EGORECOVERY: ACQUIRING FAILURE RECOVERY ABILITY THROUGH HUMAN RECOVERY DEMONSTRATION

Anonymous authors
Paper under double-blind review



Figure 1: **Overview of EGORECOVERY.** Egocentric human recovery data collection yields about $10\times$ more data and broader coverage of failure states than robot teleoperation collection, while EGORECOVERY transfers this signal through a gated corrective intent bottleneck.

ABSTRACT

Robust embodied robots should be able to recover from failures and retry tasks in order to operate reliably in unstructured and noisy real-world environments. Achieving this capability requires training policies on data that captures recovery behaviors. However, collecting such data through robot teleoperation is difficult to scale, as it is time-consuming to induce diverse failure states, perform corrective actions, and reset the environment. This challenge is further exacerbated by the high diversity of failure modes, which demands substantially more recovery data than success demonstrations. In this work, we show that egocentric human data capturing failure recovery processes provides a scalable alternative. By efficiently arranging task-level failure configurations and recording short recovery segments, human operators can generate more than $10\times$ as much valid recovery data per hour compared to robot teleoperation under our protocol. To address the embodiment gap between human and robot, we propose EGORECOVERY, a co-training framework for learning recovery behavior, where human recovery demonstrations are aligned to a compact corrective-intent space shared with robot data, which captures the timing and magnitude of correction. Only a small number of robot recovery demonstrations are required to connect this intent to executable robot actions. At deployment, a learned recovery gate predicts when correction is needed from robot observations and activates the corrective intent only in recovery states. Experiments on real-world recovery tasks show that EGORECOVERY improves task success from failure states over robot-only recovery, direct co-training with human recovery data, and direct intent-transfer baselines.

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

Robust embodied robots require policies that can recover from their own mistakes. Policies trained solely on successful demonstrations perform well from clean initial states but degrade when closed-loop rollouts drift away from the demonstrated manifold, a well-known instance of covariate shift in imitation learning (Ross et al., 2011; Laskey et al., 2017). Recent methods demonstrate that explicit recovery and correction data can substantially improve long-horizon robustness (Hu et al., 2025; Pan et al., 2026). However, collecting such data is costly. Unlike successful demonstrations, acquiring a recovery trajectory often requires repeatedly inducing or reaching a meaningful failure state, verifying it, teleoperating a corrective action, and resetting the hardware. Moreover, because failure modes can be highly diverse, covering the distribution of possible off-manifold states demands many varied trajectories, further increasing the data-collection cost of learning recovery behaviors.

Recently, learning from egocentric human data has shown strong potential to improve the generalization of robot policies (Bahl et al., 2022; Wang et al., 2023; Kareer et al., 2025; Punamiya et al., 2025; Zhu et al., 2026; Cai et al., 2025; Punamiya et al., 2026; Zheng et al., 2026b), yet these methods use human data primarily for nominal task execution or broad pretraining. In this work, **we exploit the speed advantage of human egocentric data collection to gather failure-recovery data as efficient coverage of diverse failure modes**. A human demonstrator can collect such data quickly by setting up task-level failure configurations, recording only short recovery segments, and moving rapidly to the next failure mode. In our setting, measured as accepted recovery demonstrations per operator hour under the collection protocol summarized in Table 1 (full protocol in Appendix F), this yields $10\times$ more data than robot teleoperation collection.

Deciding what to share across embodiments is a central challenge in cross-embodiment imitation (Punamiya et al., 2025; Xu et al., 2026; Han et al., 2026; Xie et al., 2026). Human hands and robot grippers differ in kinematics, contact mechanics, control frequency, and action representation. Sharing the action decoder can bias the robot toward actions that do not match its embodiment, while simply adopting a different decoder might not fully utilize the information in human recovery data. Thus, we propose to share *corrective intent* rather than full human motion across embodiments. Corrective intent describes the timing and magnitude of a short correction within a recovery segment. This choice follows cross-embodiment imitation methods that share compact behavior descriptions rather than copy trajectories (Xu et al., 2023; Kim et al., 2025; Huang et al., 2026). It is also well matched for learning recovery behaviors, since a recovery segment starts from a failure configuration and aims to return the task to a feasible state. Human recovery data can indicate when to retreat, realign, or retry, and how much adjustment is needed. The robot policy still learns direction, contact, and executable actions from robot data.

Specifically, we propose EGORECOVERY, a joint training method that maps human recovery supervision to robot recovery through corrective intent. It shares a visual trunk across embodiments and keeps embodiment-specific action decoders, following HPT (Wang et al., 2024). A shared Intent Head extracts corrective intent from human and robot recovery segments, aligning the two domains in a compact intent representation. Moreover, a Recovery Gate Head is trained to predict whether recovery is needed from current observations and states, supervised by recovery and success phase labels. The predicted gate modulates features in the robot decoder, allowing corrective intent to affect recovery states while preserving nominal execution learned from robot data.

In summary, **our contributions are as follows:**

- We introduce egocentric human recovery as a scalable source of recovery supervision that covers diverse task-level failure states and provides more than $10\times$ as much valid recovery supervision per operator hour as robot teleoperation under our protocol.
- We propose EGORECOVERY, which transfers human recovery through corrective intent rather than human action labels, with an Intent Head for shared intent and a Recovery Gate Head for deciding when it should affect robot actions.
- We evaluate real robot recovery under controlled data budgets and show that EGORECOVERY improves closed-loop recovery over robot-only recovery, human success co-training, direct co-training with human recovery data, and an always-on intent modulation variant.

2 Related Work

Recovery supervision and egocentric human data. Recovery supervision addresses the covariate shift that arises when imitation policies leave the demonstration manifold. DAgger collects labels on states induced by the policy (Ross et al., 2011), DART injects noise so that demonstrations cover perturbed states (Laskey et al., 2017), and later interactive methods decide when to query the expert (Kelly et al., 2019; Hoque et al., 2022; Mandlekar et al., 2022). Recent robot systems collect recovery trajectories (Hu et al., 2025; Pan et al., 2026), respawn failed states during rollouts (Zheng et al., 2026a), correct generative policies online (Welte et al., 2026), and generate corrective behavior with learned world models (Yu et al., 2026; Li et al., 2026c). All of them acquire corrective supervision on the robot, so covering diverse failure states costs robot time. RaC scales robot recovery and correction data and reports improved data efficiency. The $10\times$ ratio in this paper measures collection throughput in accepted recovery demonstrations per operator hour. Section 4.3 compares performance at matched collection effort.

Egocentric collection requires no robot time and yields more demonstrations per operator hour than robot teleoperation. Wearable-camera systems record the operator performing the task with bare hands (Liu et al., 2025; Bahl et al., 2022; Wang et al., 2023), learn visual representations and affordances from such recordings (Nair et al., 2023; Singh et al., 2025; Shi et al., 2025), co-train them with robot demonstrations (Kareer et al., 2025; Cai et al., 2025; Zhang et al., 2026c), adapt the human domain to the robot domain (Punamiya et al., 2025; Xu et al., 2026), or share a visual trunk while keeping embodiment-specific action decoders (Wang et al., 2024). Ma et al. (2026) survey this literature. Handheld grippers such as UMI record with a robot end effector in hand, which narrows the embodiment gap enough to use the recorded trajectories as action labels, at the cost of carrying the device and fixing the grasp to the mounted gripper (Chi et al., 2024). Across these interfaces the human data supervise nominal execution, broad pretraining, or action alignment.

Transferable representations across embodiments. Human recovery segments cannot serve as robot demonstrations unless shared task information is separated from embodiment-specific execution. Cross-embodiment imitation methods transfer compact behavior descriptions, including latent plans from play data (Lynch et al., 2020), cross-embodiment skills (Xu et al., 2023; Kim et al., 2025), actionable affordances (Heidinger et al., 2025; Han et al., 2026), human intention priors (Xie et al., 2026; Huang et al., 2026), and latent action variables learned contrastively or by joint alignment (Zhang et al., 2026a; Dai et al., 2026; Luo et al., 2026a). These targets describe what a demonstration accomplishes over an episode. Corrective intent is narrower: it encodes the timing and magnitude of one short correction. Direction, contact, and executable actions are learned from robot data, and the Recovery Gate Head activates the corrective intent only in recovery states.

3 Method

EgoRecovery co-trains on human and robot recovery data with a shared corrective intent target and embodiment-specific action decoders. Human recovery broadens the coverage of failure states, while robot recovery grounds corrective intent in executable robot actions. The Recovery Gate Head scales the intent modulation applied to the robot decoder. Figure 2 gives an overview of the proposed method. Section 3.1 defines the data sources, Section 3.2 describes how to extract the compact and shared corrective intent, and Section 3.3 explains how the intent is used to decode robot actions.

3.1 Human and Robot Data Collection

Robot recovery data provides executable corrective actions, but it is expensive to scale because each episode requires staging a failure state, checking that it is valid, teleoperating the correction, and resetting the hardware. Collecting egocentric human recovery data offers a much cheaper alternative. As shown in Table 1, under the same protocol, including staging, reset, synchronization, and discarded attempts, human recovery provides about $10\times$ more accepted recovery demonstrations per operator hour than teleoperated robot recovery.

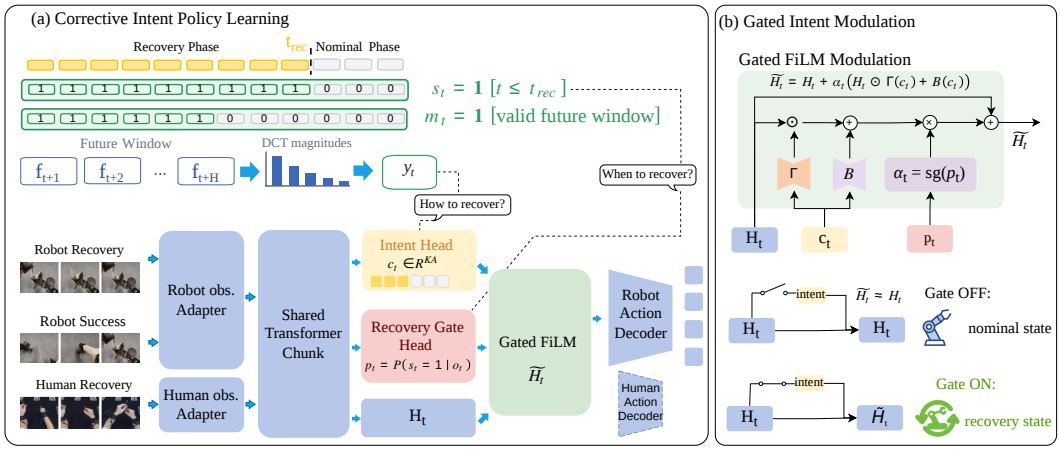


Figure 2: **EGORECOVERY method. a,** During training, robot success preserves nominal execution, robot recovery grounds corrective intent in executable robot actions, and human recovery broadens the coverage of failure states. Recovery annotations define the recovery state label s_t and the intent mask m_t, which supervise the Recovery Gate Head and the Intent Head, respectively. **b,** At deployment, the robot uses only robot observations. The predicted intent c_t modulates decoder features through gated FiLM only when the predicted gate p_t indicates that recovery is needed.

Wearable cameras have enabled large egocentric datasets (Grauman et al., 2022; Hoque et al., 2025; Li et al., 2026d) used in recent VLA pretraining (Li et al., 2026b; Yang et al., 2025; Feng et al., 2025; Bi et al., 2025; Luo et al., 2026b; Wang et al., 2026; Wei et al., 2026) and world-model training (Gao et al., 2026). We use this collection mode to record short human recovery segments staged around task-level failures.

Data types. Each task has four data types: robot success, robot recovery, human success, and human recovery. Robot success provides nominal robot action labels, and robot recovery provides failure states and corrective actions. Human demonstrations supervise the shared trunk and the auxiliary human motion decoder, and human recovery additionally supplies the corrective intent target. The default per-task counts are Robot success (50), Robot recovery (50), Human success (300), and Human recovery (300). Appendix F details collection and annotation.

Recovery labels. Each recovery episode starts from a failure state and contains a recovery phase followed by nominal execution. A motion-energy curve computed from smoothed hand or end-effector motion proposes the temporal boundary t_{rec}. An annotator confirms or adjusts it to the frame where corrective motion ends. We set $s_t = 1$ for $t \le t_{\text{rec}}$ and $s_t = 0$ afterward to supervise the Recovery Gate Head described in Section 3.3. The intent mask m_t selects frames whose full future motion window remains within the recovery phase and passes the padding, truncation, missing-motion, and near-stationary checks. Success episodes set $s_t = m_t = 0$, providing negative gate examples and excluding nominal execution from corrective intent supervision.

3.2 CORRECTIVE INTENT FOR HUMAN-TO-ROBOT RECOVERY LEARNING

Human recovery can guide robot recovery through information that remains meaningful across embodiments. Human hands and robot grippers differ in contact, kinematics, and control. We therefore use corrective intent as an intermediate supervision target shared by both embodiments. Corrective intent describes the timing and magnitude of the remaining correction over a short future window. The gate p_t separately predicts whether the current observation is a recovery state. Direction, contact, and executable robot actions are learned from robot observations and robot action supervision.

For each frame selected by m_t, let $P^d_{t,a} \in \mathbb{R}^{L\times 3}$ denote the positions of active effector a in embodiment d over a future window of L points. Robot end-effector positions are taken from the logged robot action trajectories, and human wrist positions are reconstructed in 3D from egocentric video (Appendix G). Let s_d be the corresponding embodiment scale. To make this target compact and less sensitive to small timing or contact differences, we project the normalized future motion onto a

low-frequency discrete cosine transform (DCT) basis:

$$\Delta P^d_{t,a}[\ell] = \frac{P^d_{t,a}[\ell] - P^d_{t,a}[0]}{s_d}, \qquad y^d_t = \left[\left\| \left(C_K \Delta P^d_{t,a} \right)_{k,:} \right\|_2 \right]_{a=1:A,\ k=0:K-1} \in \mathbb{R}^{KA}. \tag{1}$$

Here C_K contains the first K low-frequency DCT basis vectors over the future window. The low-frequency DCT coefficients keep the coarse timing of the motion, and the spatial norm removes direction-specific displacement and yields a magnitude profile for each active effector. Thus y_t records how the remaining correction magnitude unfolds over the window. In the main experiments, one active effector and four temporal coefficients give a 4D target.

The intent mask determines where this target can supervise the policy. Applying the intent loss only when m_t is true prevents nominal motion or invalid future segments from becoming examples of corrective intent. Human and robot recovery both provide this target. Human recovery covers many related failure states, and robot recovery pairs the target with robot action labels, from which the robot decoder learns to map the predicted intent to executable robot actions.

3.3 Gated Intent Modulation for Robot Recovery Execution

The predicted intent should affect robot action decoding only when correction is needed. If the decoder received the intent at every time step, recovery supervision could interfere with behavior learned from robot success data. We therefore train the Recovery Gate Head to predict from the current observation whether c_t should modulate the robot decoder. Runtime monitors provide a similar signal with a separate failure detector (Liu et al., 2023; 2024). Here the gate is trained with the policy from the same recovery phase labels and only scales the intent modulation.

Following HPT (Wang et al., 2024), the policy uses a heterogeneous backbone with embodiment-specific adapters and decoders around a shared trunk. Each observation is encoded by its embodiment adapter and processed by the shared trunk. A pooled representation feeds two shared prediction heads. The Intent Head predicts $c_t \in \mathbb{R}^{KA}$, and the Recovery Gate Head predicts p_t, the probability that the observation requires recovery behavior. The robot decoder maps robot features to robot actions, while an auxiliary human decoder maps human features to human motion. Each decoder is supervised only by actions from its own embodiment: the auxiliary human motion loss trains the human decoder, and robot actions train the robot decoder. The shared trunk and the Intent Head are trained on recovery observations from both embodiments.

The robot decoder receives c_t through a gated residual FiLM modulation (Perez et al., 2018) before the output projection W_r. Let $H_t \in \mathbb{R}^D$ be the robot decoder feature before this projection, and let $\Gamma, B : \mathbb{R}^{KA} \to \mathbb{R}^D$ map c_t to feature-wise scale and shift terms. We set the scalar gate $\alpha_t = \mathrm{sg}(p_t)$, which broadcasts over feature dimensions and stops gradients of the imitation loss through the gate in the modulation path. The residual form makes the identity path explicit:

$$\widetilde{H}_t = H_t + \alpha_t \left(H_t \odot \Gamma(c_t) + B(c_t) \right), \qquad \hat{a}_t = W_r(\widetilde{H}_t). \tag{2}$$

When p_t is near zero, $\widetilde{H}_t$ reduces to the base decoder feature. When p_t is high, the residual term injects corrective intent before action prediction. On robot recovery frames selected by m_t, the same observation contributes an intent loss on c_t and a robot action loss through the modulated decoder. The robot action loss also updates the Intent Head through $\Gamma(c_t)$ and $B(c_t)$, and the masked intent loss keeps c_t close to the shared target y_t.

The training objective combines behavior cloning with the intent loss, the gate loss, and the nominal regularizer:

$$\mathcal{L} = \mathcal{L}^r_{\mathrm{bc}} + \mathcal{L}^h_{\mathrm{bc}} + \lambda_c \mathcal{L}_c + \lambda_g \mathcal{L}_g + \lambda_n \mathcal{L}_n. \tag{3}$$

Here $\mathcal{L}^r_{\mathrm{bc}}$ is behavior cloning on robot success and robot recovery samples, and $\mathcal{L}^h_{\mathrm{bc}}$ is behavior cloning for the auxiliary human decoder only. The intent loss $\mathcal{L}_c = \mathbb{E}_{m_t=1}[\|c_t - y_t\|_1]$ supervises the predicted intent with the shared target on valid recovery windows. The gate loss $\mathcal{L}_g = \mathbb{E}[\mathrm{BCE}(p_t, s_t)]$ trains recovery state detection. The nominal regularizer $\mathcal{L}_n = \mathbb{E}_{s_t=0}[p_t(\|\Gamma(c_t)\|_2^2 + \|B(c_t)\|_2^2)]$ penalizes nonzero intent modulation on non-recovery frames, weighted by p_t, so that a gate false positive on nominal data also drives the modulation maps toward zero. The main experiments use $K = 4$, $\lambda_c = \lambda_g = 0.05$, and $\lambda_n = 0.01$.

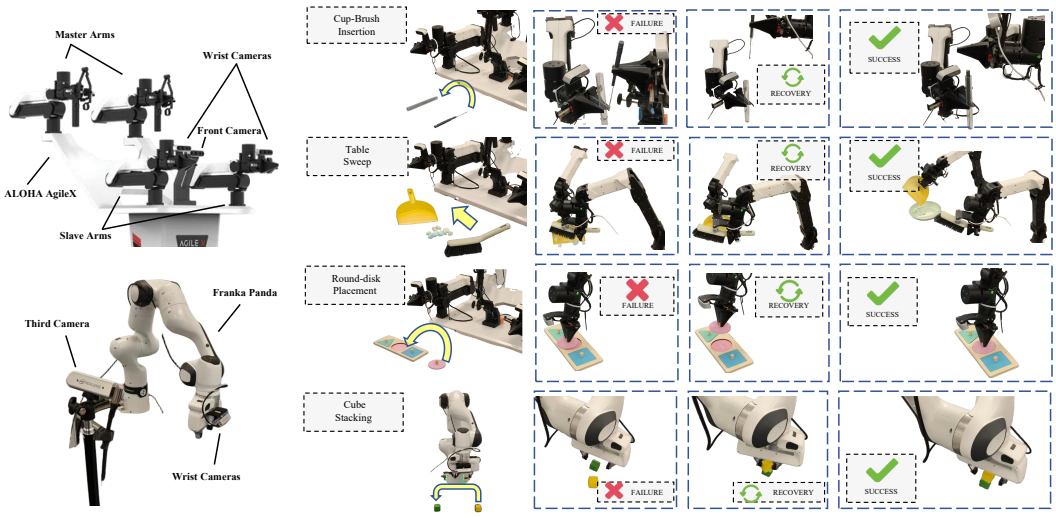


Figure 3: **Experimental platforms and task objects.** Real-robot tabletop settings used for closed-loop evaluation across four manipulation tasks: cup brush insertion, table sweep, round disk placement, and cube stacking.

At deployment, the robot uses only robot observations, the predicted intent, the predicted gate, and the robot decoder. Recovery labels and intent targets are used only during training. Appendix H and Appendix I give implementation details for the target construction, window length, scale computation, sampling ratios, and the remaining hyperparameters.

4 Experiments

4.1 Experimental Setup

Protocol. All evaluations use closed-loop rollouts on real robots. The evaluated policy receives robot observations alone, and human data and recovery annotations are used only for training. *Initial SR* and *Recovery SR* measure task completion from standard initial states and from failure states, respectively. These failure states lie outside the state distribution covered by the success demonstrations. A rollout is successful if the robot completes the task within 5 minutes. Each method is evaluated on the same 20 initial states and 20 failure states per task. Appendix C gives the rollout rules and success criteria. Figure 3 and Appendix B describe the robot platforms, and Appendix Figure 7 shows the egocentric human recovery collection setup.

Tasks. We evaluate four tabletop manipulation tasks: cup brush insertion, table sweep, round disk placement, and cube stacking. The first three tasks run on an ALOHA AgileX dual-arm system, and cube stacking runs on a Franka Panda arm. They span retreat-and-realign, corrective sweeping, lift-realign-place, and stack stabilization. Evaluation uses the same task-level failure families as training, with held-out failure configurations.

Baselines. All demonstration counts are per task. Robot success (n) denotes n robot success demonstrations, Robot recovery (n) denotes n robot recovery demonstrations, Human success (n) denotes n human success demonstrations, and Human recovery (n) denotes n human recovery demonstrations. For comparisons at matched collection effort, we define the robot-equivalent recovery budget as $B_{\text{eq}} = n_r^R + n_h^R/10$. The $1/10$ conversion uses the measured ratio of accepted human to robot recovery demonstrations per operator hour in Table 1. Equal B_{eq} therefore represents comparable operator time under our collection protocol. The direct human recovery mix applies standard egocentric human–robot co-training (Kareer et al., 2025; Cai et al., 2025) to the same recovery demonstrations and backbone as EgoRecovery, without the intent loss, Recovery Gate Head, or gated modulation. The robot-only references are Robot recovery (50) and, at matched collection effort, Robot recovery (80) (Figure 6), which scale robot recovery demonstrations as in RaC (Hu et al., 2025).

4.2 Main Results

Figure 4 summarizes per-task Initial and Recovery SR for the main training configurations. We first establish the effect of recovery data and then evaluate the additional benefit of transferring human recovery through EgoRecovery. Appendix J provides the corresponding success counts and Wilson 95% confidence intervals.

R_{suc}	R_{rec}	H_{suc}	H_{rec}	Variant	Cup brush		Table sweep		Round disk		Cube stack		**Avg.**	
					Init.	Rec.	Init.	Rec.	Init.	Rec.	Init.	Rec.	Init.	Rec.
50				Base	30	0	40	0	45	0	35	0	37.5	0.0
50		300		Human success	40	0	45	0	50	0	45	0	45.0	0.0
50			300	Human recovery	40	5	40	10	45	10	45	10	42.5	8.8
50	50			Robot recovery	50	55	60	50	60	55	55	50	56.3	52.5
50	50	300		Success mix	55	60	65	70	70	65	65	65	63.8	65.0
50	50		300	Direct mix	65	70	70	75	80	70	70	70	71.2	71.2
50	**50**		**300**	**EgoRecovery**	**75**	**80**	**80**	**90**	**85**	**85**	**80**	**85**	**80.0**	**85.0**

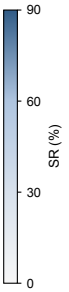


Figure 4: **Main closed-loop results across four tabletop tasks.** Each task group reports Initial SR and Recovery SR (%) for cup brush, table sweep, round disk, cube stack, and their average. In the method labels, R_{suc}, R_{rec}, H_{suc}, and H_{rec} denote robot success, robot recovery, human success, and human recovery demonstration counts.

Recovery demonstrations enable recovery from failure states. The two success-only settings, Robot success (50) and Robot success (50) + Human success (300), both achieve 0.0 average Recovery SR, whereas adding Robot recovery (50) raises it to 52.5, establishing the robot-recovery baseline.

Robot recovery grounds human recovery in executable actions. Without robot recovery, Robot success (50) + Human recovery (300) reaches 8.8 average Recovery SR. With Robot recovery (50), the direct human recovery mix reaches 71.2, and EgoRecovery reaches 85.0 while obtaining the highest average Initial SR of 80.0. Under object and background variations (Appendix K), adding 300 human recovery demonstrations collected in those settings to EgoRecovery trained on original-setting data, with no further robot data, raises Initial SR from 35.0 to 60.0 on cup brush insertion and from 50.0 to 75.0 on round disk placement, and Recovery SR from 40.0 to 55.0 and from 55.0 to 60.0. Human recovery thus supplies recovery supervision across varied failure states and settings, while robot recovery provides embodiment-specific action supervision.

Corrective intent supervision and gated modulation improve over direct co-training. The direct mix and EgoRecovery use the same data (Robot success (50)+Robot recovery (50)+Human recovery (300)) and the same backbone. Adding the intent loss, Recovery Gate Head, and gated modulation raises average Recovery SR from 71.2 to 85.0. EgoRecovery is higher than the direct mix on all four tasks and on both metrics, with per-task Recovery SR gains of 10 to 15 points. Section 4.3 compares robot and human recovery at matched collection effort, and Section 4.4 isolates these components.

4.3 Data Collection Efficiency

We evaluate collection efficiency through measured collection throughput, human data scaling, and allocations of robot and human recovery demonstrations at matched collection effort. Table 1 reports 9.8–11.4$\times$ higher human recovery throughput across the four tasks, with a mean of 10.5$\times$, which sets the $1/10$ conversion used in B_{eq}. Figure 5 varies the amount and type of human data while holding Robot success (50) and Robot recovery (50) fixed.

At all four amounts of human data, EgoRecovery with human recovery outperforms the human-success baseline. From 200 to 800 demonstrations, average Initial SR rises from 73.8 to 87.5 and Recovery SR from 75.0 to 90.0, while the human-success baseline rises only from 55.0 to 73.8 and from 58.8 to 68.8. With Human recovery (300), EgoRecovery already exceeds the Human success (800) baseline on both metrics. The two curves differ in both the type of human data and the training method. With the same human recovery data, Figure 4 isolates the training method: EgoRecovery reaches 85.0 Recovery SR and the direct mix 71.2.

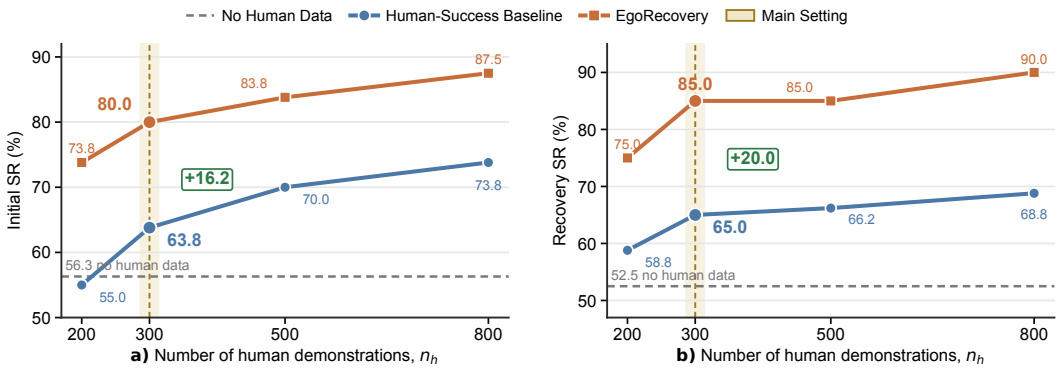


Figure 5: **Human data scaling with fixed robot recovery data. a)** Average Initial SR. **b)** Average Recovery SR. All points hold Robot success (50) and Robot recovery (50) fixed. The human-success baseline adds Human success (n), whereas EGORECOVERY adds Human recovery (n) through corrective intent transfer.

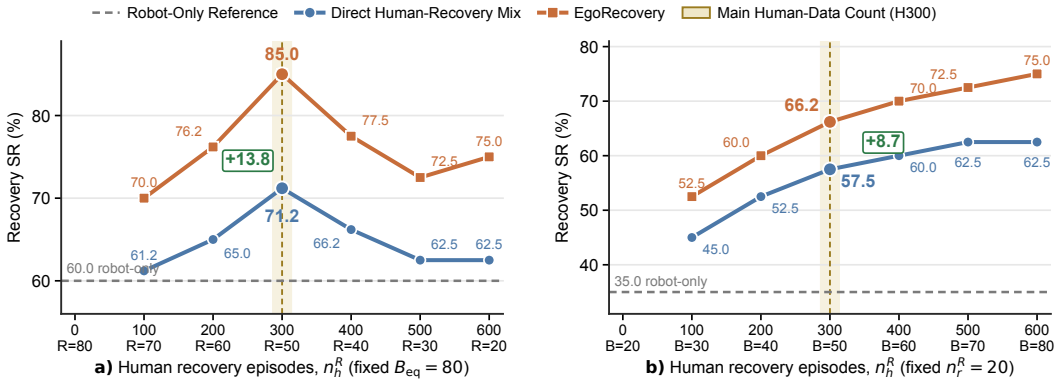


Figure 6: **Recovery budget comparison.** Average Recovery SR across four tasks with Robot success (50) fixed. We define $B_{\text{eq}} = n_r^R + n_h^R/10$ using the measured human-to-robot collection-rate ratio. **a)** Robot and human recovery allocations at fixed $B_{\text{eq}} = 80$; the second tick row gives n_r^R. **b)** Human recovery scaling with $n_r^R = 20$; the second tick row gives the resulting B_{eq}.

Figure 6 varies the allocation of robot and human recovery data. Appendix Tables 10 and 11 report the per-task counts and confidence intervals.

At the fixed robot-equivalent recovery budget $B_{\text{eq}} = 80$ in Figure 6(a), the best tested allocation, Robot recovery (50)+Human recovery (300), reaches 85.0 average Recovery SR, compared with 60.0 for Robot recovery (80) and 71.2 for the direct mix. Allocations with fewer robot recovery demonstrations reach 72.5 to 77.5. With Robot recovery (20) fixed in panel (b), increasing human recovery from 100 to 600 demonstrations raises EGORECOVERY from 52.5 to 75.0 and the direct mix from 45.0 to 62.5. EGORECOVERY benefits more from the added human recovery than the direct mix, and the gap between the two widens from 7.5 to 12.5 points over this range.

4.4 ABLATION STUDY

Table 2 evaluates the intent loss, intent mask, intent modulation, and Recovery Gate Head. Component-removal and always-on variants are retrained, while $c_t = 0$ disables the predicted intent at test time. Each entry averages 80 rollouts across four tasks for each evaluation regime. The direct human recovery mix provides the matched-data reference from Figure 4. Table 3 compares intent targets.

The intent loss and the intent mask both matter. Removing the intent loss reduces average Recovery SR from 85.0 to 65.0, and removing the intent mask reduces it to 55.0, lower than removing the intent loss altogether.

Table 1: **Collection efficiency audit.** Accepted demonstrations per operator hour, counting setup, resets, synchronization, and discarded attempts.

Task	Robot success/hr	Robot recovery/hr	Human success/hr	Human recovery/hr	Human/robot recovery
Cup brush insertion	58	48	483	522	10.9×
Table sweep	68	51	514	523	10.3×
Round disk placement	62	45	522	511	11.4×
Cube stack	60	52	498	510	9.8×
Average	62.0	49.0	504.3	516.5	10.5×

Table 2: **Closed-loop ablation.** Four-task average SR (%).

Configuration	Initial SR	Recovery SR
Direct human recovery mix	71.2	71.2
No intent loss	78.7	65.0
No intent mask	75.0	55.0
No intent modulation	76.3	65.0
$c_t = 0$ at test time	80.0	65.0
Always-on modulation	70.0	70.0
Full EgoRecovery	**80.0**	**85.0**

Table 3: **Intent target comparison.** SR (%) pooled over cup brush insertion and round disk placement. Each row retrains the model with another intent target.

Intent target	Initial SR	Recovery SR
Current and future observation latent	57.5	55.0
Learned future-trajectory latent	70.0	67.5
Raw future end-effector motion	72.5	67.5
Low-frequency DCT magnitude (ours)	**80.0**	**82.5**

Gated intent modulation connects the learned intent to robot control. Removing intent modulation during retraining lowers average Recovery SR to 65.0, below the 71.2 of the direct mix, so the intent loss does not help when the prediction never reaches the decoder. Setting $c_t = 0$ at test time produces the same 20-point drop from the full model, and this agreement indicates that the intent signal improves recovery through its modulation of the robot decoder.

The Recovery Gate Head improves both nominal and recovery execution. Applying corrective intent throughout training and evaluation lowers average Initial/Recovery SR from 80.0/85.0 to 70.0/70.0, close to the direct mix (71.2/71.2). On the trained EGORECOVERY checkpoint, fixed or shuffled gate values lower Recovery SR on two tasks from 82.5 to 35.0–45.0 (Appendix Table 14).

The DCT magnitude target gives the highest recovery success. The DCT magnitudes give the highest Initial and Recovery SR in Table 3 (per-task counts in Appendix L.1), and the observation latent reaches 55.0 Recovery SR, which equals robot-only recovery on these tasks. The ordering follows the embodiment-specific detail each target retains: the DCT target keeps only the magnitude and timing of the correction, raw motion and trajectory latents add direction and contact, and the observation latent further adds the appearance of hands and grippers. The fixed basis also needs no trained encoder, and the scale normalization in Eq. 1 removes embodiment-specific units.

5 CONCLUSION

We presented EGORECOVERY, a framework for recovery learning that uses egocentric human recovery as scalable supervision for failure states without treating human motion as robot action. The method transfers only a compact corrective intent, grounds it with limited robot recovery data, and activates it through a learned recovery gate at deployment. Across four real-robot tasks, this design improves task success from failure states over robot-only recovery and direct co-training with human recovery data, while preserving nominal execution.

Limitations. These findings are limited to related failure states within the same task-level failure family in tabletop manipulation. Human recovery reduces the robot recovery data burden, but does not replace robot recovery grounding for embodiment-specific contact handling, regrasping, or qualitatively new recovery skills. Contact-rich recovery may further require tactile sensing during collection.

AI USE STATEMENT

In this work, we used generative AI tools to clean and reformat data during data processing and filtering, and to provide feedback on the design of experiments, which the authors adopted only in part. We also used generative AI tools to search for related work, and the authors read every cited paper. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, propose or refine hypotheses, implement methods, assist with translation, support qualitative and thematic data analysis, or interpret results; formulating mathematical claims, proving mathematical claims, and writing proofs are not applicable to this work. Additionally, we used generative AI tools to draft parts of the paper and to edit it for grammar, spelling, word choice, and readability. We have reviewed all AI-assisted work: the authors inspected all processed data, evaluated all AI feedback on experimental design and made every final design decision, revised all AI-assisted text, and verified every bibliographic entry against its original arXiv record or official proceedings. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The main paper and appendix document the model, objectives, data composition, robot platforms, training details, success criteria, evaluation protocol, and per-task results, with additional implementation details and ablations for reproducibility. Appendix B describes the robot platforms, Appendices C–E the evaluation protocol, success criteria, and failure states, Appendices F–I the data collection, preprocessing, intent target, and training details, and Appendices J–L the per-task statistics, the object and background variation study, and the additional ablations. The supplementary material contains the training code and videos of data collection and closed-loop rollouts.

REFERENCES

Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022. doi: 10.15607/RSS.2022.XVIII.026.

Hongzhe Bi, Lingxuan Wu, Tianwei Lin, Hengkai Tan, Zhizhong Su, Hang Su, and Jun Zhu. H-RDT: Human manipulation enhanced bimanual robotic manipulation. *arXiv preprint arXiv:2507.23523*, 2025. doi: 10.48550/arXiv.2507.23523.

Xiongyi Cai, Ri-Zhao Qiu, Geng Chen, Lai Wei, Isabella Liu, Tianshu Huang, Xuxin Cheng, and Xiaolong Wang. In-N-On: Scaling egocentric manipulation with in-the-wild and on-task data. *arXiv preprint arXiv:2511.15704*, 2025. doi: 10.48550/arXiv.2511.15704.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin C. M. Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal Manipulation Interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.045.

Weisheng Dai, Kai Lan, Jianyi Zhou, Bo Zhao, Xiu Su, Junwen Tong, Weili Guan, and Shuo Yang. ConLA: Contrastive latent action learning from human videos for robotic manipulation. *arXiv preprint arXiv:2602.00557*, 2026. doi: 10.48550/arXiv.2602.00557.

Yicheng Feng, Wanpeng Zhang, Ye Wang, Hao Luo, Haoqi Yuan, Sipeng Zheng, and Zongqing Lu. Spatial-aware VLA pretraining through visual-physical alignment from human videos. *arXiv preprint arXiv:2512.13080*, 2025. doi: 10.48550/arXiv.2512.13080.

Harry Freeman, Chung Hee Kim, and George Kantor. WARPED: Wrist-aligned rendering for robot policy learning from egocentric human demonstrations. *arXiv preprint arXiv:2604.10809*, 2026. doi: 10.48550/arXiv.2604.10809.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Qianli Ma, Seungjun Nah, Loic Magne, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, You Liang Tan, K. R. Zentner, George Kurian, Suneel Indupuru, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi "Jim" Fan. DreamDojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026. doi: 10.48550/arXiv.2602.06949.

Manish Kumar Govind, Dominick Reilly, Pu Wang, and Srijan Das. UniLACT: Depth-aware RGB latent action learning for vision-language-action models. *arXiv preprint arXiv:2602.20231*, 2026. doi: 10.48550/arXiv.2602.20231.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18995–19012, 2022. doi: 10.1109/CVPR52688.2022.01842.

Yifan Han, Jianxiang Liu, Haoyu Zhang, Yuqi Gu, Yunhan Guo, and Wenzhao Lian. BridgeACT: Bridging human demonstrations to robot actions via unified tool-target affordances. *arXiv preprint arXiv:2604.23249*, 2026. doi: 10.48550/arXiv.2604.23249.

Marvin Heidinger, Snehal Jauhri, Vignesh Prasad, and Georgia Chalvatzaki. 2HandedAfforder: Learning precise actionable bimanual affordances from human videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14743–14753, 2025. doi: 10.1109/iccv51701.2025.01368.

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. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, volume 164 of *Proceedings of Machine Learning Research*, pp. 598–608. PMLR, 2022.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025. doi: 10.48550/arXiv.2505.11709.

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. *arXiv preprint arXiv:2509.07953*, 2025. doi: 10.48550/arXiv.2509.07953.

Renming Huang, Chendong Zeng, Wenjing Tang, Jintian Cai, Cewu Lu, and Panpan Cai. Mimic intent, not just trajectories. *arXiv preprint arXiv:2602.08602*, 2026. doi: 10.48550/arXiv.2602.08602.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13226–13233. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11127989.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. HG-DAgger: Interactive imitation learning with human experts. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083. IEEE, 2019. doi: 10.1109/ICRA.2019.8793698.

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. UniSkill: Imitating human videos via cross-embodiment skill representations. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4269–4294. PMLR, 2025.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. DART: Noise injection for robust imitation learning. In *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, volume 78 of *Proceedings of Machine Learning Research*, pp. 143–156. PMLR, 2017.

Jung Min Lee, Dohyeok Lee, Seokhun Ju, Taehyun Cho, Jin Woo Koo, Li Zhao, Sangwoo Hong, and Jungwoo Lee. MVP-LAM: Learning action-centric latent action via cross-viewpoint reconstruction. *arXiv preprint arXiv:2602.03668*, 2026. doi: 10.48550/arXiv.2602.03668.

Yu Lei, Minghuan Liu, Abhiram Maddukuri, Zhenyu Jiang, and Yuke Zhu. A mechanistic analysis of sim-and-real co-training in generative robot policies. *arXiv preprint arXiv:2604.13645*, 2026. doi: 10.48550/arXiv.2604.13645.

Chengyang Li, Kaiyi Xiong, Yuan Xu, Lei Qian, Yizhou Wang, and Wentao Zhu. GazeVLA: Learning human intention for robotic manipulation. *arXiv preprint arXiv:2604.22615*, 2026a. doi: 10.48550/arXiv.2604.22615.

Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. ManipTrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6991–7003, 2025. doi: 10.1109/CVPR52734.2025.00656.

Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, Yizhong Zhang, Xi Chen, Hao Chen, Lily Sun, Dong Chen, Jiaolong Yang, and Baining Guo. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026b.

Yaxuan Li, Zhongyi Zhou, Yefei Chen, Yanjiang Guo, Jiaming Liu, Shanghang Zhang, Jianyu Chen, and Yichen Zhu. Hi-WM: Human-in-the-world-model for scalable robot post-training. *arXiv preprint arXiv:2604.21741*, 2026c. doi: 10.48550/arXiv.2604.21741.

Yihang Li, Xuelong Wei, Jingzhou Luo, Yingjing Xiao, Yibo Bai, Guangyuan Zhou, Teng Zou, Chenguang Gui, Jiajun Wen, He Zhang, Kangliang Chen, Xing Pan, Shuaiyan Liu, Daming Wang, Tao An, Jiayi Li, Shibo Jin, Wanwan Zhang, Tianyu Wang, Boren Wei, Zhixuan Huang, Fangsheng Liu, Ruodai Li, Hui Zhang, Anson Li, Yicheng Gong, Peng Cao, Jiaming Liang, and Liang Lin. EgoLive: A large-scale egocentric dataset from real-world human tasks. *arXiv preprint arXiv:2604.23570*, 2026d. doi: 10.48550/arXiv.2604.23570.

Fanqi Lin, Kushal Arora, Jean Mercat, Haruki Nishimura, Paarth Shah, Chen Xu, Mengchao Zhang, Mark Zolotas, Maya Angeles, Owen Pfannenstiehl, Andrew Beaulieu, and Jose Barreiros. A systematic study of data modalities and strategies for co-training large behavior models for robot manipulation. *arXiv preprint arXiv:2602.01067*, 2026. doi: 10.48550/arXiv.2602.01067.

Huihan Liu, Shivin Dass, Roberto Martín-Martín, and Yuke Zhu. Model-based runtime monitoring with interactive imitation learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4154–4161. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611038.

Vincent Liu, Ademi Adeniji, Haotian Zhan, Siddhant Haldar, Raunaq Bhirangi, Pieter Abbeel, and Lerrel Pinto. EgoZero: Robot learning from smart glasses. *arXiv preprint arXiv:2505.20290*, 2025. doi: 10.48550/arXiv.2505.20290.

Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing robot experiences for failure explanation and correction. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3468–3484. PMLR, 2023.

Hao Luo, Ye Wang, Wanpeng Zhang, Haoqi Yuan, Yicheng Feng, Haiweng Xu, Sipeng Zheng, and Zongqing Lu. Joint-aligned latent action: Towards scalable VLA pretraining in the wild. *arXiv preprint arXiv:2602.21736*, 2026a. doi: 10.48550/arXiv.2602.21736.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, Yicheng Feng, and Zongqing Lu. Being-H0.5: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026b. doi: 10.48550/arXiv.2601.12993.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Proceedings of the 3rd Conference on Robot Learning (CoRL)*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1113–1132. PMLR, 2020.

Junyi Ma, Erhang Zhang, Haoran Yang, Ditao Li, Chenyang Xu, Guangming Wang, and Hesheng Wang. Robot learning from human videos: A survey. *arXiv preprint arXiv:2604.27621*, 2026. doi: 10.48550/arXiv.2604.27621.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 2022.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1820–1864. PMLR, 2023.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pp. 892–909. PMLR, 2023.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Mingjie Pan, Siyuan Feng, Qinglin Zhang, Xinchen Li, Jianheng Song, Chendi Qu, Yi Wang, Chuankang Li, Ziyu Xiong, Zhi Chen, Yi Liu, and Jianlan Luo. SOP: A scalable online post-training system for vision-language-action models. *arXiv preprint arXiv:2601.03044*, 2026. doi: 10.48550/arXiv.2601.03044.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 3942–3951, 2018. doi: 10.1609/aaai.v32i1.11671.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. doi: 10.15607/RSS.2025.XXI.012.

Ryan Punamiya, Dhruv Patel, Patcharapong Aphiwetsa, Pranav Kuppili, Lawrence Y. Zhu, Simar Kareer, Judy Hoffman, and Danfei Xu. EgoBridge: Domain adaptation for generalizable imitation from egocentric human data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 52583–52613, 2025. doi: 10.52202/085713-1569.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y. Zhu, Patcharapong Aphiwetsa, Baoyu Li, Aniketh Cheluva, Pranav Kuppili, Yangcen Liu, Dhruv Patel, Aidan Gao, Hye-Young Chung, Ryan Co, Renee Zbizika, Jeff Liu, Xiaomeng Xu, Haoyu Xiong, Geng Chen, Sebastiano Oliani, Wenkai Xuan, Chenyu Yang, Xi Wang, James Fort, Richard Newcombe, Josh Gao, Jason Chong, Garrett Matsuda, Aseem Doriwala, Marc Pollefeys, Robert Katzschmann, Xiaolong Wang, Shuran Song, Judy Hoffman, and Danfei Xu. EgoVerse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026. doi: 10.48550/arXiv.2604.07607.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

Jiaqi Shi, Xulong Zhang, Xiaoyang Qu, and Jianzong Wang. CARE: Multi-task pretraining for latent continuous action representation in robot control. *arXiv preprint arXiv:2601.22467*, 2026. doi: 10.48550/arXiv.2601.22467.

Junyao Shi, Zhuolun Zhao, Tianyou Wang, Ian Pedroza, Amy Luo, Jie Wang, Jason Ma, and Dinesh Jayaraman. ZeroMimic: Distilling robotic manipulation skills from web videos. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16939–16947. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11128283.

Himanshu Gaurav Singh, Antonio Loquercio, Carmelo Sferrazza, Jane Wu, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Hand-object interaction pretraining from videos. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3352–3360. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11127811.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026. doi: 10.48550/arXiv.2602.10098.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. MimicPlay: Long-horizon imitation learning by watching human play. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 201–221. PMLR, 2023.

James Wang, Primo Pu, Zephyr Fung, Alex Wang, Sam Wang, Bender Deng, Kevin Wang, Zivid Liu, Chris Pan, Panda Yang, Andy Zhai, Lucy Liang, Shalfun Li, Johnny Sun, Jacky Xu, Will Tian, Kai Yan, Kohler Ye, Scott Li, Qian Wang, Roy Gan, and Hao Wang. XRZero-G0: Pushing the frontier of dexterous robotic manipulation with interfaces, quality and ratios. *arXiv preprint arXiv:2604.13001*, 2026. doi: 10.48550/arXiv.2604.13001.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 124420–124450, 2024. doi: 10.52202/079017-3952.

Yifei Wei, Linqing Zhong, Yi Liu, Yuxiang Lu, Xindong He, Maoqing Yao, and Guanghui Ren. Libra-VLA: Achieving learning equilibrium via asynchronous coarse-to-fine dual-system. *arXiv preprint arXiv:2604.24921*, 2026. doi: 10.48550/arXiv.2604.24921.

Edgar Welte, Yitian Shi, Rosa Wolf, Maximillian Gilles, and Rania Rayyes. FlowCorrect: Efficient interactive correction of generative flow policies for robotic manipulation. *arXiv preprint arXiv:2602.22056*, 2026. doi: 10.48550/arXiv.2602.22056.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.

Yifan Xie, YuAn Wang, Guangyu Chen, Jinkun Liu, Yu Sun, and Wenbo Ding. Learning human-intention priors from large-scale human demonstrations for robotic manipulation. *arXiv preprint arXiv:2604.24681*, 2026. doi: 10.48550/arXiv.2604.24681.

Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. XSkill: Cross embodiment skill discovery. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3536–3555. PMLR, 2023.

Yifu Xu, Bokai Lin, Xinyu Zhan, Hongjie Fang, Yong-Lu Li, Cewu Lu, and Lixin Yang. LIDEA: Human-to-robot imitation learning via implicit feature distillation and explicit geometry alignment. *arXiv preprint arXiv:2604.10677*, 2026. doi: 10.48550/arXiv.2604.10677.

Bowen Yang, Zishuo Li, Yang Sun, Changtao Miao, Yifan Yang, Man Luo, Xiaotong Yan, Feng Jiang, Jinchuan Shi, Yankai Fu, Ning Chen, Junkai Zhao, Pengwei Wang, Guocai Yao, Shanghang Zhang, Hao Chen, Zhe Li, and Kai Zhu. AoE: Always-on egocentric human video collection for embodied AI. *arXiv preprint arXiv:2602.23893*, 2026. doi: 10.48550/arXiv.2602.23893.

Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, Hongxu Yin, Sifei Liu, Song Han, Yao Lu, and Xiaolong Wang. EgoVLA: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025. doi: 10.48550/arXiv.2507.12440.

Anlan Yu, Zaishu Chen, Peili Song, Zhiqing Hong, Haotian Wang, Desheng Zhang, Tian He, Yi Ding, and Daqing Zhang. WM-DAgger: Enabling efficient data aggregation for imitation learning with world models. *arXiv preprint arXiv:2604.11351*, 2026. doi: 10.48550/arXiv.2604.11351.

Chubin Zhang, Jianan Wang, Zifeng Gao, Yue Su, Tianru Dai, Cai Zhou, Jiwen Lu, and Yansong Tang. CLAP: Contrastive latent action pretraining for learning vision-language-action models from human videos. *arXiv preprint arXiv:2601.04061*, 2026a. doi: 10.48550/arXiv.2601.04061.

Gu Zhang, Qicheng Xu, Haozhe Zhang, Jianhan Ma, Long He, Yiming Bao, Zeyu Ping, Zhecheng Yuan, Chenhao Lu, Chengbo Yuan, Tianhai Liang, Xiaoyu Tian, Maanping Shao, Feihong Zhang, Mingyu Ding, Yang Gao, Hao Zhao, Hang Zhao, and Huazhe Xu. UniDex: A robot foundation suite for universal dexterous hand control from egocentric human videos. *arXiv preprint arXiv:2603.22264*, 2026b. doi: 10.48550/arXiv.2603.22264.

Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. HaWoR: World-space hand motion reconstruction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1805–1815, 2025. doi: 10.1109/CVPR52734.2025.00175.

Tao Zhang, Song Xia, Ye Wang, and Qin Jin. EasyMimic: A low-cost framework for robot imitation learning from human videos. *arXiv preprint arXiv:2602.11464*, 2026c. doi: 10.48550/arXiv.2602.11464.

Gehan Zheng, Sanjay Seenivasan, Matthew Johnson-Roberson, and Weiming Zhi. Rewind-IL: Online failure detection and state respawning for imitation learning. *arXiv preprint arXiv:2604.16683*, 2026a. doi: 10.48550/arXiv.2604.16683.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, Trevor Darrell, Furong Huang, Yuke Zhu, Danfei Xu, and Linxi Fan. EgoScale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026b. doi: 10.48550/arXiv.2602.16710.

Yiming Zhong, Yaoyu He, Zemin Yang, Pengfei Tian, Yifan Huang, Qingqiu Huang, Xinge Zhu, and Yuexin Ma. From noise to intent: Anchoring generative VLA policies with residual bridges. *arXiv preprint arXiv:2604.21391*, 2026. doi: 10.48550/arXiv.2604.21391.

Lawrence Y. Zhu, Pranav Kuppili, Ryan Punamiya, Patcharapong Aphiwetsa, Dhruv Patel, Simar Kareer, Sehoon Ha, and Danfei Xu. EMMA: Scaling mobile manipulation via egocentric human data. *IEEE Robotics and Automation Letters*, 11(3):3087–3094, 2026. doi: 10.1109/LRA.2026.3653320.

Yanwen Zou, Chenyang Shi, Wenye Yu, Han Xue, Jun Lv, Ye Pan, Chuan Wen, and Cewu Lu. ActiveGlasses: Learning manipulation with active vision from ego-centric human demonstration. *arXiv preprint arXiv:2604.08534*, 2026. doi: 10.48550/arXiv.2604.08534.

Appendix

A Data Sources and Budgets

All episode counts in the experiments are per task and source specific. They give the number of episodes that an experimental condition draws from each source, and the collected pool of each source can be larger. The full EgoRecovery setting uses Robot success (50), Robot recovery (50), and Human recovery (300). Human success (300) is used only as the nominal human control in the main comparison. The human data scaling study increases the human budget to 800 while holding Robot success (50) and Robot recovery (50) fixed. The budget comparison additionally uses Robot recovery (80) as a robot only reference and Robot recovery (20) as the low robot grounding setting.

B Robot Platforms

Figure 3 shows the two robot platforms, and Table 4 summarizes their configuration. Cup brush insertion, table sweep, and round disk placement run on an ALOHA AgileX dual-arm system, and cube stacking runs on a single Franka Panda arm. ALOHA demonstrations are teleoperated with two leader arms that the follower arms mirror, and Franka demonstrations are teleoperated with a SpaceMouse. Both platforms use Intel RealSense D435 cameras. The policy receives the RGB images of all cameras on its platform, resized to 240×320, together with the proprioceptive robot state. The human recording rig in Appendix F mirrors the ALOHA camera layout: the head-mounted camera takes the role of the front camera, and one camera on each wrist matches the robot wrist views. The cube stacking human data use a single hand, matching the single Franka arm.

Each task trains its own policy. At deployment, the policy predicts a 100-step action chunk, the robot executes its first 50 steps, and the policy is then queried again with new observations. At each chunk boundary, the first command is linearly interpolated from the measured robot state. Inference runs on a single NVIDIA RTX 4090 GPU.

Table 4: Robot platforms used for data collection and closed-loop evaluation.

	ALOHA AgileX	Franka Panda
Tasks	Cup brush insertion, table sweep, round disk placement	Cube stacking
Arms	Two 6-DoF arms with parallel grippers	One 7-DoF arm
Teleoperation	Leader–follower arms	SpaceMouse
Cameras	Front and left/right wrist, RealSense D435	Third-person and wrist, RealSense D435
Action space	14D absolute joint positions	Absolute end-effector pose and gripper

C Closed-Loop Evaluation Protocol

Each rollout is run for at most 5 minutes. A rollout that does not complete the task within this window is counted as a failure. Policies may retry task actions such as reinserting the brush, sweeping residual objects, or restacking a fallen cube, provided that the robot remains engaged with the same task instance. A rollout is terminated if the policy executes a reset action that returns the robot to the initial pose or reinitializes the task, so recovery cannot be achieved by erasing the failed attempt and restarting from scratch. For each method, we use the same fixed set of 20 standard initial states and 20 failure states per task. In addition to success rates, we report the corresponding success counts. Wilson 95% binomial confidence intervals (Wilson, 1927) are computed from the pooled success counts over the 80 rollouts that form each four task aggregate.

D Task Success Criteria

All task outcomes are scored by a human evaluator from the synchronized rollout videos. The evaluator applies the rules below to the video alone; recovery labels and corrective intent targets are

training signals and do not enter scoring. A rollout is counted as successful only if the task specific goal state is reached before the timeout and remains visually stable for at least 3 s after the final corrective action. Multiple attempts are allowed within the same rollout unless a task specific terminal failure occurs or the policy executes a reset action as defined in Appendix C.

Cup brush insertion. The brush is successful if the brush tip and shaft are visibly inserted into the tube, the tube remains upright, and the brush does not pop out or tilt out of the tube during the 3 s stability window. Multiple insertion attempts are allowed before timeout. The rollout fails if the tube is knocked over, the brush is dropped outside the reachable workspace, or the robot resets the task.

Table sweep. The task is successful if all predefined trash objects are inside the dustpan before pouring and are then transferred into the bin. Multiple sweeping passes are allowed before pouring, but only one pouring action from the dustpan to the bin is permitted. The rollout fails if any predefined trash object remains on the table after pouring, if trash is spilled outside the bin during pouring, or if the robot resets the task.

Round disk placement. The disk is successful if it is seated flush inside the target groove, with no visible part of the disk protruding above the rim or resting on the rim during the 3 s stability window. Multiple placement attempts are allowed before timeout. The rollout fails if the disk is dropped outside the reachable workspace, the target fixture is displaced, or the robot resets the task.

Cube stacking. The task is successful if the yellow cube is placed on top of the green cube and the stack remains upright for the 3 s stability window. If the yellow cube falls before the final stable state, the robot may retry stacking within the same rollout. The rollout fails if the green cube is displaced from its original position, either cube leaves the reachable workspace, or the robot resets the task.

E Failure States for Recovery Evaluation

Recovery SR evaluates policies from failure states. These failure states define the recovery distribution used for closed-loop evaluation. Robot and human recovery training episodes are collected within the same task-level failure families, and the evaluation failure states are held-out failure configurations from those families. The benchmark therefore tests transfer to new failure configurations of the trained failure families. This section specifies the failed task state and the robot arm state at the beginning of recovery evaluation. The recovery rollout uses only robot observations. No external failure detector, human video, or recovery label is provided at test time.

Cup brush insertion. The failed task state is that the brush has not been inserted into the tube after an insertion attempt. Recovery starts with the left and right arms holding the brush and tube, respectively, in a post-attempt state where insertion has failed.

Table sweep. The failed task state is that the left-arm sweep has not covered all trash objects, so trash remains on the table before the dustpan is poured. Recovery starts with the left arm holding the broom and the right arm holding the dustpan, after a partial sweep into the dustpan but before pouring.

Round disk placement. The failed task state is that the round disk has not been inserted into the groove because it is misaligned with the target groove. Recovery starts with the right arm holding the disk handle in a post-placement-attempt state where the disk is near the groove but not seated.

Cube stacking. The failed task state is that the yellow cube has not been stacked on the green cube after the stacking attempt. Recovery starts with the robot arm hovering above the green cube after the failed placement attempt, while the yellow cube remains grasped or within the reachable correction region.

F Data Collection and Annotation

This section describes the data collection and labeling process used for the throughput audit in Table 1 and for the recovery labels in Section 3.1.

Recording setup. Human recordings use synchronized RGB-D streams at 25 Hz. A head mounted egocentric camera records the task scene and the demonstrator's first person view. Two wrist mounted

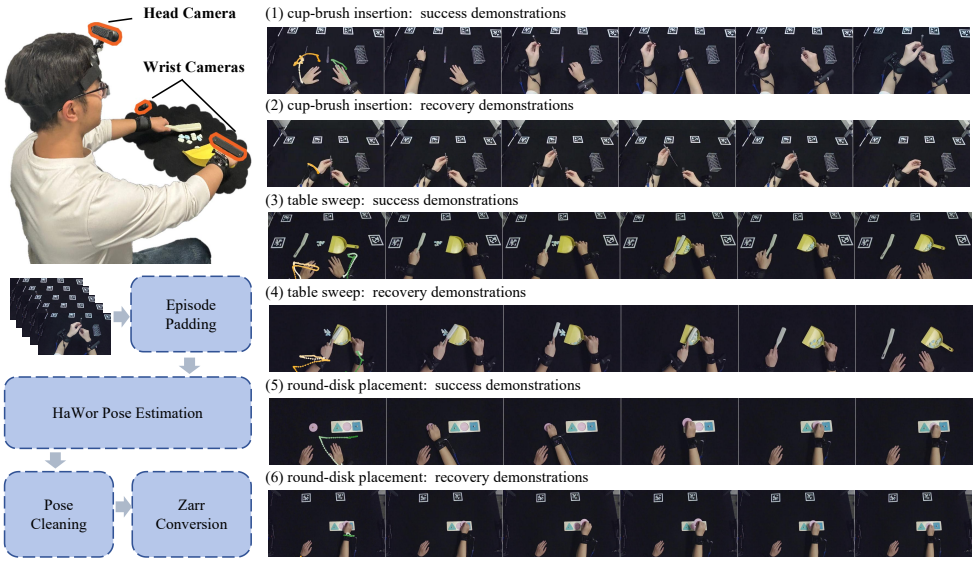


Figure 7: **Egocentric human recovery collection setup.** Head mounted first person recording is used to collect human recovery demonstrations from staged failure states, with wrist mounted streams recording close range hand object interaction. These videos provide failure state and corrective intent supervision during training only; policies are evaluated using robot observations only.

RGB-D cameras record close range hand object interaction for the left and right hands. The current implementation uses Orbbec RGB-D cameras, storing RGB frames as compressed images, depth frames as millimeter scale uint16 maps, and intrinsics and timestamps for each episode. This head mounted plus wrist mounted configuration follows recent always-on egocentric collection rigs (Yang et al., 2026). Other rigs additionally record gaze or actively controlled viewpoints (Zou et al., 2026; Li et al., 2026a); the present setup uses fixed head and wrist cameras only, so no gaze or viewpoint control signal enters the corrective intent target.

Table 5: Human recording streams used for egocentric recovery collection. The head mounted stream captures scene context and task progress, while the wrist mounted streams capture close range hand object interaction. All streams store RGB, intrinsics, and timestamps for synchronization.

Stream	Sensor position	Main use	Stored data
Ego RGB	Head mounted	Scene context and task progress	RGB, intrinsics, timestamps
Left wrist RGB	Left wrist	Hand object contact	RGB, intrinsics, timestamps
Right wrist RGB	Right wrist	Hand object contact	RGB, intrinsics, timestamps

Collection protocol. When used as a nominal human control, human success demonstrations start from standard initial states and run to task completion. Human and robot recovery episodes use the task level failure families in Appendix E, with varied failure configurations inside each family. Human recovery clips contain only the corrective segment after a staged failure state. For the throughput accounting in Table 1, setup time, resets, synchronization, and discarded attempts are counted for both human and robot collection.

Episode filtering. An episode is accepted if it begins from a valid failure state, contains a complete corrective maneuver, and has synchronized observations throughout the active phase. Human episodes are additionally filtered for reliable bilateral hand tracking and the absence of severe occlusion or sensor dropout. Robot episodes are filtered for synchronized robot state and action logs, no major teleoperation interruption, and no severe sensor dropout. Discarded episodes are not used for training but remain part of the operator hour accounting.

Recovery segment annotation. Each accepted recovery clip starts at a failure state, so the annotator marks only one boundary, t_{rec}, where corrective motion ends and ordinary task execution resumes. We obtain this boundary in two steps. First, we smooth the hand or end effector motion and compute a framewise motion energy curve, which gives an automatic candidate for t_{rec}. The annotator then checks the synchronized video together with the curve and either confirms the candidate or moves it to the frame where corrective motion has returned to nominal execution. The same pass records a quality flag and whether the episode should be discarded.

For accepted episodes, t_{rec} is converted into the labels summarized in Table 6. The recovery state label s_t marks the recovery phase for the Recovery Gate Head. The intent mask m_t is a stricter training mask whose validity checks are detailed in Appendix H. These labels are used only during training and are not available at deployment.

Table 6: Recovery annotation schema used to convert each accepted recovery episode into training labels. The episode boundary defines the recovery phase, while the derived frame labels supervise the Recovery Gate Head and the corrective intent loss.

Field	Level	Use
t_{rec}	Episode	Boundary between recovery and normal execution
`quality`, `discard`	Episode	Filtering and auditing
s_t	Frame	Target for the Recovery Gate Head
m_t	Frame or segment	Mask for corrective intent supervision

G Motion Reconstruction and Data Preprocessing

Hand trajectory reconstruction. We estimate bilateral 3D hand motion from the egocentric video using HaWoR (Zhang et al., 2025). From the reconstructed MANO hands we extract the wrist joint position and root orientation. The orientation is converted to roll-pitch-yaw angles, giving a 6-DoF wrist pose $[x, y, z, r_x, r_y, r_z]$ in meters and radians. Together with the gripper aperture proxy below, each hand is represented as $[x, y, z, r_x, r_y, r_z, \bar{g}]$, giving a 14D bilateral human state stream, or a 7D stream for the single-hand cube stacking data. This reconstructed pose stream is used as the human embodiment trajectory for the auxiliary human decoder and for corrective intent target construction. It is not used as robot frame action supervision. Other systems retarget reconstructed hands onto robot or dexterous hand kinematics to obtain direct action labels (Li et al., 2025; Zhang et al., 2026b), or re-render the egocentric view into a wrist aligned robot view (Freeman et al., 2026). Both approaches commit to an embodiment specific mapping, which the corrective intent target avoids.

We compute a gripper-aperture proxy from fingertip distance:

$$g_t = \left\| p^t_{\text{thumb}} - p^t_{\text{index}} \right\|_2, \tag{4}$$

where p^t_{thumb} and p^t_{index} are the MANO fingertip positions. This distance is linearly mapped to $\bar{g}_t \in [0, 1]$ over $[0.03\,\text{m}, 0.08\,\text{m}]$, with clipping.

Filtering and resampling. Frames with wrist position jumps above 30 mm or Euler-angle jumps above 20° are replaced by interpolation from neighboring valid frames. We then apply median filtering and Savitzky–Golay smoothing; rotation channels are unwrapped before smoothing and re-wrapped afterwards. Continuous state vectors are exported to the 25 Hz trajectory format used by the training zarrs through linear interpolation; video frames are selected by nearest neighbor. Segments that remain unreliable after filtering are rejected or masked out before intent supervision.

H Corrective Intent Target

Target construction. The intent loss uses the target only where the intent mask m_t is true. We first define target construction and then define the validity mask. Following Eq. 1, we compute the DCT magnitude target from the scaled relative future motion of each active end effector. This appendix specifies the concrete windowing and masking choices used in the experiments. Raw

trajectories are stored at 25 Hz, but the intent target is computed on a phase resampled grid rather than on a fixed duration window. Each episode is linearly resampled to $P = 200$ phase points. For an anchor frame t, we map the frame to its phase index p_t and take a future window of $L = 16$ phase points, which covers 8% of that episode. Thus the window duration varies with episode length. For example, it is about 0.4 s for a 5 s human recovery clip. The DCT magnitude target is a fixed analytic basis. Learned alternatives include latent action variables trained by cross-viewpoint or depth-aware reconstruction (Lee et al., 2026; Govind et al., 2026), multi-task continuous action representations (Shi et al., 2026), and frequency-domain action tokenizers (Pertsch et al., 2025). A fixed basis keeps the target independent of the policy being trained, so the intent loss does not move with the action decoder.

The main experiments use $K = 4$ DCT coefficients and one active end effector, resulting in a 4D target. The active effector is the task designated active arm, while the other arm holds a static grip pose when applicable. The same construction supports two active effectors, which would give an 8D target, but that two effector variant is not used in the main results.

Masking and normalization. The effective intent mask is the conjunction of two masks. The first mask, `gt_intent_valid`, requires the frame to be unpadded, the future window to fit inside the phase resampled episode, and the peak to peak ℓ_2 displacement of the window to be at least 0.01 m, which removes near stationary segments. The second mask, `recovery_intent_valid`, requires the anchor frame to lie inside the annotated recovery phase and the full future window to remain inside that phase. Success episodes set both masks to false. The loss uses their conjunction, so intent supervision is applied only to valid recovery windows.

Invalid segments are stored as masked targets during preprocessing and do not contribute to the intent loss or normalization statistics. Consistent with the main text target in Eq. 1, embodiment scale calibration is applied before the DCT through s_d. Normalization statistics are computed only from valid recovery windows, so padded frames, invalid windows, and success episode targets are excluded.

I Training Details

The training implementation follows the heterogeneous policy training setup used in EgoMimic style co training, with a shared visual and Transformer trunk and embodiment specific observation adapters and action decoders. The human decoder is trained only with reconstructed human hand trajectories, while the robot decoder is trained only with robot teleoperation actions. Thus human action labels help train the shared trunk and human decoder but are not used as robot action supervision.

Training uses Robot and Human dataloaders whose selected episode pools follow the budget of each experimental condition. For the main comparison, this means Robot success (50), Robot recovery (50), and Human recovery (300); Human success episodes are used only in the corresponding human success control. Frames are sampled uniformly within each selected pool, without an additional class balanced or phase balanced sampler. The direct mix baseline and mechanism ablations use the same sampling and optimization settings unless the ablation explicitly removes a data source or disables one model component. The co-training ratio and the choice of sharing a trunk across heterogeneous data follow reported practice for multi-embodiment robot datasets (Open X-Embodiment Collaboration, 2024; Lin et al., 2026; Lei et al., 2026). No synthetic demonstration augmentation is used (Mandlekar et al., 2023). Both action decoders are action chunking transformer heads; the implementation also supports a diffusion decoder (Chi et al., 2023), and all results in this paper use the transformer head.

Robot actions are 14D absolute joint position targets on ALOHA, with six arm joints and one gripper value per arm, and absolute end-effector pose and gripper targets on Franka (Appendix B). Human actions are absolute hand state targets, with x, y, z, r_x, r_y, r_z and a gripper proxy per hand. Both streams are stored at 25 Hz and are used to form 100 step action chunks for the corresponding decoder. We use the objective in Eq. 3 with the constants in Table 7.

Table 7: Compact training details for the main EGORECOVERY runs.

Model	
Policy	HPT co-training
Backbone	16 blocks, $d = 256$, 8 heads, drop path 0.1
Decoder	8 layers, 100 step horizon
Heads	4D intent, 1D gate, gated FiLM
Optimization	
Batch	24 per GPU, 192 global
Optimizer	AdamW, lr 10^{-4}, wd 10^{-4}
Scheduler	Cosine, $T_{\max} = 200$, min lr 2×10^{-6}
Loss	$\lambda_c = 0.05$, $\lambda_g = 0.05$, $\lambda_n = 0.01$
Input processing	
Augmentation	Color jitter, ImageNet norm

All main models are trained for 20,000 optimization steps with BF16 mixed precision on 8 NVIDIA L20 GPUs. Checkpoints are saved every 10,000 steps, and reported rollouts use the final 20,000 step checkpoint unless otherwise noted. The implementation uses dynamic median absolute deviation gradient clipping after a 100 step warmup.

For compute accounting, the primary paper-reported runs were trained on a single 8 GPU node with 48 GB NVIDIA L20 devices. A typical main co-training run took about 10 hours, or roughly 80 L20 GPU hours. The training runs reported in the paper required approximately 2000 L20 GPU hours in total, with exploratory, failed, or superseded runs during method development adding on the order of a few hundred additional L20 GPU hours. Each reported model was trained with a single seed. Peak memory was not logged per run, but all reported runs fit within the 48 GB per GPU memory budget under BF16 mixed precision.

J FULL CLOSED-LOOP STATISTICS FOR MAIN RESULTS

Following the closed loop protocol in Appendix C, Table 8 reports the pooled success counts and Wilson confidence intervals behind the main comparison. The task wise percentages are shown in Figure 4; this table provides the count level audit. Table 9 provides the same audit for the human data scaling study in Figure 5, while Tables 10 and 11 report the two regimes in Figure 6. All statistics in this section pool the four tasks used for the corresponding main text average. An entry such as $80.0\,[70.0, 87.3]$ reports the success rate in percent followed by the Wilson 95% binomial confidence interval.

These tables list the success counts behind the main figures and add no experimental settings. They show the same pattern as the main text: EGORECOVERY reaches its highest Recovery SR when human recovery is combined with limited robot recovery data, and direct co-training or replacing most robot recovery demonstrations with human ones lowers Recovery SR.

Table 8: Aggregate statistics for the main closed loop comparison. All methods are evaluated on the same fixed set of 20 initial states and 20 failure states per task. Counts aggregate the four tasks, giving 80 rollouts per start state regime. Intervals are Wilson 95% binomial confidence intervals. Bold values mark the best aggregate result.

Method	Robot success	Robot recovery	Human success	Human recovery	Initial successes	Initial SR (%, Wilson CI)	Recovery successes	Recovery SR (%, Wilson CI)
Robot success only	50	–	–	–	30/80	37.5 [27.7, 48.5]	0/80	0.0 [0.0, 4.6]
$R_{\text{suc}}(50) + H_{\text{suc}}(300)$	50	–	300	–	36/80	45.0 [34.6, 55.9]	0/80	0.0 [0.0, 4.6]
$R_{\text{suc}}(50) + H_{\text{rec}}(300)$	50	–	–	300	34/80	42.5 [32.3, 53.4]	7/80	8.8 [4.3, 17.0]
Robot recovery	50	50	–	–	45/80	56.3 [45.3, 66.6]	42/80	52.5 [41.7, 63.1]
Human success mix	50	50	300	–	51/80	63.8 [52.8, 73.4]	52/80	65.0 [54.1, 74.5]
Direct human recovery mix	50	50	–	300	57/80	71.2 [60.5, 80.0]	57/80	71.2 [60.5, 80.0]
EGORECOVERY	**50**	**50**	**–**	**300**	**64/80**	**80.0 [70.0, 87.3]**	**68/80**	**85.0 [75.6, 91.2]**

Table 9: Pooled statistics for the human data scaling study in Figure 5. All rows hold Robot success (50) and Robot recovery (50) fixed. Counts aggregate four tasks with 80 rollouts per start state regime. Bold values mark the better matched condition at each human data budget.

Method	Robot success	Robot recovery	Human success	Human recovery	Initial successes	Initial SR (%, Wilson CI)	Recovery successes	Recovery SR (%, Wilson CI)
Human success co training	50	50	200	–	44/80	55.0 [44.1, 65.4]	47/80	58.8 [47.8, 68.9]
EGORECOVERY	50	50	–	200	**59/80**	**73.8 [63.2, 82.1]**	**60/80**	**75.0 [64.5, 83.2]**
Human success co training	50	50	300	–	51/80	63.8 [52.8, 73.4]	52/80	65.0 [54.1, 74.5]
EGORECOVERY	50	50	–	300	**64/80**	**80.0 [70.0, 87.3]**	**68/80**	**85.0 [75.6, 91.2]**
Human success co training	50	50	500	–	56/80	70.0 [59.2, 78.9]	53/80	66.2 [55.4, 75.7]
EGORECOVERY	50	50	–	500	**67/80**	**83.8 [74.2, 90.3]**	**68/80**	**85.0 [75.6, 91.2]**
Human success co training	50	50	800	–	59/80	73.8 [63.2, 82.1]	55/80	68.8 [57.9, 77.8]
EGORECOVERY	50	50	–	800	**70/80**	**87.5 [78.5, 93.1]**	**72/80**	**90.0 [81.5, 94.8]**

Table 10: Pooled recovery statistics for the regime at matched collection effort in Figure 6(a). The robot-equivalent recovery budget is fixed at $B_{\text{eq}} = 80$. Counts aggregate four tasks with 80 recovery rollouts per row. Bold values mark the higher result for each matched recovery budget.

Method	Robot success	Robot recovery	Human success	Human recovery	B_{eq}	Recovery successes	Recovery SR (%, Wilson CI)
Robot only reference	50	80	–	–	80	48/80	60.0 [49.0, 70.0]
Direct human recovery mix	50	70	–	100	80	49/80	61.2 [50.3, 71.2]
EGORECOVERY	50	70	–	100	80	**56/80**	**70.0 [59.2, 78.9]**
Direct human recovery mix	50	60	–	200	80	52/80	65.0 [54.1, 74.5]
EGORECOVERY	50	60	–	200	80	**61/80**	**76.2 [65.9, 84.2]**
Direct human recovery mix	50	50	–	300	80	57/80	71.2 [60.5, 80.0]
EGORECOVERY	50	50	–	300	80	**68/80**	**85.0 [75.6, 91.2]**
Direct human recovery mix	50	40	–	400	80	53/80	66.2 [55.4, 75.7]
EGORECOVERY	50	40	–	400	80	**62/80**	**77.5 [67.2, 85.3]**
Direct human recovery mix	50	30	–	500	80	50/80	62.5 [51.5, 72.3]
EGORECOVERY	50	30	–	500	80	**58/80**	**72.5 [61.9, 81.1]**
Direct human recovery mix	50	20	–	600	80	50/80	62.5 [51.5, 72.3]
EGORECOVERY	50	20	–	600	80	**60/80**	**75.0 [64.5, 83.2]**

Table 11: Pooled recovery statistics for the low robot grounding regime in Figure 6(b). Robot recovery is fixed at Robot recovery (20), and human recovery is varied. Counts aggregate four tasks with 80 recovery rollouts per row. Bold values mark the higher result for each human recovery budget.

Method	Robot success	Robot recovery	Human success	Human recovery	B_{eq}	Recovery successes	Recovery SR (%, Wilson CI)
Robot only reference	50	20	–	–	20	28/80	35.0 [25.5, 45.9]
Direct human recovery mix	50	20	–	100	30	36/80	45.0 [34.6, 55.9]
EGORECOVERY	50	20	–	100	30	**42/80**	**52.5 [41.7, 63.1]**
Direct human recovery mix	50	20	–	200	40	42/80	52.5 [41.7, 63.1]
EGORECOVERY	50	20	–	200	40	**48/80**	**60.0 [49.0, 70.0]**
Direct human recovery mix	50	20	–	300	50	46/80	57.5 [46.6, 67.7]
EGORECOVERY	50	20	–	300	50	**53/80**	**66.2 [55.4, 75.7]**
Direct human recovery mix	50	20	–	400	60	48/80	60.0 [49.0, 70.0]
EGORECOVERY	50	20	–	400	60	**56/80**	**70.0 [59.2, 78.9]**
Direct human recovery mix	50	20	–	500	70	50/80	62.5 [51.5, 72.3]
EGORECOVERY	50	20	–	500	70	**58/80**	**72.5 [61.9, 81.1]**
Direct human recovery mix	50	20	–	600	80	50/80	62.5 [51.5, 72.3]
EGORECOVERY	50	20	–	600	80	**60/80**	**75.0 [64.5, 83.2]**

K Recovery under Object and Background Variation

We evaluate all training configurations on the same object-instance changes and background distractors for cup brush insertion and round disk placement. Table 12 reports Initial and Recovery SR under these variations.

Table 12: **Recovery under object and background variation.** Initial and Recovery SR (%, Wilson 95% CI); each entry uses 20 rollouts under the same evaluation variations.

Training data	Cup brush Initial SR	Cup brush Recovery SR	Round disk Initial SR	Round disk Recovery SR
Robot data only	20.0 [8.1, 41.6]	15.0 [5.2, 36.0]	40.0 [21.9, 61.3]	35.0 [18.1, 56.7]
+ Human recovery (original setting)	35.0 [18.1, 56.7]	40.0 [21.9, 61.3]	50.0 [29.9, 70.1]	55.0 [34.2, 74.2]
+ Human recovery (varied settings)	60.0 [38.7, 78.1]	55.0 [34.2, 74.2]	75.0 [53.1, 88.8]	60.0 [38.7, 78.1]

All configurations use the same 50 robot success and 50 robot recovery demonstrations collected in the original setting. The two configurations with human recovery use EgoRecovery. The second row adds 300 human recovery demonstrations from the original setting. The final row further adds 300 human recovery demonstrations collected across varied environments, object instances, and background clutter, without additional robot data. The additional varied-setting human recovery data raises Initial SR from 35.0 to 60.0 on cup brush and from 50.0 to 75.0 on round disk; Recovery SR rises from 40.0 to 55.0 and from 55.0 to 60.0, respectively.

Figure 8 shows the corresponding human recovery collection conditions and robot rollouts.

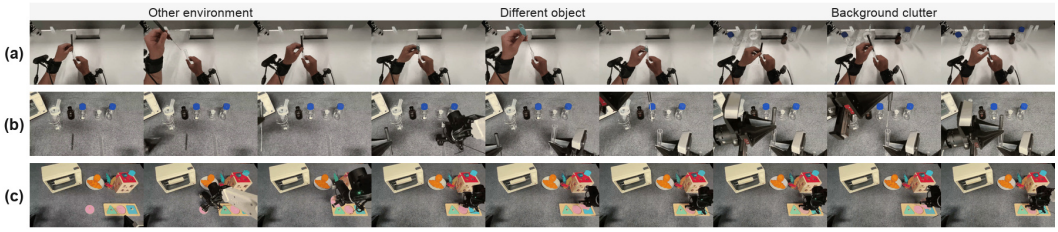


Figure 8: **Qualitative examples under object and background variation.** **(a)** Human recovery collected in another environment, with a different object instance, and with background clutter. **(b–c)** Robot rollouts for cup brush insertion and round disk placement under the evaluation variations.

L Additional Ablations and Mechanism Analysis

The closed-loop ablations in Table 2 test the intent loss, intent mask, gated modulation, and Recovery Gate Head. This section evaluates three alternative intent targets under the same protocol and then analyzes the learned intent and gate signals.

L.1 Intent Target Choice

Equation 1 defines the corrective intent target as the $K = 4$ low-frequency DCT magnitudes of the scale-normalized future end-effector motion. To test that choice we replace the target with three alternatives and retrain the full pipeline for each, holding the data composition (Robot success (50), Robot recovery (50), Human recovery (300)), the backbone, the Recovery Gate Head, and the gated modulation of Eq. 2 fixed, so that only the quantity the Intent Head is asked to predict changes. The raw future end-effector target keeps the same prediction window but drops the frequency basis and the scale normalization, leaving the target in embodiment-specific units. The learned future-trajectory latent replaces the fixed basis with an encoder over the same window trained jointly with the policy. The observation latent conditions on current and future observation features rather than on motion, so the target no longer describes a correction. Evaluation covers cup brush insertion and round disk

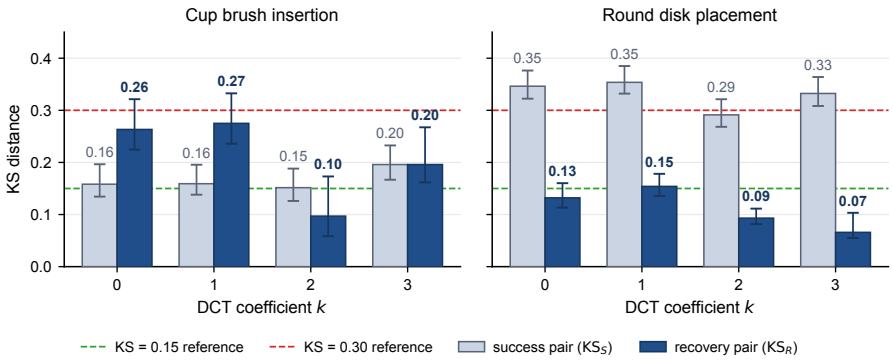


Figure 9: **Cross-embodiment alignment of the $K = 4$ DCT magnitude intent target.** Bars report two-sample KS distances between robot and human DCT target distributions for each coefficient. Light bars compare success trajectories and dark bars compare recovery sub-trajectories; lower values indicate stronger cross-embodiment overlap. Left and right panels show cup brush insertion and round disk placement, respectively.

placement under the closed loop protocol of Appendix C, with 20 initial states and 20 failure states per task.

Table 13: Corrective intent target ablation. Each row retrains the full pipeline with a different intent target and leaves the Recovery Gate Head, the gated modulation, the backbone, and the data composition unchanged. Per task cells report successes out of 20 rollouts. Pooled columns aggregate the two tasks, giving 40 rollouts per start state regime, with Wilson 95% binomial confidence intervals (Wilson, 1927). Bold values mark the best aggregate result.

Intent target	Cup brush Initial	Cup brush Recovery	Round disk Initial	Round disk Recovery	Pooled Initial SR (%, Wilson CI)	Pooled Recovery SR (%, Wilson CI)
Current and future observation latent	11/20	10/20	12/20	12/20	57.5 [42.2, 71.5]	55.0 [39.8, 69.3]
Learned future-trajectory latent	13/20	13/20	15/20	14/20	70.0 [54.6, 81.9]	67.5 [52.0, 79.9]
Raw future end-effector motion	14/20	13/20	15/20	14/20	72.5 [57.2, 83.9]	67.5 [52.0, 79.9]
Low-frequency DCT magnitude (ours)	**15/20**	**16/20**	**17/20**	**17/20**	**80.0 [65.2, 89.5]**	**82.5 [68.1, 91.3]**

The DCT target gives the highest pooled Initial and Recovery SR, 80.0 and 82.5. The raw future motion target and the learned future-trajectory latent both reach 67.5 Recovery SR, and the observation latent reaches 55.0, the rate of robot-only recovery on these two tasks. The ordering follows the embodiment-specific detail each target retains. The DCT magnitudes keep the magnitude and timing of the correction and discard its direction. The raw target and the trajectory latent keep direction and the small contact motions that differ between hands and grippers, and the observation latent additionally encodes their appearance. The Wilson intervals overlap at 40 rollouts per row. The fixed DCT basis also requires no learned encoder, and the scale normalization in Eq. 1 removes the embodiment-specific units retained by the raw target.

L.2 Mechanism Diagnostics

The diagnostics follow the method design in Sections 3.2 and 3.3: first, whether the DCT intent target carries a comparable recovery signal across embodiments; second, whether the trained Intent Head maps human and robot recovery observations to a shared 4D representation; and third, whether the Recovery Gate activates near the annotated recovery boundary. We first test the target itself by comparing the human and robot distributions of the same $K = 4$ low-frequency DCT magnitude features used as the intent target in Eq. 1. We use the two-sample Kolmogorov–Smirnov (KS) distance for each coefficient $k \in \{0, 1, 2, 3\}$; lower KS indicates stronger cross-embodiment distributional overlap. Conditioning an action decoder on an intermediate signal takes other forms, including residual bridges from noise to intent (Zhong et al., 2026) and latent world model features (Sun et al., 2026). In EgoRecovery, this conditioning is gated by the predicted recovery state.

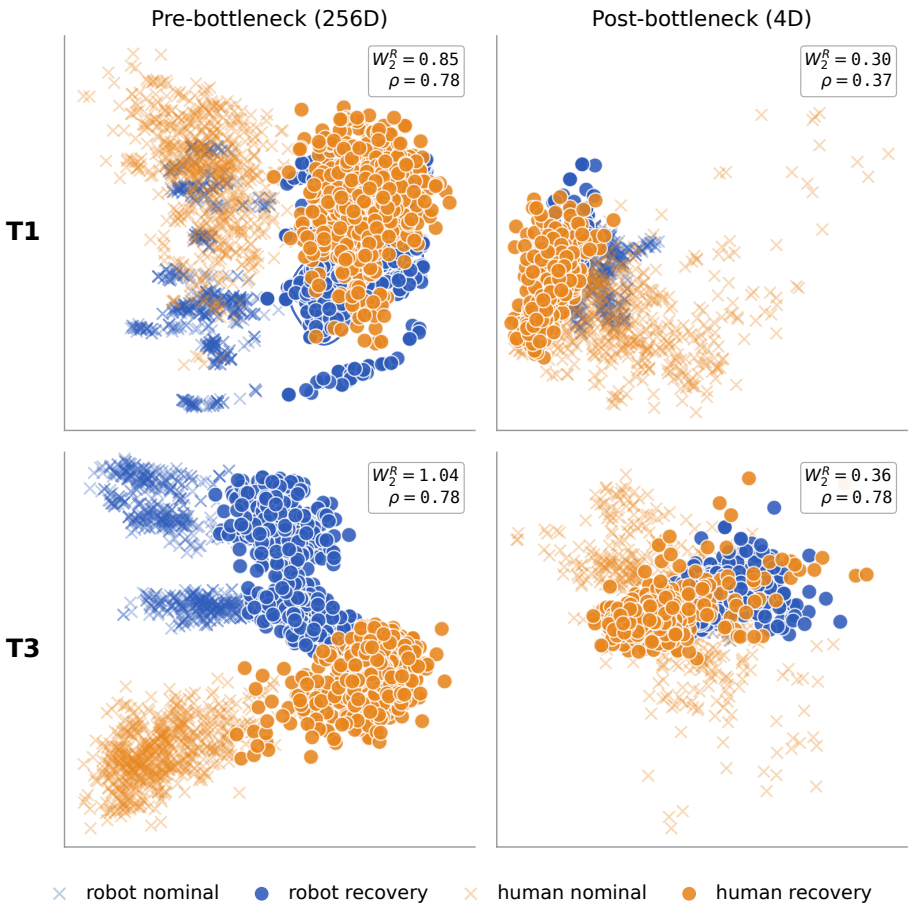


Figure 10: **Learned alignment in the corrective intent bottleneck.** PCA-2 visualizations of the pre-bottleneck Intent Head input (left) and the trained 4D intent latent c_t (right) for cup brush insertion (T1) and round disk placement (T3). Color denotes embodiment and marker shape denotes phase. The boxed values are computed in the original feature space rather than the PCA plane: W_2^R is the sliced Wasserstein-2 distance between robot and human recovery features, and $\rho = W_2^R / W_2^N$ compares recovery alignment with nominal alignment.

Recovery-window alignment is strong on round disk placement and weaker on cup brush insertion. Whiskers show 95% bootstrap confidence intervals ($n_{\text{boot}} = 1{,}000$, seed 42); the green and red lines are numerical reference guides at $\text{KS} = 0.15$ and $\text{KS} = 0.30$, respectively. On round disk placement, the full success trajectories remain embodiment specific, while the recovery-window DCT magnitude features become substantially more aligned ($\text{KS}_\text{R} = 0.11$ vs. $\text{KS}_\text{S} = 0.33$). Cup brush insertion is weaker ($\text{KS}_\text{R} = 0.21$, $\text{KS}_\text{S} = 0.17$). Its recovery often contains small left-arm translation and larger rotational adjustment, so an xyz-only DCT magnitude feature can retain embodiment-specific micro-motion differences.

We next inspect whether the trained Intent Head keeps the intended cross-embodiment structure in its learned bottleneck. For each frame, we compare the pooled trunk feature that enters the Intent Head (256D) with the predicted intent latent c_t (4D). We keep recovery frames and nominal frames with valid intent windows, excluding pure-success frames that do not receive intent supervision. Distances are computed in the original feature spaces, not in the 2D visualization. Let W_2^R denote the sliced Wasserstein-2 distance between robot and human recovery features, and let W_2^N denote the corresponding distance for nominal features. The ratio $\rho = W_2^R / W_2^N$ measures whether recovery states are more cross-embodiment aligned than nominal states.

Figure 10 shows how the Intent Head changes the distance between robot and human recovery features. The absolute robot-human recovery distance becomes much smaller after the trained bottleneck on both tasks, while random JL projection and PCA-4 provide little or moderate shrinkage

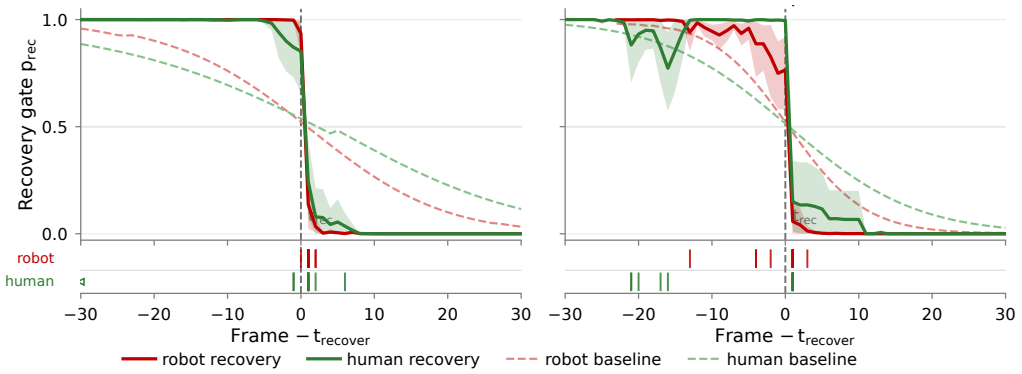


Figure 11: **Recovery Gate Head aligned to annotated recovery boundaries.** Predictions $p_{\text{rec}}(\cdot)$ are aligned to t_{recover} on 15 robot and 15 human recovery episodes per task. Solid curves show per-domain means with 95% bootstrap confidence intervals, and dashed curves show a phase-only baseline. Left and right panels show cup brush insertion and round disk placement. The gate drops near the annotated transition for both embodiments, limiting intent modulation to recovery frames.

after correcting for dimension. This indicates that the alignment is learned from the corrective intent supervision. Reducing 256 dimensions to 4 with random JL projection or PCA-4 does not produce it. We use dimension-normalized sliced Wasserstein distances, because raw sliced Wasserstein estimates have a $1/\sqrt{d}$ dependence. Under this normalization, the trained bottleneck reduces robot-human recovery distance by 22.5× on cup brush and 23.2× on round disk. The corresponding random JL projection gives only 1.2×/1.1×, and PCA-4 gives 2.0×/1.1×. The ratio ρ gives a complementary reading. In both tasks, recovery features are more aligned across embodiments than nominal features ($\rho < 1$), consistent with the data-collection goal of sampling related failure states. The bottleneck further reduces this ratio on cup brush ($0.78 \rightarrow 0.37$), but leaves it nearly unchanged on round disk ($0.78 \rightarrow 0.78$). The absolute distance reduction is consistent across both tasks, whereas the change in ρ depends on the task. On round disk, PCA-4 is almost identical to random projection, while the trained bottleneck still gives a large reduction; this suggests that the recovery alignment is not captured by the top-variance directions of the pre-bottleneck feature.

Finally, we examine the gate by aligning episodes to the human-labeled recovery boundary t_{recover} and plotting the predicted recovery probability p_{rec} for robot and human demonstrations. The phase-only baseline uses only normalized frame position, so it measures how much of the recovery label can be inferred from episode timing without visual input.

Figure 11 shows the mean gate output over aligned episodes, bootstrap uncertainty across episodes, a timing-only reference, and per-episode threshold-crossing offsets. In both tasks, p_{rec} is high before t_{recover} and drops near the annotated transition, allowing corrective intent to affect recovery frames without remaining active during nominal execution. The phase-only baseline is a logistic regression on normalized frame position, and bottom ticks mark the first frame in each episode where $p_{\text{rec}} < 0.5$. Cup brush shows the clearest visual gain over this timing-only reference (gate AUC 0.9998 vs. phase-only AUC 0.980/0.969 for robot/human). All 30 robot and human episodes cross the 0.5 threshold within ± 1 frame of t_{recover}, indicating precise boundary alignment despite the weaker KS alignment in Figure 9. On round disk placement, the evidence is weaker: the phase-only baseline is already near saturated (0.994/0.996) because recovery timing is highly regular, and human episodes show a wider crossing-offset spread (IQR $[-16.5, +1]$ frames), consistent with larger hand-motion variability. This diagnostic therefore supports gate alignment, while Table 14 tests how gate values affect closed-loop robot behavior.

Table 14: **Recovery Gate override analysis.** All rows evaluate the same trained EGORECOVERY checkpoint without retraining; only the gate value p_t that sets α_t in Eq. 2 changes. Per task cells report successes out of 20 rollouts. Pooled columns aggregate the two tasks, giving 40 rollouts per start state regime, with Wilson 95% binomial confidence intervals (Wilson, 1927). Bold values mark the best aggregate result.

Gate during evaluation	Cup brush Initial	Cup brush Recovery	Round disk Initial	Round disk Recovery	Pooled Initial SR (%, Wilson CI)	Pooled Recovery SR (%, Wilson CI)
Disabled ($p_t \equiv 0$)	13/20	10/20	15/20	12/20	70.0 [54.6, 81.9]	55.0 [39.8, 69.3]
Fixed ($p_t \equiv 0.5$)	8/20	9/20	10/20	9/20	45.0 [30.7, 60.2]	45.0 [30.7, 60.2]
Always active ($p_t \equiv 1$)	7/20	8/20	8/20	9/20	37.5 [24.2, 53.0]	42.5 [28.5, 57.8]
Shuffled p_t	6/20	7/20	8/20	7/20	35.0 [22.1, 50.5]	35.0 [22.1, 50.5]
Learned p_t (ours)	**15/20**	**16/20**	**17/20**	**17/20**	**80.0 [65.2, 89.5]**	**82.5 [68.1, 91.3]**

The learned gate gives the highest Initial and Recovery SR on both tasks. Setting $p_t \equiv 0$ removes intent modulation and mainly lowers Recovery SR, from 82.5 to 55.0 pooled over the two tasks. Fixed values of 0.5 and 1 and shuffled values, which are no longer aligned with the current observation, lower both metrics to 35.0–45.0. These overrides act on a decoder trained with the learned gate, whereas the always-on variant in Table 2 is retrained with modulation at every state and reaches 70.0/70.0 over four tasks. Both comparisons favor modulation that follows the predicted recovery state.

L.3 Data Collection and Rollout Visualization

This section provides a visual overview of the coverage of failure states, the data collection pipeline, and closed-loop EgoRecovery inference results. Tasks 1–4 correspond to cup-brush insertion, table sweep, round-disk placement, and cube stacking, respectively.

L.3.1 Qualitative Failure-State Coverage

Figure 12 provides qualitative examples of the coverage role assigned to human recovery data in the main experiments.

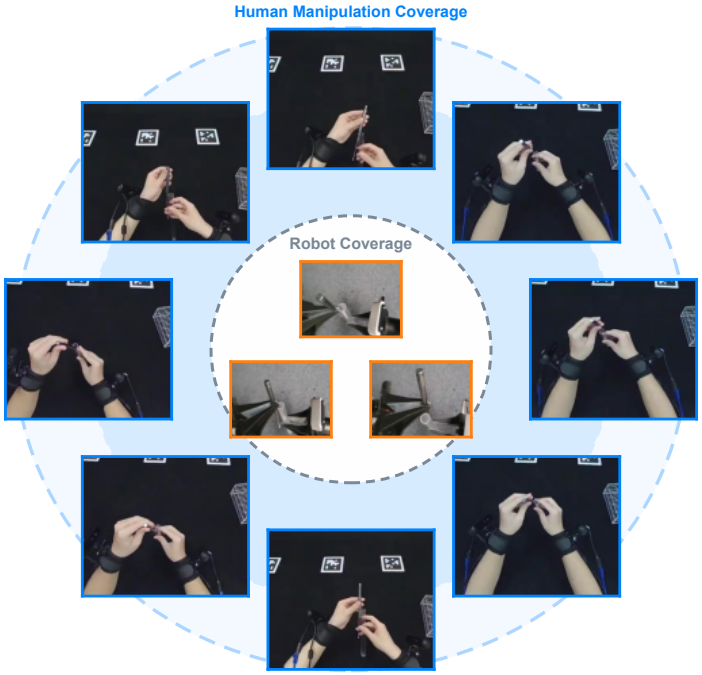


Figure 12: **Qualitative coverage of failure states for cup-brush insertion.** The center shows representative failure configurations from the 50 robot recovery episodes, while the surrounding frames show configurations from the 300 egocentric human recovery episodes. The examples illustrate how human collection adds variation within the same task-level recovery family.

L.3.2 Data Collection

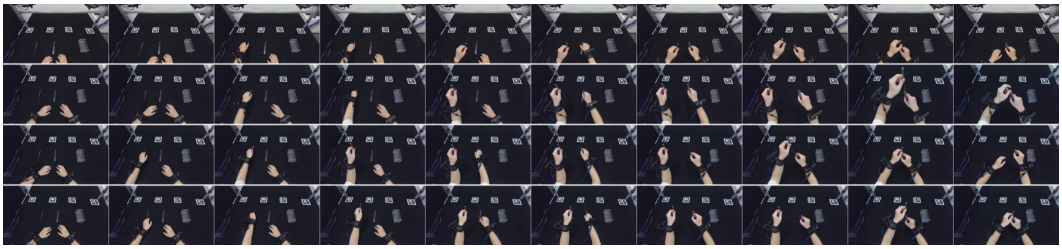


Figure 13: Human success collection for cup-brush insertion (Task 1).



Figure 14: Human recovery collection for cup-brush insertion (Task 1).

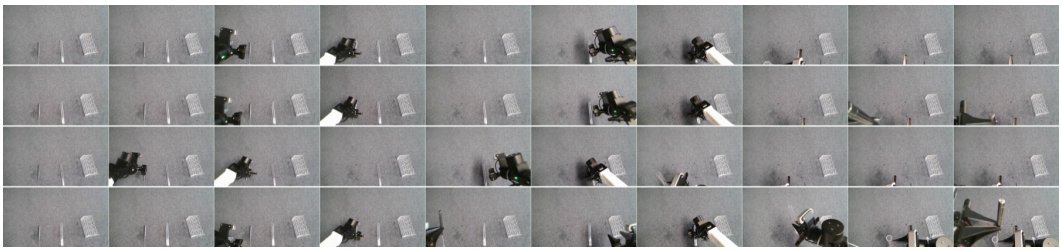


Figure 15: Robot success collection for cup-brush insertion (Task 1).

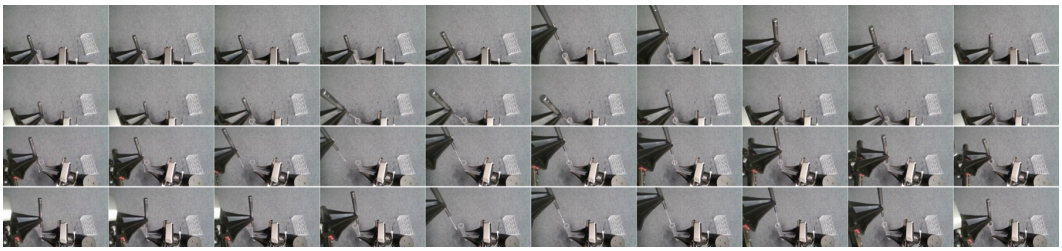


Figure 16: Robot recovery collection for cup-brush insertion (Task 1).

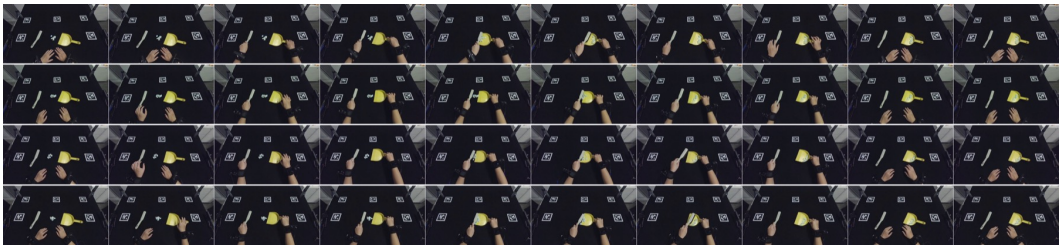


Figure 17: Human success collection for table sweep (Task 2).

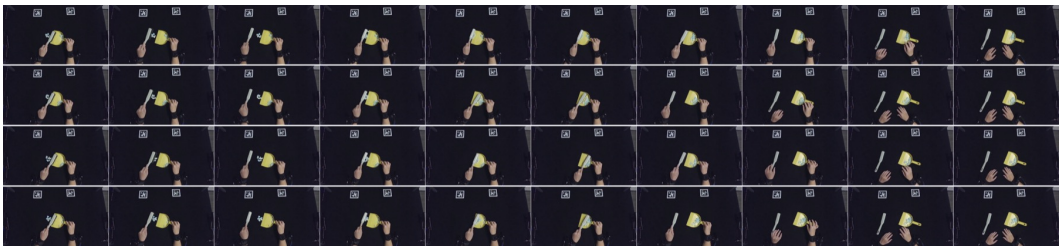


Figure 18: Human recovery collection for table sweep (Task 2).

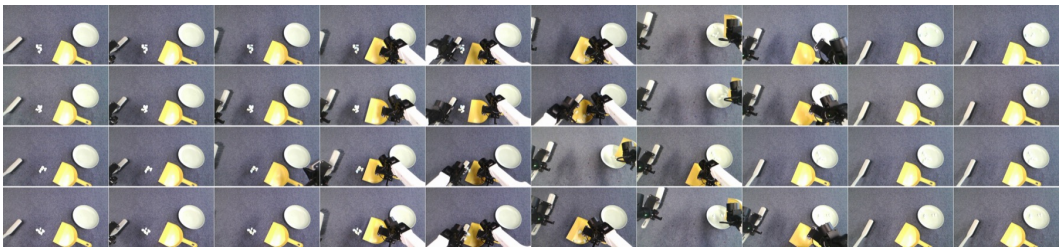


Figure 19: Robot success collection for table sweep (Task 2).

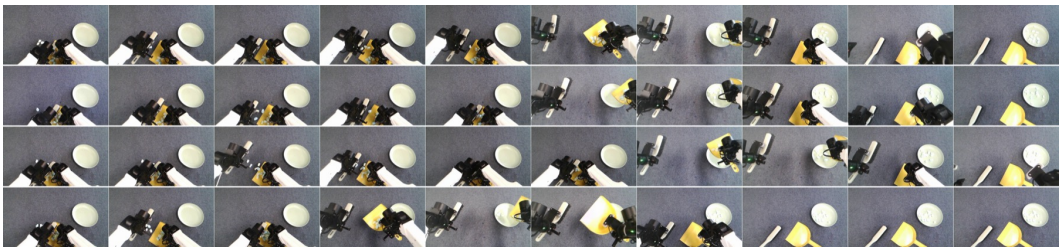


Figure 20: Robot recovery collection for table sweep (Task 2).

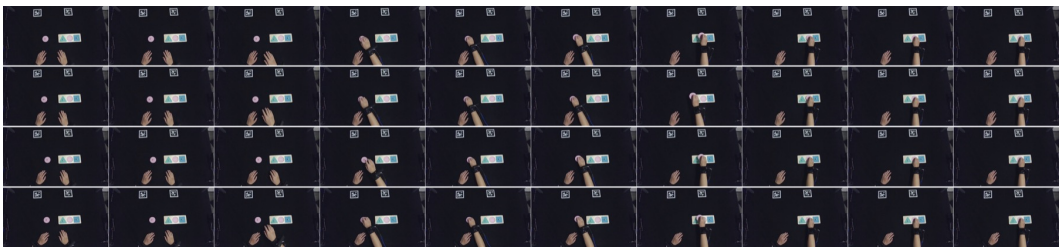


Figure 21: Human success collection for round-disk placement (Task 3).

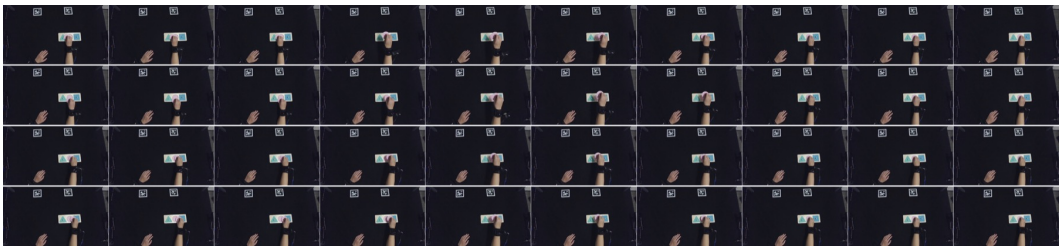


Figure 22: Human recovery collection for round-disk placement (Task 3).

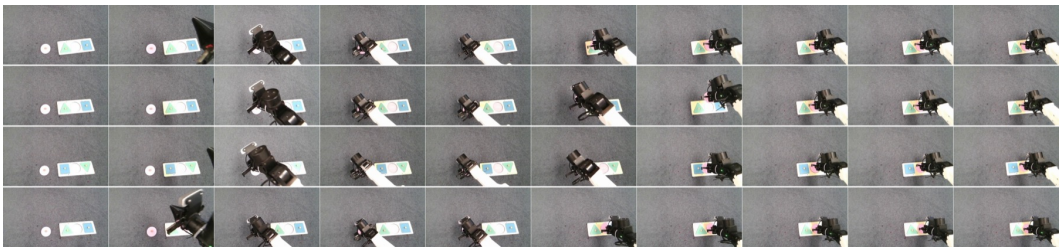


Figure 23: Robot success collection for round-disk placement (Task 3).

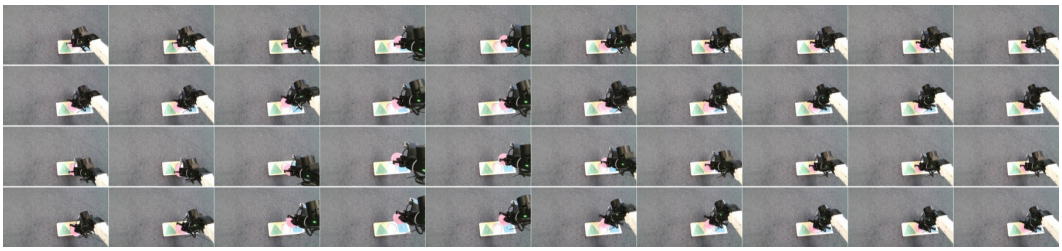


Figure 24: Robot recovery collection for round-disk placement (Task 3).

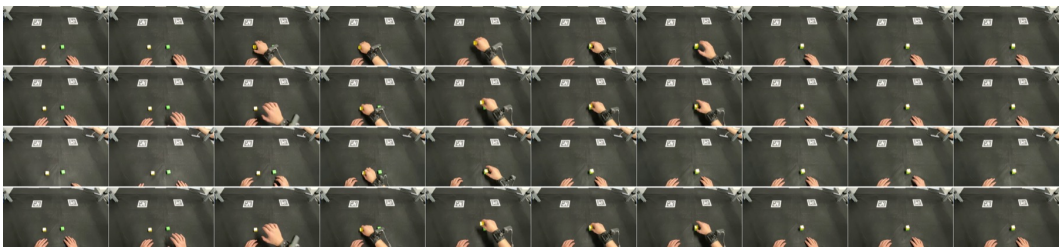


Figure 25: Human success collection for cube stacking (Task 4).

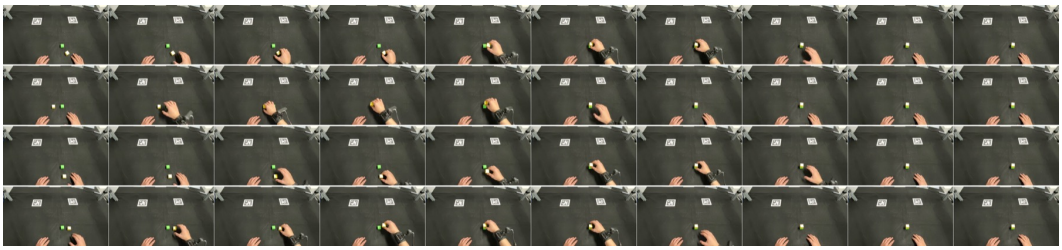


Figure 26: Human recovery collection for cube stacking (Task 4).

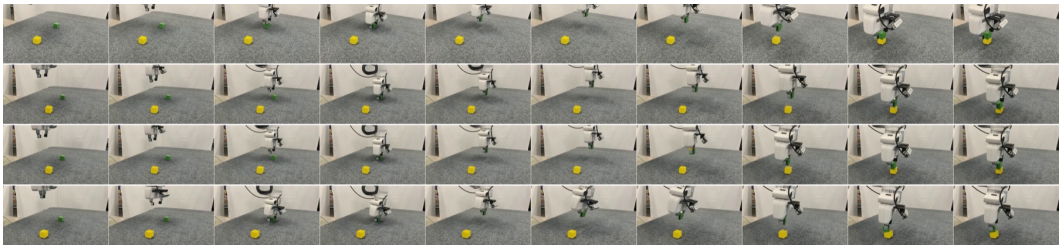


Figure 27: Robot success collection for cube stacking (Task 4).

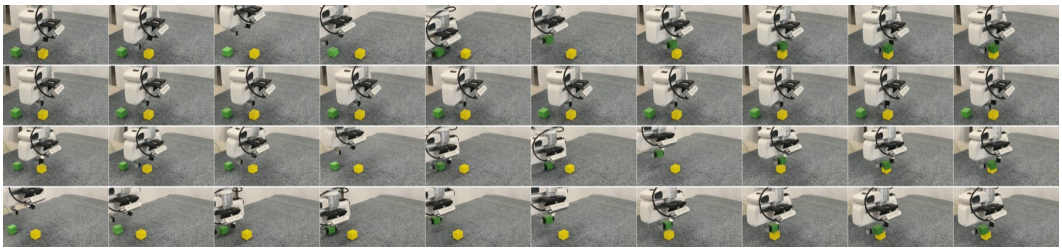


Figure 28: Robot recovery collection for cube stacking (Task 4).

L.3.3 EGORECOVERY INFERENCE



Figure 29: EGORECOVERY nominal rollout for cup-brush insertion (Task 1).



Figure 30: EGORECOVERY recovery rollout for cup-brush insertion (Task 1).

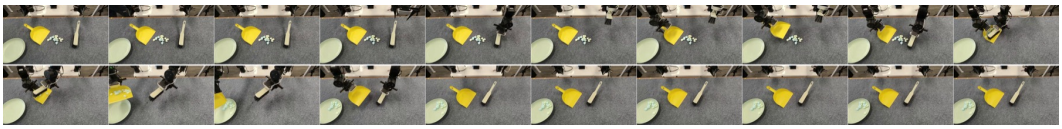


Figure 31: EGORECOVERY nominal rollout for table sweep (Task 2).

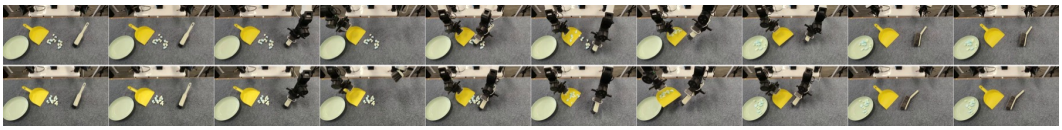


Figure 32: EGORECOVERY recovery rollout for table sweep (Task 2).

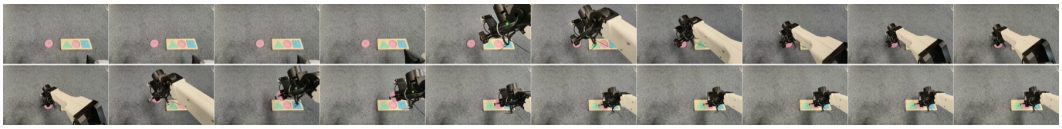


Figure 33: EGORECOVERY nominal rollout for round-disk placement (Task 3).

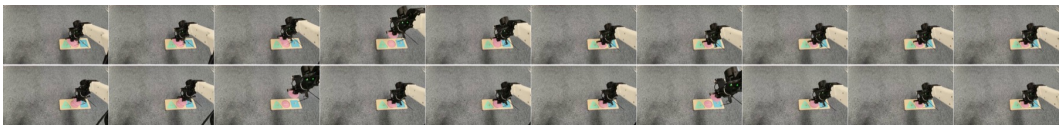


Figure 34: EGORECOVERY recovery rollout for round-disk placement (Task 3).

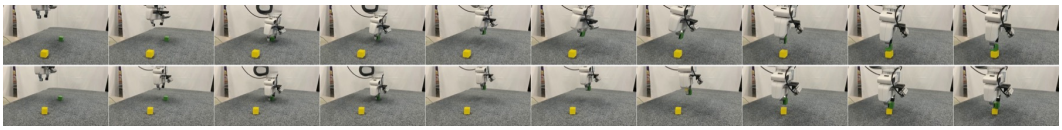


Figure 35: EGORECOVERY nominal rollout for cube stacking (Task 4).

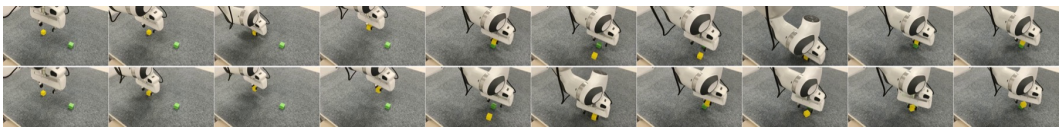


Figure 36: EGORECOVERY recovery rollout for cube stacking (Task 4).