adapter-Pro, and SmolVLA, we use the dataset released by OpenVLA[1] (Kim et al., 2024) and convert it to the LeRobot format (Cadene et al., 2026). For X-VLA, we instead use the HDF5-format dataset released by the X-VLA authors[2], which represents actions in the EE6D action space. Each suite consists of 10 tasks, and each task is evaluated over 50 rollouts.

LIBERO-Plus. LIBERO-Plus (Fei et al., 2025) extends LIBERO with seven dimensions of visual and configuration perturbations. In our pilot study (Figure 1), we evaluate the "Background" setting, which perturbs scene and surface appearance, and the "Robot" setting, which varies the manipulator's initial pose. These settings contain 1,076 and 1,550 task instances, respectively. The evaluations are performed using checkpoints trained exclusively on the original LIBERO data, thereby measuring out-of-distribution generalization under environmental and embodiment perturbations.

A.4 Training Setup

We train $\pi_{0.5}$[3], X-VLA[4], VLA-Adapter[5], and SmolVLA[6] based on their official codebases, with modifications to support our training pipeline and data formats. For the action-pretraining-free variants of $\pi_{0.5}$ and X-VLA, we initialize the model from VLM backbones: PaliGemma (Beyer et al., 2024) for $\pi_{0.5}$, and Florence-2-large (Xiao et al., 2024) for X-VLA, while training the action experts from scratch.

We follow the default or recommended training configurations for each model, with the key hyperparameters summarized in Table 8 and Table 9. For real-robot experiments, demonstrations are recorded in absolute joint space at 30FPS and converted to relative joint actions by subtracting the corresponding proprioceptive states if needed.

For AutoDAgger data collection, the student collection policy is first trained on $\mathcal{D}_{\text{demo}}$ using behavior cloning only. We then perform AutoDAgger collection using this fixed BC baseline. After collection, we reinitialize the student and train on the aggregated dataset.

During training, samples are drawn randomly from the pooled training datasets without additional dataset balancing, and the training loss for each sample is determined by its data source. The supervision applied to each training variant is summarized in Table 10. We set the distillation loss weight $\lambda_{\text{KD}} = 1$ for all experiments.

All models are trained on 8 NVIDIA A800 GPUs with 80GB VRAM. Training takes approximately 1–2 days for $\pi_{0.5}$ and less than one day for the other models.

A.5 Robot Deployment Setup

We deploy the policy and task-monitor models on a workstation equipped with NVIDIA RTX 4090 GPU. The inference latency is 80.0 ± 1.6ms for $\pi_{0.5}$ and 266.9 ± 0.5ms for the task monitor Robometer. The robot-side client runs on ROS at a fixed control frequency of 30Hz. We employ asynchronous inference together with temporal ensembling to enable continuous control and smooth the action trajectories.

A.6 Cross-architecture Distillation Setup

For the cross-architecture experiment in Table 5, We follow the same collection and training protocol described above. We use $\pi_{0.5}$ as the teacher and SmolVLA as the student, which is first trained by

[1] HuggingFace: `openvla/modified_libero_rlds`
[2] HuggingFace: `2toINF/Libero-XVLA-format`
[3] GitHub: `Physical-Intelligence/openpi`
[4] GitHub: `2toinf/X-VLA`
[5] GitHub: `OpenHelix-Team/VLA-Adapter`
[6] GitHub: `huggingface/lerobot`

Table 8: Hyperparameter settings for training $\pi_{0.5}$ and X-VLA policies.

Model	$\pi_{0.5}$		X-VLA	
Task	AgileX	LIBERO	AgileX	LIBERO
Prediction Horizon H	50	10	30	30
Action Space	Relative Joint	Delta EEF	Abs Joint	Abs EE6D
Global Batch Size	128	256	128	128
Training Steps	50k	30k	50k	60k
Optimizer	AdamW	AdamW	AdamW	AdamW
Weight Decay	0	0	0	0
Betas	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.95)
Base LR	2.5e-5	5e-5	1e-4	1e-4
LR Scheduler	Cosine Decay	Cosine Decay	Constant	Constant
Warmup Steps	1k	10k	2k	2k
Grad Norm Clip	1.0	1.0	1.0	1.0
EMA Decay	0.99	0.999	N/A	N/A

Table 9: Hyperparameter settings for training VLA-Adapter-Pro and SmolVLA policies.

Model	VLA-Adapter-Pro		SmolVLA
Task	AgileX	LIBERO	LIBERO
Prediction Horizon H	30	8	10
Action Space	Abs Joint	Delta EEF	Delta EEF
Global Batch Size	128	64	64
Training Steps	50k	100k	100k
Optimizer	AdamW	AdamW	AdamW
Weight Decay	0.01	0.01	1e-10
Betas	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.95)
Base LR	2e-4	2e-4	1e-4
LR Scheduler	Constant	Constant	Cosine Decay
Warmup Steps	0	0	1k
Grad Norm Clip	N/A	N/A	10.0
EMA Decay	N/A	N/A	N/A

behavior cloning on the LIBERO dataset. We then use this BC student for AutoDAgger collection and train TILDE using the resulting student-rollout and teacher-recovery data together with the demonstrations.

B Additional Experimental Results

B.1 Validation of Task Monitor on Real-world Rollouts

We further evaluate the task monitor and intervention trigger using 20 real-robot rollouts, consisting of 10 student rollouts and 10 teacher rollouts. Ground-truth outcomes are manually determined according to whether the corresponding policy successfully completes the task. Among the student rollouts, two are successful and eight are failed; among the teacher rollouts, eight are successful and two are failed. The complete evaluation set is therefore balanced, containing 10 successful and 10 failed trajectories.

In Table 11, *Positive* denotes that the monitoring system predicts task success, while *Negative* denotes that it detects task failure. *True* and *False* refer to ground-truth task success and failure, respectively. The monitoring system correctly identifies 8 of the 10 successful trajectories and 9 of the 10 failed trajectories, yielding an overall accuracy of 85%. The two FNs indicate that the monitor

Table 10: Comparison of different training configurations.

Method	$\mathcal{D}_{\text{demo}}$	$\mathcal{D}_{\text{roll}}$	$\mathcal{D}_{\text{rec}}$
BC Baseline	BC	–	–
Offline Distill	BC+KD	–	–
OPD	BC+KD	KD	–
Recovery BC	BC	–	BC
Recovery Distill	BC+KD	–	BC+KD
TILDE	BC+KD	KD	BC+KD

Table 11: Rollout outcome prediction results on real-world tasks.

Prediction	Ground-truth outcome		Total
	True	False	
Positive	8 (TP)	1 (FP)	9
Negative	2 (FN)	9 (TN)	11
Total	10	10	20

adopts a conservative criterion for confirming task success. This conservative behavior is useful for teacher recovery verification, where a trajectory may complete the task only after multiple corrective attempts. Rejecting such trajectories reduces success recall but helps prevent lower-quality recovery data from entering the training set. Overall, the detector achieves a precision of 88.9%, a recall of 80.0%, and an F1 score of 84.2%. Although the evaluation set is limited by the cost of real-robot experiments, these results indicate that the task monitor provides a practical signal for triggering teacher takeover and verifying recovery outcomes.

B.2 Experimental Result Statistics

We supplement the performance statistics of experiments in the main text in Table 12, Table 13, Table 14, and Table 15.

B.3 Visualization of AutoDAgger Collection

We visualize representative AutoDAgger collection trajectories for Fold Dishcloth, Pick Pen, and Pick Beverage tasks in Figure 7. Before intervention, the predicted task progress initially increases during student execution but subsequently stagnates, while the success probability remains well below $\theta_s = 0.5$. The task monitor consequently detects the combination of sustained progress stagnation and low success probability, triggering teacher takeover at the red-bordered frames. Following takeover, the predicted progress may decrease temporarily as the teacher corrects the student-induced state. It subsequently resumes increasing as recovery proceeds, reaching 0.869, 0.814, and 0.847 for the three tasks. Meanwhile, the success probability remains low during most of the recovery process and rises above θ_s only near task completion, reaching 0.844, 0.582, and 0.926, respectively.

C Limitations and Future Work

Despite the improved student performance demonstrated by TILDE, several limitations remain. First, TILDE is evaluated with a single round of AutoDAgger collection followed by student training, rather than iterative data aggregation. Extending the experiments to alternate between policy updates and subsequent rounds of AutoDAgger could progressively expose the student to increasingly relevant failure and recovery states. Second, AutoDAgger relies on the accuracy of the task monitor and the recoverability of the teacher policy; incorrect intervention decisions or failure states beyond the teacher's capability may reduce the quality or efficiency of collected data. More robust task monitoring and adaptive intervention strategies are therefore promising directions. Finally, our

Table 12: Performance statistics for the real-world tasks, corresponding to Figure 3 and Figure 4. The 96% confidence intervals are computed using bootstrap resampling.

Method	Fold Dishcloth		Pick Pen		Pick Beverage	
	Mean	96% CI	Mean	96% CI	Mean	96% CI
Teacher	87.5	(69.4, 98.6)	87.0	(78.7, 94.4)	79.2	(65.0, 91.7)
BC Baseline	20.8	(2.8, 41.7)	37.0	(22.2, 51.9)	16.7	(8.3, 25.8)
Offline Distill	27.8	(6.9, 51.4)	45.4	(31.5, 60.2)	26.7	(14.2, 40.0)
OPD	43.1	(16.7, 69.4)	58.3	(45.4, 70.4)	40.0	(25.0, 55.0)
Recovery BC	48.6	(22.2, 75.0)	63.0	(48.1, 76.9)	50.0	(32.5, 67.5)
Recovery Distill	54.2	(27.8, 80.6)	72.2	(56.5, 86.1)	68.3	(54.2, 80.8)
TILDE	63.9	(38.9, 87.5)	80.6	(72.2, 88.0)	69.2	(53.3, 83.3)

Table 13: Performance statistics for the real-world tasks corresponding to Table 3.

Data Source	Distill	Fold Dishcloth		Pick Pen		Pick Beverage	
		Mean	96% CI	Mean	96% CI	Mean	96% CI
Human Demo	✗	20.8	(2.8, 41.7)	37.0	(22.2, 51.9)	16.7	(8.3, 25.8)
+Extra Demo	✗	27.8	(6.9, 51.4)	45.4	(27.8, 63.0)	44.2	(28.3, 60.0)
+AutoDAgger	✗	48.6	(22.2, 75.0)	63.0	(48.1, 76.9)	50.0	(32.5, 67.5)
Human Demo	✓	27.8	(6.9, 51.4)	45.4	(31.5, 60.2)	26.7	(14.2, 40.0)
+Extra Demo	✓	29.2	(9.7, 51.4)	54.6	(40.7, 68.5)	48.3	(33.3, 63.3)
+AutoDAgger	✓	63.9	(38.9, 87.5)	80.6	(72.2, 88.0)	69.2	(53.3, 83.3)

Table 14: Performance statistics for the real-world tasks corresponding to Table 4.

Method	Distill	Fold Dishcloth		Pick Pen		Pick Beverage	
		Mean	96% CI	Mean	96% CI	Mean	96% CI
HG-DAgger	✗	34.7	(11.1, 59.7)	75.9	(60.2, 89.8)	60.0	(46.7, 73.3)
	✓	65.3	(33.3, 90.3)	80.6	(68.5, 90.7)	63.3	(48.3, 77.5)
AutoDAgger	✗	48.6	(22.2, 75.0)	63.0	(48.1, 76.9)	50.0	(32.5, 67.5)
	✓	63.9	(38.9, 87.5)	80.6	(72.2, 88.0)	69.2	(53.3, 83.3)

Table 15: Recovery data collection statistics for the real-world tasks corresponding to Table 4. Rec. Time is reported as mean $\pm$ standard deviation from successful recovery.

Task	HG-DAgger			AutoDAgger		
	Rec. Time	Rec. SR (%)	96% CI	Rec. Time	Rec. SR (%)	96% CI
Fold Dishcloth	54.4 ± 10.9s	76.7	(60.0, 90.0)	21.5 ± 5.8s	73.3	(56.7, 90.0)
Pick Pen	29.4 ± 2.4s	76.7	(60.0, 90.0)	13.2 ± 1.4s	66.7	(50.0, 83.3)
Pick Beverage	32.2 ± 2.8s	90.0	(76.7, 100.0)	12.4 ± 1.8s	90.0	(76.7, 100.0)

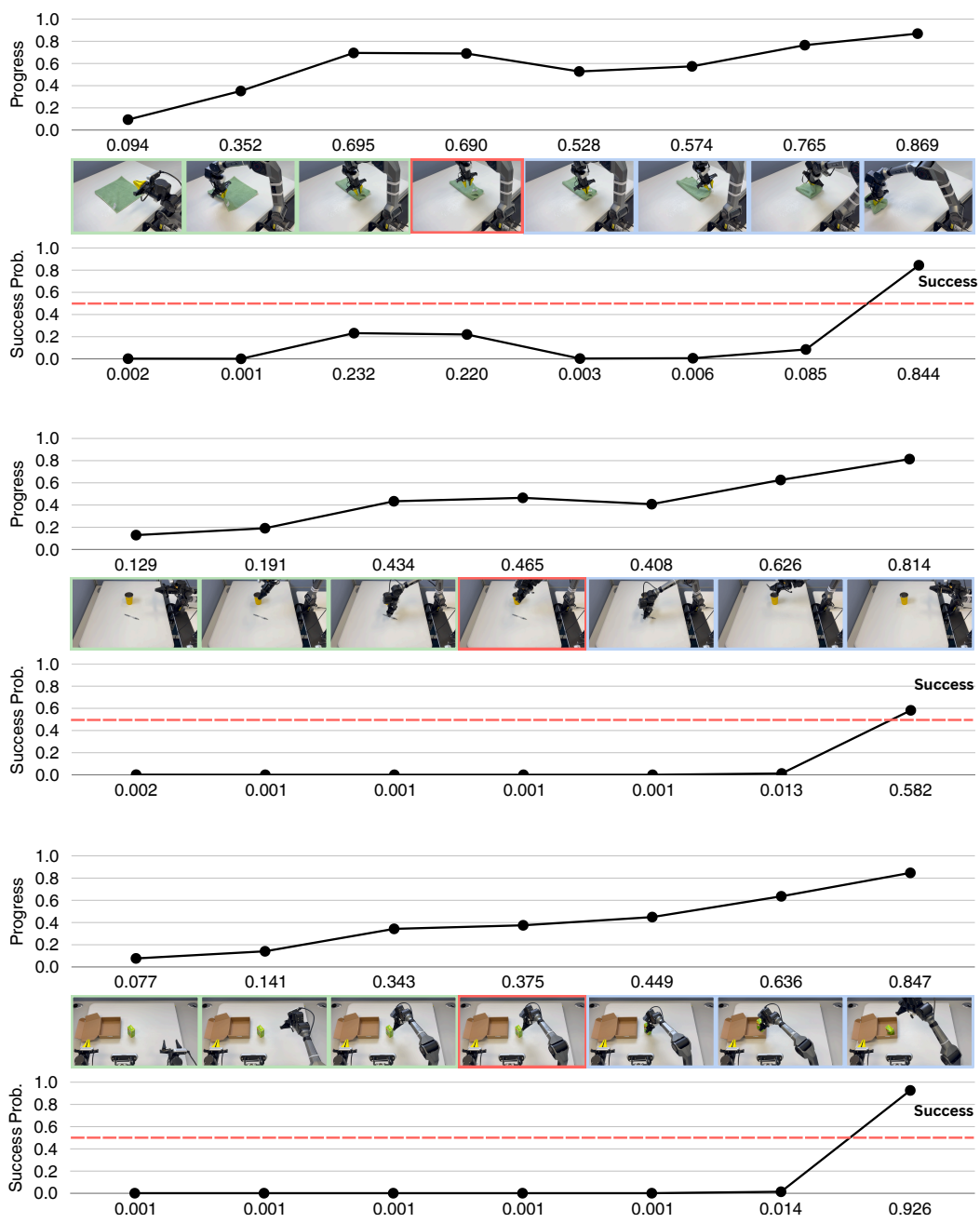

Figure 7: **Representative AutoDAgger collection trajectories.** Examples for Fold Dishcloth, Pick Pen, and Pick Beverage are shown from top to bottom. Green, red, and blue frame borders denote student execution, detected failure, and teacher recovery, respectively. The red dashed line marks the success threshold $\theta_s = 0.5$.

experimental evaluation focuses on a limited set of real-world tasks, simulation benchmarks, and teacher–student architectures. Future work will scale TILDE to broader tasks, embodiments, and heterogeneous policy families, and investigate how closely student policies can approach EFM with continued autonomous data aggregation and distillation.