Twin-DAgger: Synergizing Digital Twins and Human Corrections for Efficient Robot Manipulation

Jiahang Li[1,2#], Zhirui Zhang[1,2#], Kairan Ding[1,2], Fan Fei[1,2], Yunkai Tang[1,2], Jiaming Liu[1], Xiao He[2], Ziyu Chen[2], Gaohao Zhou[2], Jieji Ren[3], Yandong Guo[2], Shanghang Zhang[1], and Boxin Shi[1*]
[1]Peking University [2]AI[2] Robotics [3]Shanghai Jiao Tong University

jiahangli25@stu.pku.edu.cn shiboxin@pku.edu.cn

Abstract. Imitation learning enables robots to acquire complex manipulation skills from expert demonstrations, but offline-trained policies struggle to generalize beyond their training distribution, where small errors compound into severe distribution shift. Dataset Aggregation (DAgger) mitigates this by collecting human corrections during deployment, yet conventional DAgger operates entirely in the real world with a one-to-one ratio between expert effort and training trajectories makes comprehensive out-of-distribution (OOD) coverage impractical. Digital Twin (DT) techniques can scale data generation, but when applied to DAgger data without modification, they ignore the spatial and temporal failure patterns that DAgger rollouts reveal, leaving policies vulnerable precisely where and when they fail most. We propose Twin-DAgger, which treats each real-world intervention as simultaneously a spatial probe identifying failure-prone workspace regions and a temporal prior marking critical deviation moments. Twin-DAgger exploits these signals through two targeted augmentation mechanisms: difficulty-based stratified sampling concentrates synthetic data generation on high-failure-rate regions, and intervention-focused trajectory warping perturbs trajectories at each intervention onset to simulate a distribution of near-failure states while preserving the expert's corrective recovery. On real-world manipulation tasks, Twin-DAgger matches conventional DAgger performance using only 15% human interventions and outperforms failure-agnostic DT baselines by 20% success rate under equal data budgets, demonstrating that generating data where and when the policy fails, rather than generating more data uniformly, is the critical driver of sample-efficient robustness.

Keywords: imitation learning · digital twins · dataset aggregation

# Equal contribution. * Corresponding author. [1] Jiahang Li, Zhirui Zhang, Kairan Ding, Fan Fei, Yunkai Tang, Jiaming Liu, Shanghang Zhang, and Boxin Shi are with State Key Laboratory of Multimedia Information Processing, National Engineering Research Center of Visual Technology, and PKU-AI[2] Robotics Joint Lab of Embodied AI, School of Computer Science, Peking University. [3] Jieji Ren is with State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University.

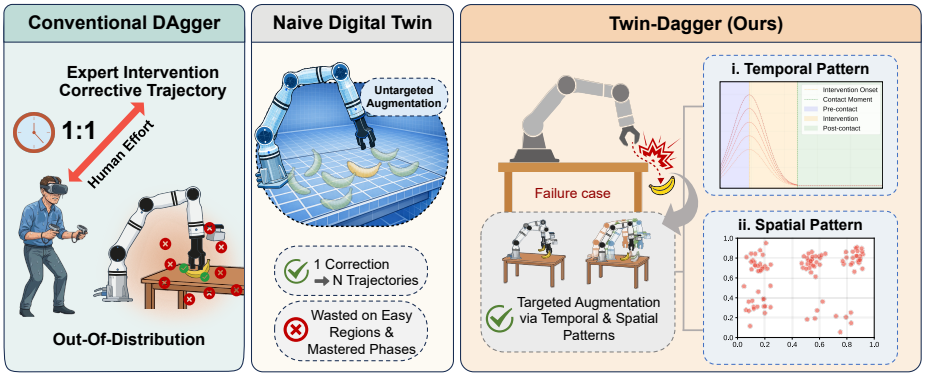

Fig. 1: Ideology of the proposed Twin-DAgger. Left: Conventional DAgger relies entirely on costly real-world interventions, resulting in a highly inefficient one-to-one ratio between expert effort and data production. Middle: Directly applying existing DT augmentations designed for fully teleoperated data is inefficient, as they inherently fail to leverage the critical *temporal* and *spatial* failure patterns revealed by the DAgger process. Right: Our proposed Twin-DAgger explicitly leverages these spatial and temporal patterns to convert a single sparse human correction into a batch of targeted, high-utility training samples concentrated where and when the policy needs them most.

1 Introduction

Imitation learning [28] has proven to be effective for robot manipulation [12, 18], enabling policies to acquire complex visuomotor skills directly from expert demonstrations. However, learned policies often struggle to generalize beyond their training distribution [19]. When deployed in out-of-distribution (OOD) scenarios, their small errors can easily accumulate into a severe distribution shift [28, 39], which eventually leads to task failure.

Dataset Aggregation (DAgger) [13, 28] mitigates the distribution shift of the base policy by leveraging human corrections during its deployment for additional post-training. Yet, conventional DAgger (Fig. 1, left) operates entirely in the real world and remains highly inefficient [16, 36]: Each costly intervention yields only a single trajectory with expert corrections (*i.e.*, a DAgger trajectory), creating a one-to-one ratio between expert effort and data production, which is insufficient if the vast diversity of potential OOD failures [36] is to be captured. To address this inefficiency of DAgger, Digital Twin (DT) techniques [17, 24, 38] , which enable training data to be generated at scale, could be a promising solution. This motivates us to consider leveraging DT techniques to overcome the aforementioned practical ineffectiveness of DAgger and eventually achieve cost-efficient policy robustness.

Directly applying existing DT-based augmentation techniques (Fig. 1, middle) to the DAgger trajectories is technically feasible [8] but inefficient. As these

augmentations originally target fully teleoperated data instead of DAgger data, they are inherently unable to leverage two important failure patterns of the base policy the DAgger process reveals: the *temporal pattern* and the *spatial pattern*. Temporally, distribution shift compounds over time — small errors at critical decision points cascade into larger deviations from the training distribution [28]. The time intervals of human interventions reveal these critical timesteps where the policy begins to drift out-of-distribution. Spatially, the distribution shift does not occur uniformly in space — OOD failures consistently occur more frequently in some regions. Existing augmentation techniques [21, 22, 37, 38] ignore these critical spatial-temporal information, treating all timesteps within the DAgger trajectory and all regions within the workspace as equally important during augmentation. Therefore, their generated data are not informative for post-training the deployed base policy, and the overall efficiency of the DAgger framework remains low.

In this paper, we propose **Twin-DAgger** (Fig. 1, right), which explicitly leverages the failure patterns revealed by DAgger for DT-based targeted data generation. Rather than treating each human intervention as a single data point to consume, Twin-DAgger treats it as both a spatial probe revealing failure-prone regions and a temporal prior marking critical decision points. As shown in Fig. 1, this allows a single correction to seed an entire batch of targeted training samples concentrated where and when the policy needs them most. Specifically, to exploit the temporal pattern, we propose an *intervention-focused trajectory warping* that treats expert intervention timestamps as temporal signals marking critical task stages, generating diverse scenario variations around those moments while skipping already-mastered phases where the policy reliably succeeds. To exploit the spatial pattern, we introduce a *difficulty-based stratified sampling* that uses DAgger rollout outcomes to map the success-rate distribution across the workspace, identifying failure boundaries and concentrating synthetic data generation in high-risk regions. This departs from untargeted uniform coverage with failure-targeted augmentation precisely where the policy is most error-prone. Together, these two mechanisms convert one human correction into a batch of high-utility training samples that directly target the policy's incapability, enabling the post-trained policy to generalize to diverse OOD scenarios with substantially fewer real-world interventions than a conventional DAgger framework. To summarize, the contributions of this work are threefold:

- We introduce Twin-DAgger, a pipeline that uses DT augmentation techniques to multiply the value of each real-world correction, substantially reducing real-world data collection cost.
- We design a dual-axis priority sampling strategy that concentrates generation on failure-prone spatial regions and critical temporal windows identified from expert corrections.
- Through experiments on real-world manipulation tasks, Twin-DAgger outperforms both DT methods and conventional real-world DAgger baselines using only 15% of the human intervention budget.

2 Related Work

2.1 Imitation Learning for Robot Manipulation

Imitation learning enables robot manipulation policies to acquire visuomotor skills from expert demonstrations. Behavior cloning methods such as ALOHA [40] and Diffusion Policy [3] learn action predictions by directly mimicking demonstrations, achieving strong performance on dexterous and multi-modal manipulation tasks. RISE [6] further integrates 2D semantic features with 3D geometric representations for robust action generation. More recently, Vision-Language-Action (VLA) models have scaled imitation learning through pre-training on large, heterogeneous robot datasets. The RT series [2, 41] leverage Open X-Embodiment [25] for zero-shot generalization, OpenVLA [15] combines Internet-scale vision-language data with robot demonstrations, and the π series [1, 11] introduce flow matching-based architectures for general-purpose manipulation. However, these offline-trained policies lack mechanisms to recover from deployment-time errors: once the policy deviates from its training distribution, it cannot self-correct, necessitating interactive supervision to close the loop.

2.2 Interactive Imitation Learning and DAgger

DAgger [28] addresses distribution shift by iteratively collecting expert corrections on states visited by the learned policy, ensuring the policy learns to recover from its own mistakes. Subsequent work has improved DAgger along several axes. HG-DAgger [13] introduces a human-gated intervention mechanism with uncertainty-driven risk metrics, allowing experts to selectively take over control at unsafe states. ThriftyDAgger [7] further reduces expert burden by learning a switching policy that solicits interventions only at states that are sufficiently novel or risky, operating within a fixed human budget. Diff-DAgger [16] leverages diffusion-based uncertainty estimation to predict task failure and trigger timely human interventions. DMD [39] combines diffusion-based policies with DAgger for eye-in-hand manipulation, synthesizing augmented views to reduce the need for real-world corrections. CR-DAgger [36] introduces compliant residual control during human interventions, enabling smoother and more informative corrections for contact-rich tasks. IntervenGen [8] takes a step toward scaling corrections by automatically generating synthetic interventions in simulation from a small set of human demonstrations. However, existing methods either optimize when to intervene or amplify corrections uniformly across the workspace, without considering where and at what stage the policy struggles most. Conversely, our method addresses both spatial and temporal dimensions simultaneously by using intervention signals to guide DT augmentation.

2.3 Sim-to-Real Transfer for Robot Learning

DT technologies construct high-fidelity virtual replicas of real-world environments, enabling scalable synthetic data generation. RoboTwin [24] presents a

procedural generation framework for synthesizing diverse task variations within DT environments. TwinAligner [4] addresses the visual-dynamic alignment challenge by reconstructing simulation environments that match both photorealistic appearance and physical dynamics of real-world scenes. Recently, Twin RL-VLA [35] leverages DT for reinforcement learning-based exploration to refine VLA models before real-world deployment. RialTo [32] proposes to robustify real-world policies by fine-tuning them in DT via reinforcement learning, leveraging an 'inverse distillation' procedure to transfer real-world expertise into simulation. Our Twin-DAgger builds on these reconstruction techniques to bridge the gap between real-world intervention and simulation-based augmentation.

2.4 Digital Twins for Robot Manipulation

Recent work leverages DT to expand limited real-world datasets through synthetic augmentation. RoboSplat [37] utilizes 3D Gaussian Splatting [14] (3DGS) to construct photorealistic virtual environments, enabling robust one-shot manipulation by varying object poses and camera viewpoints. MimicGen [21] automates diverse demonstration generation via object-centric spatial and temporal transformations within simulation. Real2Render2Real [38] utilizes 3DGS to extract object assets and motions from a single video, enabling scalable synthetic data generation by kinematically rendering diverse trajectories and visual contexts. However, these approaches apply uniform augmentation that fails to cover critical failure patterns, and remain decoupled from human expert feedback. In contrast, Twin-DAgger utilizes DAgger intervention signals to direct augmentation toward failure-prone spatial regions and critical temporal windows.

3 Method

3.1 Overview of Twin-DAgger

The Twin-DAgger framework is designed to improve a base policy that has been pre-trained on the initial demonstration data using the DAgger process [28]. The core idea is to adapt existing DT techniques according to the unique insight about the base policy that the DAgger process reveals, augmenting a few trajectories with human corrections into a rich manifold of failure pattern-aware data, thus maximally reducing the post-training data acquisition cost. As illustrated in Fig. 2, the system operates as a closed-loop "train-explore-augment-train" pipeline containing four main components. We briefly describe each pipeline component in the following paragraphs, while our targeted augmentations are further detailed in Sec. 3.2 and Sec. 3.3.

Policy Pre-training and Post-training. Twin-DAgger operates upon a base policy π_θ that is trained on initial demonstration data $\mathcal{D}_{\text{init}}$. By performing DAgger data acquisition and the proposed DAgger-tailored trajectory augmentation (introduced later), a new, augmented set of data $\mathcal{D}_{\text{aug}}$ is collected and

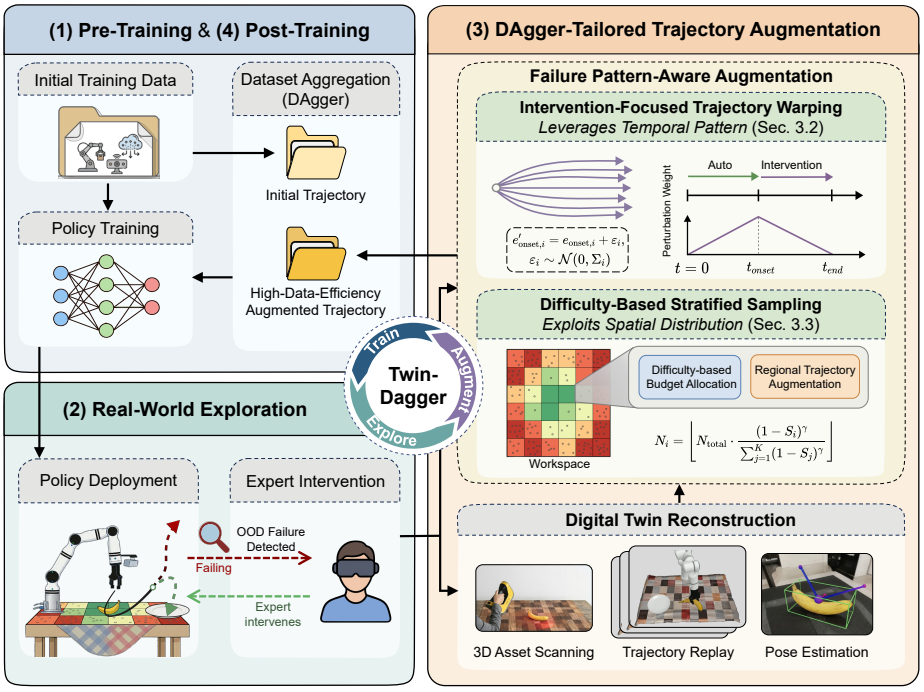

Fig. 2: The Twin-DAgger framework. Twin-DAgger operates as a closed loop containing the following steps: (1) *Policy Pre-training*: a pre-trained base policy is to be improved by Twin-DAgger. (2) *Real-world deployment & expert intervention*: the base policy is deployed in real world to explore its failure patterns and collect corrective trajectories with human interventions. (3) *DAgger-tailored Trajectory Augmentation*: The collected trajectories are augmented within a DT of the real environments, where the temporal failure patterns are leveraged by the proposed intervention-focused trajectory warping (Sec. 3.2), and the spatial patterns are exploited by the difficulty-based stratified sampling (Sec. 3.3). (4) *Policy Post-training*: The augmented data are joined with the original data to post-train the base policy. Finally, the loop closes as this improved policy serves as the new base for the next iteration of deployment, effectively completing the refinement cycle.

feeds back into policy training along with $\mathcal{D}_{\text{init}}$ to form the post-training dataset $\mathcal{D}_{\text{post}} = \mathcal{D}_{\text{init}} \cup \mathcal{D}_{\text{aug}}$, enhancing the generalization of π_θ.

Human Expert-Intervened Exploration. To initiate the DAgger process, π_θ is deployed onto a robot in the real world to explore its failure patterns and collect corrective data. During its deployment, a human expert monitors the robot's execution through a VR-based interface. Upon observing abnormal behaviors — such as a significant deviation from the expected trajectory — the expert switches to teleoperation, provides a brief corrective segment, then hands control back to π_θ. Each successful trajectory τ_θ of length T collected in

such way (*i.e.*, a DAgger trajectory) consists of timestep-wise observations and actions necessary for later augmentation:

$$\tau_\theta = \left\{ \left(\boldsymbol{s}_t,\ c_t \right) \right\}_{t=1}^{T} \cup \boldsymbol{p}_0^{\text{obj}}, \tag{1}$$

where $\boldsymbol{s}_t \in \mathbb{R}^{d_s}$ is the robot's proprioceptive state (*i.e.*, joint angles), and $\boldsymbol{p}_0^{\text{obj}} \in \text{SE}(3)$ is the initial pose of the object to manipulate, estimated via FoundationPose [33]. Crucially, a binary indicator $c_t \in \{0, 1\}$ distinguishes between autonomous execution ($c_t = 0$) and human expert intervention ($c_t = 1$). The above process collects near-failure trajectories that are highly informative for the post-training, but is high-cost in human power as an expert has to continuously monitor the execution.

DT Construction. As a preparation for the DT-based augmentations, within the ManiSkill 3 simulator [31], we construct a DT that matches both dynamics and visual appearance of the real operational environment by importing the robot's Unified Robot Description Format (URDF) model and scanning high-fidelity 3D meshes for surroundings and objects to manipulate. The collected DAgger trajectories can be replayed and augmented within this DT, where the corresponding camera observations and action outcomes that are necessary for training can be obtained in a physically correct manner.

Object-Based Trajectory Warping. To generate new successful trajectories from existing ones, standard object-based trajectory warping methods typically warp the replayed trajectory by either linearly interpolating between the original and modified end-effector poses [38] or adding a trajectory at the beginning to reach the new starting pose [21]. Rather than applying such conventional augmentations designed for purely teleoperated data, we leverage both the temporal and spatial failure patterns exposed during the DAgger process to increase data efficiency. Specifically, we propose two DAgger-tailored augmentation techniques to generate the targeted dataset $\mathcal{D}_{\text{aug}}$: intervention-focused trajectory warping for the temporal aspect (detailed in Sec. 3.2) and difficulty-based stratified sampling for the spatial aspect (detailed in Sec. 3.3). Through these two augmentation techniques, we obtain diverse yet representative data, effectively bridging the gap between sparse expert corrections and the vast OOD spaces.

3.2 Intervention-Focused Trajectory Warping

Not all timesteps in a corrective trajectory carry equal information. The human expert intervenes only when observing failures, and hands control back to the policy when the state returns to normal. Thus, the start of an intervention segment accurately captures the critical failure state of the current policy, while the endpoint means a successful recovery back to the expert manifold. We observe that the local spatial neighborhood surrounding this initial failure point constitutes a dense region of potential OOD states that similarly require corrective actions, thus adding small perturbations to this point yields diverse yet

still error-prone trajectories. Conversely, the in-distribution state at the endpoint must remain untouched to guarantee final task completion. Motivated by this characteristic of corrective trajectories, we propose an intervention-focused trajectory warping technique (Fig. 2, top-right) to exploit the temporal failure patterns of the base policy. It concentrates augmentation around the onset of each intervention to train the base policy to recover from a distribution, rather than a single observed point, of near-failure states.

This technique is superimposed on the aforementioned object-based trajectory warping, which only provides variations of the target object location, at each timestep. Firstly, we partition the trajectory into autonomous and intervention segments to identify the intervention onset. At each onset point, a local Gaussian perturbation is injected to simulate near-failure states. We then apply a two-phase weight scheduling to smoothly interpolate from the original pose to the perturbed pose, followed by sequential Inverse Kinematics to ensure kinematic feasibility.

Separating Intervention Segments. For a complete DAgger trajectory, we partition it into consecutive segments based on the transition points of the intervention indicators c_t. Specifically, we identify N intervention segments, each defined by a starting timestamp $t_{\text{onset},i}$ (where c_t transitions from 0 to 1) and an ending timestamp $t_{\text{end},i}$ (where c_t transitions back to 0). This results in a sequence of alternating segments:

$$\tau = \{\underbrace{\tau_{\text{auto},1}}_{\text{Policy}}, \underbrace{\tau_{\text{int},1}}_{[t_{\text{onset},1}, t_{\text{end},1}]}, \underbrace{\tau_{\text{auto},2}}_{\text{Policy}}, \cdots, \underbrace{\tau_{\text{int},N}}_{[t_{\text{onset},N}, t_{\text{end},N}]}, \underbrace{\tau_{\text{auto},N+1}}_{\text{Policy}}\} \tag{2}$$

where $\tau_{\text{auto},i}$ denotes the i-th policy inference period and $\tau_{\text{int},j}$ denotes the j-th expert intervention period. Although we support multiple segments of intervention, most of our data contains only one interval of intervention. By using these t_{onset} and t_{end} points as anchors, we can isolate segments where the policy failed and the expert provided recovery guidance.

Injecting Perturbation at Intervention Onset. For each identified intervention segment $\tau_{\text{int},i}$ where $i \in \{1, \ldots, N\}$, we focus on the onset moment $t_{\text{onset},i}$ where the policy deviates severely from the expert manifold. To broaden the policy's corrective range, we inject a local perturbation into the initial EE pose $\boldsymbol{e}_{\text{onset},i}$, while keeping the corresponding endpoint pose $\boldsymbol{e}_{\text{end},i}$ unperturbed. The perturbed initial pose $\boldsymbol{e}'_{\text{onset},i}$ is given by:

$$\boldsymbol{e}'_{\text{onset},i} = \boldsymbol{e}_{\text{onset},i} + \boldsymbol{\varepsilon}_i, \quad \boldsymbol{\varepsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_i), \tag{3}$$

where $\boldsymbol{\varepsilon}$ is a Gaussian noise vector and $\boldsymbol{\Sigma}_i$ controls perturbation magnitude in translation and rotation. By selecting a $\boldsymbol{\Sigma}_i$ with proper scale, the warped trajectory increases data diversity while remaining representative of failure cases by avoiding pushing the robot into unrecoverable states that lie beyond the distribution of the current demonstration.

Weight Scheduling. To maintain kinematic continuity across the augmented trajectory, for each intervened segment $\tau_{\text{int},i}$, we apply a two-phase weight scheduling mechanism centered at its onset $t_{\text{onset},i}$, influencing the segment $\tau_{\text{auto},i}$ and $\tau_{\text{int},i}$ expanding a time interval from timestep $t_{\text{end},i-1}$ (or 0 if $i = 1$) to $t_{\text{end},i}$. The warping strength $w(t)$ in this period is defined as follows. From the previous intervention's end (or the start of the trajectory) up to the current intervention onset, or the pre-intervention phase, the warping strength ramps up from 0 to 1:

$$w(t) = \frac{t - t_{\text{end},i-1}}{t_{\text{onset},i} - t_{\text{end},i-1}}, \quad t \in (t_{\text{end},i-1}, t_{\text{onset},i}]. \tag{4}$$

This ensures the initial state of the trajectory remains unchanged while smoothly transitioning to the perturbed state at the moment of failure. During the expert correction phase, the strength decays from 1 back to 0:

$$w(t) = 1 - \frac{t - t_{\text{onset},i}}{t_{\text{end},i} - t_{\text{onset},i}}, \quad t \in (t_{\text{onset},i}, t_{\text{end},i}]. \tag{5}$$

This ensures that the augmented trajectory gradually converges to the original expert manifold, reaching the unperturbed recovery point at $t_{\text{end},i}$. We then warp the end effector from the original pose toward the perturbed pose at each timestep t, during which $w(t)$ serves as the weight of both spherical linear interpolation (Slerp) [30] for rotations and the linear interpolation for translations.

Obtaining Robot State from EE Pose. To map these Cartesian EE poses back to the robot's joint space, we employ an Inverse Kinematics (IK) solver frame-by-frame. To maintain kinematic continuity and prevent joint-space discontinuities, each IK solve is started using the joint configuration from the previous frame. This process yields smooth and physically executable trajectories that simulates the error-then-correction dynamics, effectively transforming a single DAgger demonstration into a diverse set of augmented training samples.

3.3 Difficulty-Based Stratified Sampling

The demand for training samples varies significantly across different regions of the robot's workspace: In regions with high task success rates, such as the middle of the workspace, a minimal amount of DAgger intervention data is sufficient to maintain policy stability; conversely, more supervisory samples are needed in low-success-rate regions. To maximize the data efficiency of the augmented trajectories, we propose a difficulty-based stratified sampling strategy (Fig. 2, middle-right) to effectively leverage the spatial failure patterns of the base policy. This strategy quantifies the operational difficulty of different spatial regions and selectively strengthens the model's performance in error-prone regions under a total resource budget.

Difficulty-based Augmentation Budget Allocation. We first partition the workspace into K sub-regions and record each region's empirical success rate S_i

during the exploration stage. Then, given total budget N_{total} of new trajectories to generate, each region's allocation is given by:

$$N_i = \left\lfloor N_{\text{total}} \cdot \frac{(1 - S_i)^\gamma}{\sum_{j=1}^{K} (1 - S_j)^\gamma} \right\rfloor, \tag{6}$$

where an empirically-set concentrating factor $\gamma = 5$ further up-weight high-failure-rate regions ($S_i \approx 0$).

The resulting quotas $\{N_i\}$ govern the trajectory augmentation, concentrating corrective data on spatial regions where the policy is most deficient.

Trajectory Augmentation in Regions. Once the regional quotas $\{N_i\}$ are determined, we utilize them to guide the large-scale synthesis of training data. For each region, we select existing DAgger corrective trajectories within the region to serve as seeds. To generate the required N_i samples, we randomly perturb the target object's pose within the spatial bounds of the region.

For each perturbed object pose, we apply the aforementioned trajectory warping mechanisms, using both object-based and intervention-focused trajectory warping to generate a diverse set of augmented corrective trajectories. By aggregating these synthesized trajectories across all regions, we construct a balanced training dataset that is concentrated in high-difficulty areas, thereby efficiently strengthening corner-case robustness while maintaining global stability.

Using temporal intervention-focused trajectory warping and spatial difficulty-based stratified sampling, we successfully transformed few real-world DAgger trajectories into the a large amount of targeted data $\mathcal{D}_{\text{aug}}$. As illustrated in the top-left component of Fig. 2, these generated data are sent to the Dataset Aggregation module to post-train the base policy π_θ, closing the "train-explore-augment-train" loop and yielding a robust policy capable of recovering from failure states.

4 Experiments

In this section, we provide some implementation details in Sec. 4.1, describe the asynchronous policy inference system in Sec. 4.2, perform comparisons of the proposed Twin-DAgger against conventional data collection frameworks in Sec. 4.3, and conduct ablation studies on the proposed key components in Sec. 4.4. More implementation details and experimental results can be found in the supplementary material.

4.1 Implementation Details

Hardware. Our real-world experiments are conducted on a RealMan RM-75-6FB [27] robot equipped with an Inspire [9] EG2-4C2 1-DoF parallel gripper. The robot features 7-DoF with a maximum reach of ∼640mm. For visual perception, we employ a dual-camera setup at 640×480 resolution: a RealSense [10] D435 camera in third-person view captures the global workspace, while a wrist-mounted RealSense D405 provides an egocentric view.

Teleoperation and Data Collection. We construct a teleoperation system using the Meta Quest 3 [23] VR headset for two purposes: (1) collecting initial demonstration data for baseline policy training, and (2) enabling expert intervention during the DAgger correction phase when policies encounter failure modes. For each manipulation task, we collect 100 human demonstrations to train baselines and 50 real-world correction demonstrations where experts intervene to provide corrective actions. Demonstrations are recorded at 30 Hz, capturing synchronized observations from both cameras along with robot joint states and end-effector poses.

Training Details. We evaluate all methods on two representative policies: ACT [40] and π_0 [1], adopting the RoboCOIN [34] and OpenPI [26] implementations, respectively. Both policies predict delta joint actions and absolute gripper positions over a 50-step prediction horizon. ACT uses a batch size of 64 and trains for 100K steps with the AdamW [20] optimizer at a constant learning rate of 1×10^{-4}. π_0 uses a batch size of 32 and trains for 50K steps, following the OpenPI training protocol with a peak learning rate of 5×10^{-5} and exponential moving average (EMA, decay rate 0.999) for model weight stabilization. All experiments run on a single NVIDIA H200 GPU with 142 GB memory.

4.2 Asynchronous Policy Inference

Policy learning requires smooth, temporally consistent demonstrations [29], yet naive synchronous inference introduces significant latency between observation acquisition and action execution, causing trajectory stuttering that degrades data quality. To ensure DAgger data meets the necessary quality standard, we implement an asynchronous inference mechanism for all policies during the DAgger process.

System Architecture. Our asynchronous inference system decouples policy inference from action execution through a dual-thread architecture. The *inference thread* continuously queries the policy network via a high-performance Remote Procedure Call (RPC) framework and writes predicted action chunks into a thread-safe ring buffer, while the *execution thread* operates at a fixed frequency f_{exec} to retrieve and execute actions from the buffer, holding the last action when the buffer is empty to maintain trajectory continuity.

Temporal Alignment and Scheduling. A critical challenge is temporal misalignment: with execution frequency f_{exec}, the inference latency ℓ_{inf} causes the first $n_{\text{clip}} = \lceil \ell_{\text{inf}} \cdot f_{\text{exec}} \rceil$ actions of each predicted chunk $\boldsymbol{a} = [a_0, \ldots, a_{L-1}]$ to become outdated by the time it arrives. We therefore clip these stale actions and write only the valid remainder $\boldsymbol{a}_{\text{valid}} = [a_{n_{\text{clip}}}, \ldots, a_{L-1}]$ to the buffer, while adaptively scheduling the next query from the remaining valid actions and ℓ_{inf} to avoid buffer depletion. In our experiments, we set $f_{\text{exec}} = 30$ Hz for both ACT and π_0, achieving buffer hold rates below 2% for smooth, continuous rollouts throughout DAgger collection.

Table 1: Success rates with different augmentation strategies. "#Collected by Expert" denotes the number of real-world corrective trajectories provided by the human expert via VR intervention. "#After DT Augmentation" indicates the total number of synthetic trajectories generated within the DT to form the post-training dataset.

Augmentation Strategy	**#Collected by Expert**	**#After DT Augmentation**	**Success Rate**	
			ACT [40]	π_0 [1]
None (Base Policy)	0	0	30.0%	55.0%
Real2Render2Real [38]	0	100	33.0%	55.0%
	0	500	45.0%	43.3%
Twin-DAgger (w/o Spatial)	6	100	46.7%	68.3%
Twin-DAgger (w/o Temporal)	6	100	48.3%	76.7%
Twin-DAgger (Full)	6	100	55.0%	85.0%

4.3 Comparison

To evaluate Twin-DAgger, we compare it against two baselines: 1) Conventional DAgger, which collects expert corrections directly from real-world deployment without any DT augmentation, and 2) Real2Render2Real (R2R2R) [38], a state-of-the-art DT augmentation method. To ensure a fair comparison, we re-implement R2R2R's object-based trajectory warping within ManiSkill 3 [31] to share the same DT environment as Twin-DAgger. We report success rate of ACT [40] and π_0 [1].

Task and Scene Setup. As illustrated in Fig. 2, all experiments are conducted on a pick-and-place task in a workspace positioned directly in front of the robot arm. We first collect 100 teleoperation demonstrations via our teleoperation system to train the base policy π_θ. For the DAgger exploration stage, we divide the workspace uniformly into 6 regions; within each region, the target object is randomly placed and π_θ is evaluated for 10 trials. A human expert monitors each rollout and intervenes whenever task failure is imminent, guiding the robot back on track; any trial requiring intervention is recorded as a failure. After all 60 trials complete, we proceed to DT augmentation using one failure trajectory per region — 6 real-world DAgger corrections in total — as seeds for targeted data synthesis, followed by policy post-training. Success is defined as completing the task within 2000 timesteps without human intervention.

Performance against Other Augmentation. Tab. 1 reports average success rate (SR) across all six workspace regions, Fig. 4 provides the per-region breakdown. ACT and π_0 trained solely on the initial 100 demonstration trajectories achieve 30.0% and 55.0% SR, respectively. We then utilize R2R2R to generate 100 augmented samples, combine them with the original 100 demonstrations, and conduct post-training. This yields only marginal improvement for

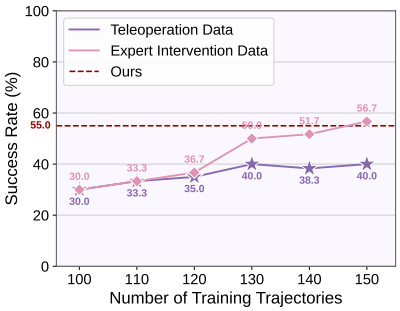

Fig. 3: Comparison against two common post-training strategies: teleoperation and conventional DAgger.

Fig. 4: Visualization of success rates across workspace regions for base policy, R2R2R, and Twin-DAgger.

ACT (30.0% → 33.0%) and no gain for π_0, showing that applying DT augmentation to purely teleoperated data is largely ineffective. We further scale R2R2R to 500 augmented samples, again combined with the same 100 initial demonstrations for post-training. This improves ACT to 45.0%, yet even with 5× the data budget, R2R2R still falls 10% behind Twin-DAgger at equal sample count. For π_0, performance actually drops to 43.3%, which falls below the base policy, suggesting that large-scale naive DT augmentation during post-training may cause the policy to forget previously well-learned scenarios, effectively increasing the proportion of OOD cases rather than resolving them. This reveals a fundamental limitation of failure-agnostic augmentation: indiscriminate data generation not only fails to cover the policy's actual failure states, but can actively erode its existing competence. In contrast, post-training on 100 Twin-DAgger augmented trajectories — combined with the same 100 initial demonstrations — achieves 55.0% for ACT and 85.0% for π_0.

Data Efficiency vs. Conventional DAgger. To evaluate the cost efficiency of Twin-DAgger, we compare it against two real-world post-training data source by incrementally augmenting the initial 100 demonstrations with additional trajectories: (1) Teleoperation data, adding fully-teleoperated demonstrations, and (2) expert intervention data, adding human-corrected DAgger trajectories from real-world deployment — representing conventional DAgger. We report average SR as a function of total training trajectories in Fig. 3. Both real-world strategies improve with more data, but with diminishing returns. Supplementary teleoperation data plateaus near 40.0% after 30 additional trajectories, confirming that more in-distribution data alone cannot resolve distribution shift. Conventional DAgger makes steadier progress but still requires 50 real-world expert interventions to reach 56.7%. Twin-DAgger, by contrast, matches that result with just 6 real-world DAgger corrections — roughly 15% of the 40 interventions conventional DAgger requires.

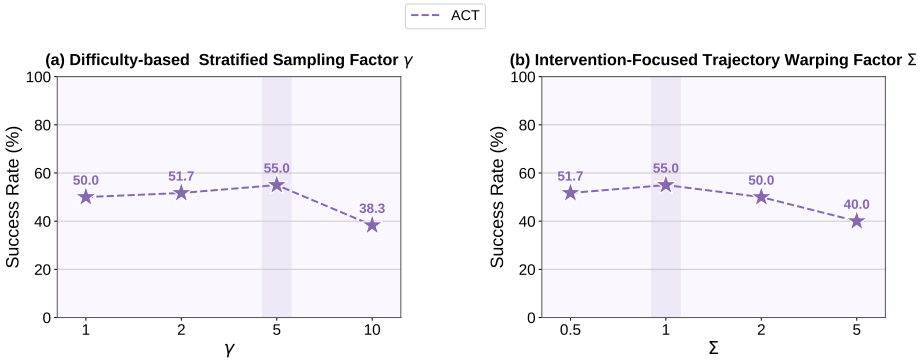

Fig. 5: Ablation study results for the spatial focusing factor γ (a) and temporal perturbation scale Σ (b).

Spatial Analysis. Fig. 4 visualizes the per-region success rates. The base policy (a) fails consistently across the workspace, with right-column regions uniformly at 0.1–0.2, indicating systematic difficulty under lateral displacement. R2R2R with 100 samples (b) improves only the center-left region, leaving all others unchanged. Scaling to 500 R2R2R samples (c) broadly lifts the left column but leaves the top-right region entirely unimproved despite 5× the budget, because uniform augmentation has no mechanism to direct samples toward persistently failing locations. Our method with only 100 augmented samples (d) breaks this pattern: top-right improves from 0.2 to 0.6 and top-left from 0.4 to 0.9, a direct consequence of difficulty-based stratified sampling concentrating the augmentation budget on high-failure-rate regions.

4.4 Ablation Study

We first conducted ablation study to evaluate the two proposed strategies individually. As shown in Tab. 1, both components individually already outperform DT baseline at equal data budget, confirming that each failure signal — temporal and spatial — carries irreplaceable value. Their combination achieves 55.0%/85.0%, a larger gain than either alone, indicating that the two mechanisms target complementary failure modes rather than overlapping ones.

We then examine the sensitivity of each component to its key hyperparameter. As shown in Fig. 5, for the spatial concentrating factor γ, performance peaks at γ=5, where failure-prone regions receive substantially more budget while easy regions retain a small non-zero allocation, preventing catastrophic forgetting on already-mastered scenarios. Increasing to γ=10 degrades performance to 38.3%: over-concentration on the hardest regions narrows the training distribution and leaves the policy brittle on ordinary cases. For the temporal perturbation magnitude Σ, Σ=0.5 produces perturbations too small to meaningfully expand coverage beyond the original intervention state, yielding ∼51.7% average. At Σ=5, performance collapses to 40%: perturbations of this magnitude

displace the robot into configurations far outside the task-relevant workspace, filling augmented trajectories with anomalous, task-irrelevant actions that corrupt the policy's learning of normal behavior rather than reinforcing corrective responses. Based on these results, we adopt γ=5 and Σ=1 as defaults.

5 Conclusion

In this paper, we proposed Twin-DAgger that treats each DAgger trajectory as a spatial probe marking where the policy broke down and a temporal prior identifying when drift initiated, allowing a small number of interventions to seed a much larger set of targeted augmented trajectories within a DT. The two proposed mechanisms each exploited one axis of the failure patterns: Difficulty-based stratified sampling concentrated data generation on failure-prone workspace regions, and intervention-focused trajectory warping ensured those trajectories captured the precise moments policy errors initiated. On real manipulation tasks, Twin-DAgger matched conventional DAgger performance at 15% of the intervention cost and outperformed uniform DT augmentation by 20% in success rate under equal data budgets. The gains come not from generating more data, but from generating it in the right places at the right moments — a distinction that turns out to matter considerably in imitation learning.

Limitations. One limitation of Twin-DAgger is that, as we need to replay the real-world trajectory in DT, Twin-DAgger primarily supports rigid-body manipulation, making it highly challenging to reconstruct and import flexible or deformable objects (*e.g.*, cloth or string). Future work could explore integrating advanced scene representations, such as dynamic 3D Gaussian Splatting or neural radiance field, to effectively model and augment the deformations of such objects in the DT. Another limitation is that our DAgger data is collected using a VR device, requiring all joint angles to be solved through IK. Near the boundaries of the workspace, this reliance on IK can result in unnatural joint configurations. In the future, we may utilize a robot-isomorphic exoskeleton [5] to directly capture more physically reasonable joint angles. Finally, as Twin-DAgger builds upon DAgger, it inherits a fundamental limitation of correction-based learning: failures that have already become unrecoverable by the time an expert can react, such as shattering a fragile object, cannot be remedied by any post-hoc correction. Handling such cases would require predictive intervention, *e.g.*, guided by tactile feedback, which we leave to future work.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant Nos. 62136001 and 52505029), Beijing High Innovation Plan (Contract No. 20260 4841443), and Science and Technology Commission of Shanghai Municipality (Grant No. 25ZR1401191).

References

1. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L.X., Tanner, J., Vuong, Q., Walling, A., Wang, H., Zhilinsky, U.: π_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)
2. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: RT-1: robotics transformer for real-world control at scale. In: Proc. of Robotics Science and Systems (RSS) (2023)
3. Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. International Journal of Robotics Research (IJRR) **44**(10-11), 1684–1704 (2025)
4. Fan, H., Dai, H., Zhang, J., Li, J., Yan, Q., Zhao, Y., Gao, M., Wu, J., Tang, H., Dong, H.: TwinAligner: Visual-dynamic alignment empowers physics-aware real2sim2real for robotic manipulation. arXiv preprint arXiv:2512.19390 (2025)
5. Fang, H., Fang, H., Wang, Y., Ren, J., Chen, J., Zhang, R., Wang, W., Lu, C.: AirExo: Low-cost exoskeletons for learning whole-arm manipulation in the wild. In: Proc. of IEEE International Conference on Robotics and Automation (ICRA). pp. 15031–15038 (2024)
6. Fang, H., Wang, C., Wang, Y., Chen, J., Xia, S., Lv, J., He, Z., Yi, X., Guo, Y., Zhan, X., Yang, L., Wang, W., Lu, C., Fang, H.: AirExo-2: Scaling up generalizable robotic imitation learning with low-cost exoskeletons. In: Proc. of Conference on Robot Learning (CoRL). pp. 198–220 (2025)
7. Hoque, R., Balakrishna, A., Novoseller, E.R., Wilcox, A., Brown, D.S., Goldberg, K.: ThriftyDAgger: Budget-aware novelty and risk gating for interactive imitation learning. In: Proc. of Conference on Robot Learning (CoRL). vol. 164, pp. 598–608 (2021)
8. Hoque, R., Mandlekar, A., Garrett, C.R., Goldberg, K., Fox, D.: IntervenGen: Interventional data generation for robust and data-efficient robot imitation learning. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2840–2846 (2024)
9. Inspire Robots: EG2-4C2 Electric Gripper (2025), `https://en.inspire-robots.com/product/eg2-4c`. Accessed: 2025-11-13
10. Intel: D435: A powerful, full-featured depth camera (2022), `https://realsenseai.com/stereo-depth-cameras/stereo-depth-camera-d435/`. Accessed: 2025-11-14
11. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Galliker, M.Y., Ghosh, D., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., LeBlanc, D., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Ren, A.Z., Shi, L.X., Smith, L., Springenberg, J.T., Stachowicz, K., Tanner, J., Vuong, Q., Walke, H., Walling, A., Wang, H., Yu, L., Zhilinsky, U.: $\pi_{0.5}$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)
12. Jia, Y., Liu, J., Chen, S., Gu, C., Wang, Z., Luo, L., Li, X., Wang, P., Wang, Z., Zhang, R., Zhang, S.: Lift3D policy: Lifting 2D foundation models for robust 3D robotic manipulation. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17347–17358 (2025)
13. Kelly, M., Sidrane, C., Driggs-Campbell, K.R., Kochenderfer, M.J.: HG-DAgger: Interactive imitation learning with human experts. In: Proc. of IEEE International Conference on Robotics and Automation (ICRA). pp. 8077–8083 (2019)

14. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d Gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (TOG) **42**(4), 139:1–139:14 (2023)
15. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E.P., Sanketi, P.R., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: OpenVLA: An open-source vision-language-action model. In: Proc. of Conference on Robot Learning (CoRL). vol. 270, pp. 2679–2713 (2024)
16. Lee, S., Kang, X., Kuo, Y.: Diff-Dagger: Uncertainty estimation with diffusion policy for robotic manipulation. In: Proc. of IEEE International Conference on Robotics and Automation (ICRA). pp. 4845–4852 (2025)
17. Lim, V., Huang, H., Chen, L.Y., Wang, J., Ichnowski, J., Seita, D., Laskey, M., Goldberg, K.: Real2Sim2Real: Self-supervised learning of physical single-step dynamic actions for planar robot casting. In: Proc. of IEEE International Conference on Robotics and Automation (ICRA). pp. 8282–8289 (2022)
18. Liu, J., Chen, H., An, P., Liu, Z., Zhang, R., Gu, C., Li, X., Guo, Z., Chen, S., Liu, M., et al.: HybridVLA: Collaborative diffusion and autoregression in a unified vision-language-action model. arXiv preprint arXiv:2503.10631 (2025)
19. Liu, J., Gao, F., Wei, B., Chen, X., Liao, Q., Wu, Y., Yu, C., Wang, Y.: What Can RL Bring to VLA Generalization? An Empirical Study. In: Proc. of Advances in Neural Information Processing Systems (NeurIPS) (2025)
20. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: Proc. of International Conference on Learning Representations (ICLR). OpenReview.net (2019)
21. Mandlekar, A., Nasiriany, S., Wen, B., Akinola, I., Narang, Y., Fan, L., Zhu, Y., Fox, D.: MimicGen: A data generation system for scalable robot learning using human demonstrations. In: Proc. of Conference on Robot Learning (CoRL). vol. 229, pp. 1820–1864 (2023)
22. Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., Fei-Fei, L., Savarese, S., Zhu, Y., Martín-Martín, R.: What matters in learning from offline human demonstrations for robot manipulation. In: Proc. of Conference on Robot Learning (CoRL). vol. 164, pp. 1678–1690 (2021)
23. Meta: Meta Quest 3: Next-Gen Virtual Reality Headset. (2023), `https://www.meta.com/quest/quest-3/`. Accessed: 2026-02-28
24. Mu, Y., Chen, T., Chen, Z., Peng, S., Lan, Z., Gao, Z., Liang, Z., Yu, Q., Zou, Y., Xu, M., Lin, L., Xie, Z., Ding, M., Luo, P.: RoboTwin: Dual-arm robot benchmark with generative digital twins. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 27649–27660 (2025)
25. O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., et al.: Open X-Embodiment: Robotic learning datasets and RT-X models. In: Proc. of IEEE International Conference on Robotics and Automation (ICRA). pp. 6892–6903 (2024)
26. Physical Intelligence team: Training codebase for π series models. (2024), `https://github.com/Physical-Intelligence/openpi`. Accessed: 2026-02-28
27. Realman: RM75-6F: 7-DOF Robotic Arm 6-Axis-Force Version (2024), `https://www.realman-robotics.com/rm75-6f.html`. Accessed: 2025-11-13
28. Ross, S., Gordon, G.J., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Proc. of International Conference on Artificial Intelligence and Statistics (ICAIS). vol. 15, pp. 627–635 (2011)

29. Sakr, M., Van der Loos, H.M., Kulic, D., Croft, E.: Consistency Matters: Defining Demonstration Data Quality Metrics in Robot Learning from Demonstration. ACM Transactions on Human-Robot Interaction (2024). https://doi.org/10.1145/3773904
30. Shoemake, K.: Animating rotation with quaternion curves. In: Proc. of the ACM SIGGRAPH Conference and Exhibition On Computer Graphics and Interactive Techniques (SIGGRAPH). pp. 245–254 (1985)
31. Tao, S., Xiang, F., Shukla, A., Qin, Y., Hinrichsen, X., Yuan, X., Bao, C., Lin, X., Liu, Y., Chan, T., Gao, Y., Li, X., Mu, T., Xiao, N., Gurha, A., Huang, Z., Calandra, R., Chen, R., Luo, S., Su, H.: ManiSkill3: GPU Parallelized Robotics Simulation and Rendering for Generalizable Embodied AI (2024)
32. Villasevil, M.T., Simeonov, A., Li, Z., Chan, A., Chen, T., Gupta, A., Agrawal, P.: Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. In: Proc. of Robotics Science and Systems (RSS) (2024)
33. Wen, B., Yang, W., Kautz, J., Birchfield, S.: FoundationPose: Unified 6D pose estimation and tracking of novel objects. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17868–17879 (2024)
34. Wu, S., Liu, X., Xie, S., Wang, P., Li, X., Yang, B., Li, Z., Zhu, K., Wu, H., Liu, Y., et al.: RoboCOIN: An open-sourced bimanual robotic data collection for integrated manipulation. arXiv preprint arXiv:2511.17441 (2025)
35. Xu, Q., Liu, J., Zhou, R., Shi, S., Han, N., Liu, Z., Gu, C., Gu, S., Yue, Y., Huang, G., et al.: TwinRL-VLA: Digital Twin-Driven Reinforcement Learning for Real-World Robotic Manipulation. arXiv preprint arXiv:2602.09023 (2026)
36. Xu, X., Hou, Y., Liu, Z., Song, S.: Compliant residual DAgger: Improving real-world contact-rich manipulation with human corrections. In: Proc. of Advances in Neural Information Processing Systems (NeurIPS) (2025)
37. Yang, S., Yu, W., Zeng, J., Lv, J., Ren, K., Lu, C., Lin, D., Pang, J.: Novel demonstration generation with Gaussian splatting enables robust one-shot manipulation. In: Proc. of Robotics Science and Systems (RSS) (2025)
38. Yu, J., Fu, L., Huang, H., El-Refai, K., Ambrus, R.A., Cheng, R., Irshad, M.Z., Goldberg, K.: Real2Render2Real: Scaling robot data without dynamics simulation or robot hardware. In: Proc. of Conference on Robot Learning (CoRL). pp. 547–577 (2025)
39. Zhang, X., Chang, M., Kumar, P., Gupta, S.: Diffusion meets DAgger: Supercharging eye-in-hand imitation learning. In: Proc. of Robotics Science and Systems (RSS) (2024)
40. Zhao, T.Z., Kumar, V., Levine, S., Finn, C.: Learning fine-grained bimanual manipulation with low-cost hardware. In: Proc. of Robotics Science and Systems (RSS) (2023)
41. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: RT-2: Vision-language-action models transfer web knowledge to robotic control. In: Proc. of Conference on Robot Learning (CoRL). pp. 2165–2183 (2023)

Twin-DAgger: Synergizing Digital Twins and Human Corrections for Efficient Robot Manipulation

Supplementary Material

Jiahang Li[1,2#], Zhirui Zhang[1,2#], Kairan Ding[1,2], Fan Fei[1,2], Yunkai Tang[1,2], Jiaming Liu[1], Xiao He[2], Ziyu Chen[2], Gaohao Zhou[2], Jieji Ren[3], Yandong Guo[2], Shanghang Zhang[1], and Boxin Shi[1*]
[1]Peking University [2]AI[2] Robotics [3]Shanghai Jiao Tong University

jiahangli25@stu.pku.edu.cn shiboxin@pku.edu.cn

In this supplementary material, we provide additional implementation details of Twin-DAgger in Appendix A, additional results in Appendix B, cross-task generalization results in Appendix C, and the attached video.

A Additional Implementation Details

A.1 Teleoperation and Data Collection Pipeline

During deployment, an expert monitors the robot in real time. When the robot exhibits abnormal behavior such as deviating significantly from its target trajectory, the expert switches to VR teleoperation mode, provides a brief corrective segment to restore normal operation, then releases control back to the policy. The expert is not required to demonstrate complete task execution. Instead, they provide only short, targeted corrective interventions that return the robot to the expert manifold or a recoverable state. This sparse correction mechanism reduces the expert's effort in collecting informative corrective trajectories when the robot encounters out-of-distribution (OOD) scenarios during deployment, which are then used for subsequent training. In practice, DAgger interventions typically require only 10–15% of the effort needed for full teleoperation demonstrations. Fig. 3 demonstrates that DAgger corrective data produces larger performance gains over the base policy compared to an equivalent volume of standard teleoperated demonstrations.

# Equal contribution. * Corresponding author. [1] Jiahang Li, Zhirui Zhang, Kairan Ding, Fan Fei, Yunkai Tang, Jiaming Liu, Shanghang Zhang, and Boxin Shi are with State Key Laboratory of Multimedia Information Processing, National Engineering Research Center of Visual Technology, and PKU-AI[2] Robotics Joint Lab of Embodied AI, School of Computer Science, Peking University. [3] Jieji Ren is with State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University.

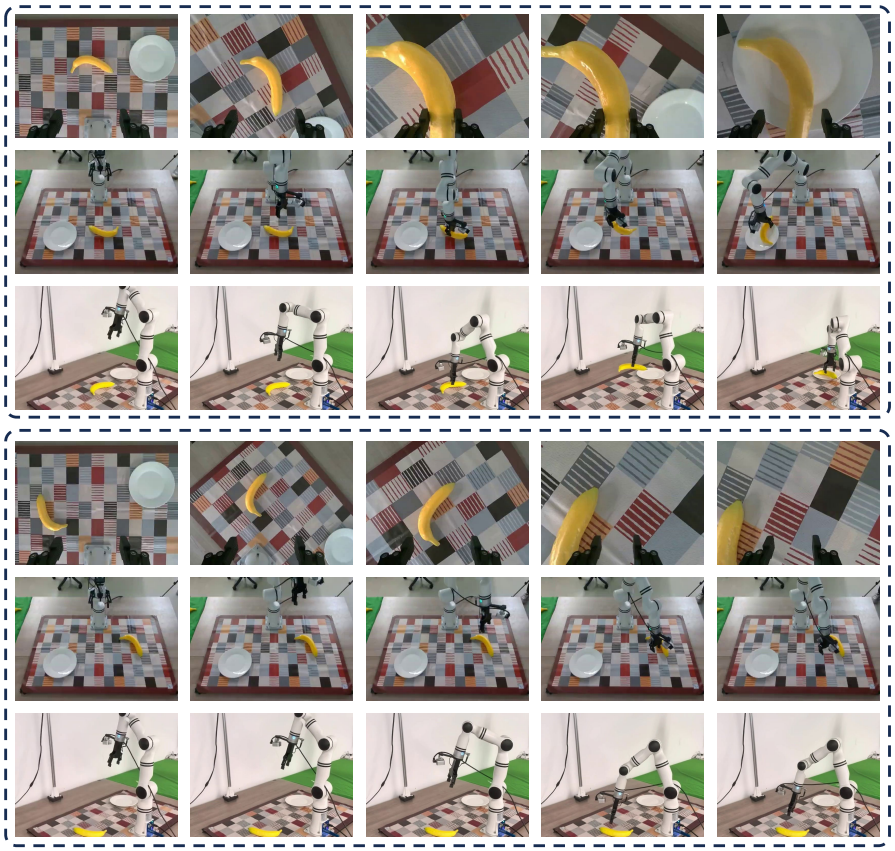

Fig. S1: Visualization of the base policy (trained on initial data) during inference (illustrated from left to right).

A.2 Digital Twin Construction

To support subsequent digital twin (DT) replay and data augmentation, we utilize the ManiSkill 3 simulator [31] to construct a high-fidelity virtual environment. To enhance data diversity, we perform large-scale augmentation of the DAgger data within the simulation platform. First, we randomize environmental lighting by configuring various combinations of ambient, directional, and point light sources, capturing visual data under diverse illumination conditions. Next, we vary the background of the workspace and introduce distracted objects to the table to improve the generalization of the model. Finally, the simulator's rendering capabilities allow us to efficiently generate novel-view demonstrations by sampling camera poses in the local neighborhood of the original third-person view.

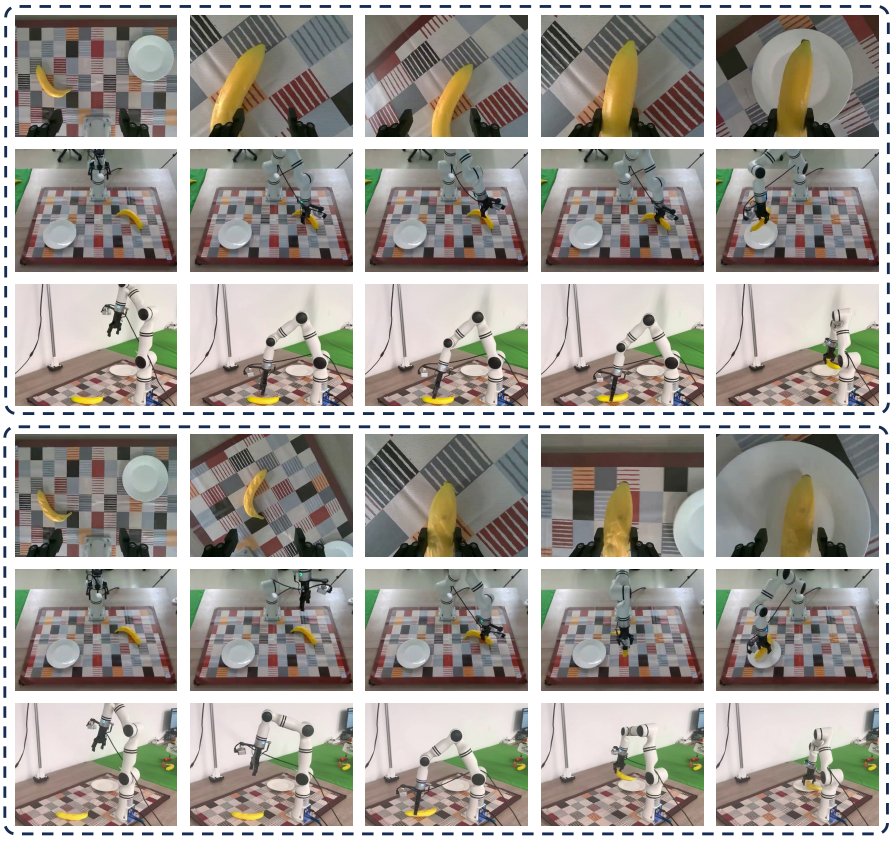

Fig. S2: The DAgger data collection process and the subsequent deployment of the post-trained policy (illustrated from left to right).

A.3 Trajectory Warping and Physical Feasibility

Twin-DAgger generates augmented data by warping each replayed DAgger trajectory toward new object configurations. The resulting end-effector poses are converted to joint commands with the Pinocchio dynamics library as the IK solver, where each frame is initialized from the joint configuration solved for the previous frame. This warm-start seeding suppresses discontinuous joint solutions and preserves the smoothness of the warped trajectory. To ensure that the augmented data remains physically executable, the augmentation stage applies collision detection in addition to the task-success criterion, retaining only trajectories that satisfy both conditions.

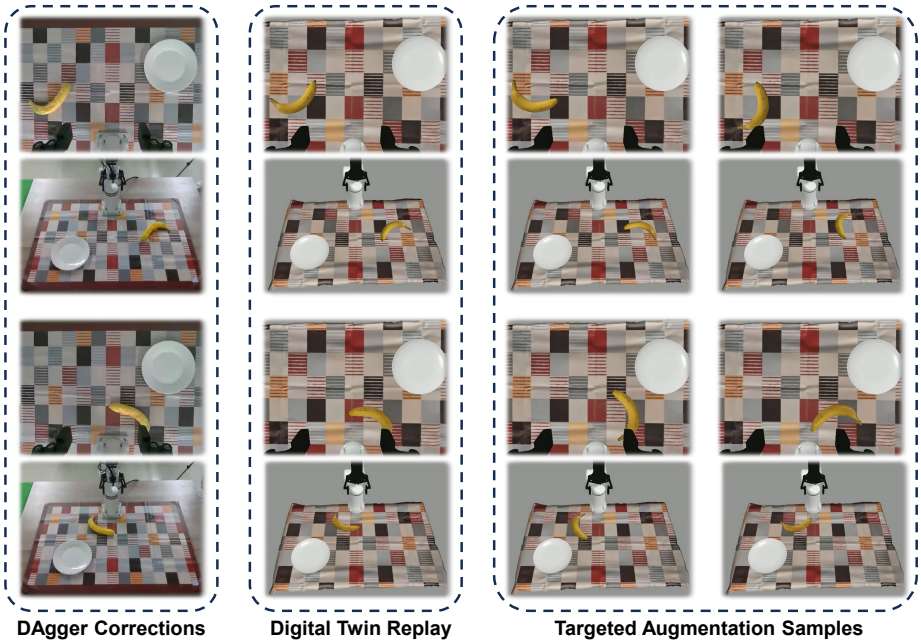

Fig. S3: The augmented data generation process of Twin-DAgger in a Digital Twin (DT) environment.

A.4 Workspace Partitioning for Stratified Sampling

Our difficulty-based stratified sampling distributes the augmentation budget over a partition of the robot's reachable workspace. We choose the partition to match this reachable region: for the RealMan RM-75 arm used in our experiments, the area with minimal motion constraints is a rectangle of $50\,\text{cm} \times 32\,\text{cm}$, which we divide into a 3×2 grid of six regions. For a different arm, the partition is adapted to its own reachable workspace; the effectiveness of the spatial allocation strategy does not depend on this particular choice.

B Additional Results

B.1 Demonstration

To illustrate the Twin-DAgger process, Fig. S1 and Fig. S2 show synchronized frames from a wrist-mounted camera and two third-person cameras during policy deployment. In Fig. S1, the base policy succeeds in an in-distribution (ID) scenario (top) but fails when encountering an OOD object placement (bottom). Fig. S2 then demonstrates how a human expert corrects this failure to collect DAgger data (top), and how the post-trained policy successfully completes the exact same OOD scenario that previously caused failure (bottom).

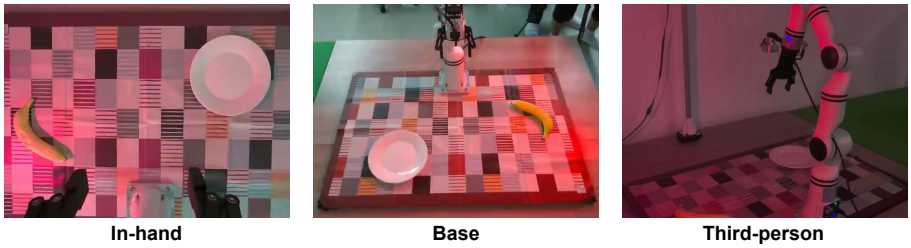

Fig. S4: Unseen environmental setup with red lighting for zero-shot robustness evaluation.

Table S1: Zero-shot robustness evaluation of base policy and Twin-DAgger under unseen environmental conditions.

Strategy	Success Rate	
	ACT [40]	π_0 [1]
Base Policy	0%	36.7%
Twin-DAgger	26.7%	70.0%

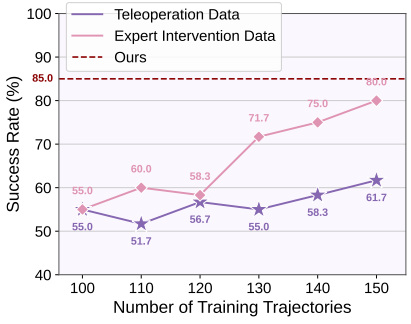

Fig. S5: Comparison of Twin-DAgger against two common post-training data sources for π_0: teleoperation data and conventional DAgger data.

We further illustrate the DT augmentation pipeline in Fig. S3. The first column shows the real-world DAgger trajectory, which is replayed within the DT. Using this replay as a seed, Twin-DAgger generates diverse augmented samples through intervention-focused trajectory warping and difficulty-based stratified sampling.

B.2 Robustness Evaluation

To evaluate the generalization ability of Twin-DAgger, we conduct a zero-shot robustness evaluation under unseen visual conditions. Specifically, we apply a red lighting setting (Fig. S4) that differs substantially from the training environment. As reported in Tab. S1, the base ACT policy completely fails under this perturbation, with π_0 dropping from 55.0% to 36.7%. In contrast, Twin-DAgger retains 26.7% for ACT and 70.0% for π_0. This robustness stems from the visual augmentations applied within the DT, including randomized lighting, objects, which expose the policy to diverse visual conditions during training, improving its resilience to unseen perturbations at deployment.

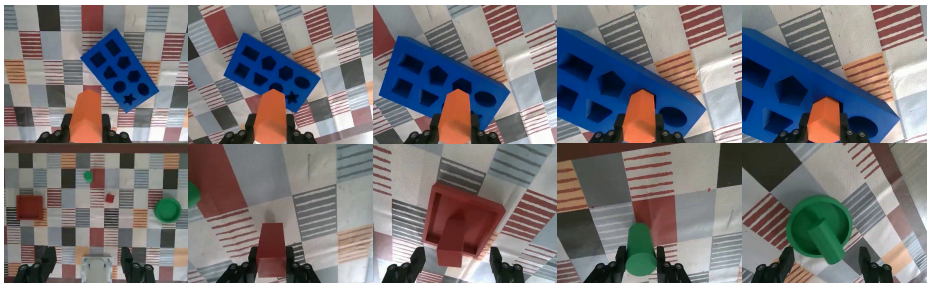

Fig. S6: Visualization of the two additional tasks used to evaluate the cross-task generalization of Twin-DAgger: the contact-rich Peg-in-Hole (PIH) task (top) and the long-horizon Categorical-Sorting (CS) task (bottom), illustrated from left to right.

Beyond visual perturbations, the simulation-based augmentation also lets us improve robustness to physical variation by randomizing the physical parameters of the augmented rollouts. As a concrete instance, we post-train the base ACT policy on augmented trajectories with randomized gripper–object friction, applying the corresponding adjustments to the gripper actions, and evaluate it in the real world under varied friction conditions realized with gripper fingertips of different surface materials. The policy trained with friction randomization reaches 20.0% success, compared with 15.0% without randomization and 3.3% for the base policy, showing that physical domain randomization during augmentation improves robustness to real-world dynamics.

B.3 Additional Data Efficiency Comparison

Fig. 3 in the main paper evaluates the cost efficiency of Twin-DAgger with the ACT policy. Here we repeat the same comparison with π_0 (Fig. S5). Both real-world strategies improve with more data: supplementary teleoperation plateaus near 61.7% after 50 additional trajectories, confirming that in-distribution data alone cannot resolve distribution shift. Conventional DAgger makes steadier progress but still requires 50 real-world expert interventions to reach 80.0%. In contrast, Twin-DAgger achieves 85.0% with only six real-world corrections, surpassing this level with significantly fewer interventions.

C Cross-Task Generalization

The experiments in the main paper focus on a pick-and-place setting. To examine whether the benefits of Twin-DAgger transfer to manipulation problems with different demands, we further evaluate it on two additional tasks that stress complementary capabilities (Fig. S6): (1) the contact-rich Peg-in-Hole (PIH) task, in which a hexagonal block must be inserted into a matching slot under a strict 3 mm tolerance that demands high manipulation precision; and (2) the long-horizon Categorical-Sorting (CS) task, in which the robot must sequentially sort a red block and a green cylinder into receptacles of the matching color. Tabs. S2

Table S2: Success rates of different augmentation strategies on the contact-rich **Peg-in-Hole (PIH)** task.

Augmentation Strategy	**#Collected by Expert**	**#After DT Augmentation**	**Success Rate**	
			ACT [40]	π_0 [1]
None (Base Policy)	0	0	16.7%	15.0%
VR Teleoperation	50	50	23.3%	16.7%
Conventional DAgger	50	50	46.7%	35.0%
Real2Render2Real [38]	6	100	31.7%	26.7%
Twin-DAgger (Ours)	6	100	43.3%	41.7%

Table S3: Success rates of different augmentation strategies on the long-horizon **Categorical-Sorting (CS)** task.

Augmentation Strategy	**#Collected by Expert**	**#After DT Augmentation**	**Success Rate**	
			ACT [40]	π_0 [1]
None (Base Policy)	0	0	13.3%	33.3%
VR Teleoperation	50	50	16.7%	48.3%
Conventional DAgger	50	50	28.3%	43.3%
Real2Render2Real [38]	6	100	23.3%	40.0%
Twin-DAgger (Ours)	6	100	33.3%	60.0%

and S3 report the success rates of ACT and π_0 under the same set of augmentation strategies as in our main experiments. On both tasks, Twin-DAgger surpasses VR teleoperation and matches conventional DAgger while using substantially fewer expert corrections, and it consistently outperforms R2R2R under an equal training budget. These results indicate that the advantages of Twin-DAgger extend from pick-and-place to both contact-rich and long-horizon manipulation.