Explicit Recovery Behavior for Diffusion Policies

Supplementary Material

Appendix

Item	Value or method
batch size	128
epoch	50
learning rate	1e-4
$lr_{scheduler}$	cosine
observation frames	2
horizon	16
Executed steps	8

Table 5. Hyper-parameters used for training REACH model.

A. Training Details In this section, we will illustrate our training process for different experiments in detail.

We use a unified training setting for all models trained for evaluation in the robomimic and mimicgen simulation and real-world task.

For data processing, we extract the action-prompt in the dataset and embed them into the same size of the visual image feature-space. So the condition feature map of the **Conditioned diffusion policy** is twice the size of the diffusion-policy.

When the model is trained in the robomimic and mimicgen dataset, the training setting is treated the same as the setting in [35] and [36]. For the real-task training setting, we collect the three-views image using the realsense D435 with the resolution (640,480) and resize them into the (84,84) resolution to accelerate the training processing. For the action-space, we use the joint-angle as the control-signal. The training-hyperparameters used are shown in Tab 5.

Error Detector Training Regime. The error detector is purposefully trained using the *same dataset as the policy training*, requiring no additional task-specific data collection or explicit out-of-distribution (OOD) supervision. We evaluate the framework under different demonstration regimes: **PH** (proficient operator demonstrations), **MH** (multiple operator demonstrations), and **Paired** (both successful and failed trajectories). Training the error detector under these different regimes alters its observed distribution, which inherently shifts the ID/OOD decision boundary, influences recovery triggering, and ultimately impacts final performance.

B. Real-robot Experimental Setup and Deployment Camera calibration is essential to establish the geometric relationship between the image plane and the physical workspace, enabling accurate perception, consistent multi-view alignment, and reliable vision-based control (Fig 10).

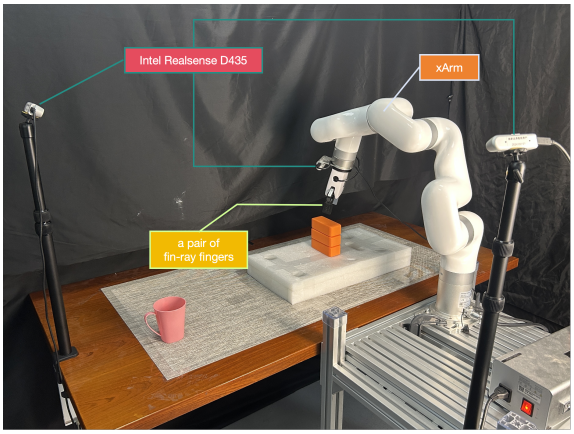

Figure 10. Real-robot experimental setup

Extended Tasks. To demonstrate the broader scope of our framework beyond standard pick-and-place variants, we evaluate REACH on complex, non-pick-and-place tasks including "Open the drawer" and "Fold clothes" (Fig 11). The quantitative improvements for these tasks are reported in Tab 6.

Figure 11. Left: Open the drawer; Right: Fold clothes

	Open the drawer	Fold clothes
Diffusion	23%	28%
REACH (-ED)	26%	31%
REACH (+ED)	**35%**	**38%**

Table 6. Real-world experiments for extended non-pick-and-place tasks.

Real-Time Practicality & Rollback Frequency. Regarding deployment overhead, the execution time per step for REACH is almost equivalent to the standard diffusion policy. However, due to the safe rollback mechanism actively preventing failures, the number of repeated trials in-

creases to approximately three times that of the original policy specifically at crucial, bottleneck scenes.

C. Probability Distribution and Uncertainty Statistics

We analyze the statistical properties of the reconstruction-error distribution produced by the autoencoder-based error detector. The empirical distribution is visualized across training and evaluation trajectories (Fig 12).

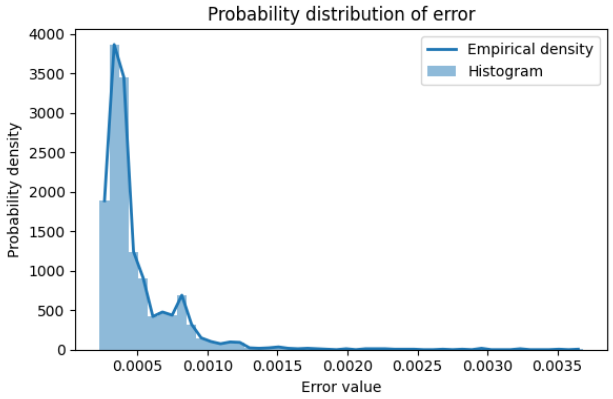

Figure 12. Probability distribution of the error-detector outputs

Although the detector lacks explicit OOD supervision, the validation demonstrates that the reconstruction error successfully separates ID and OOD observations, yielding a clear discriminative gap between failure and success scenes. To rigorously quantify robustness across trials, we utilize **Conformal Prediction (CP)**. CP determines the precise threshold for the generation boundary based on a target confidence level, dictating exactly when a state is classified as OOD and triggering recovery.

D. Extended Simulation Benchmarks and Ablations

We expand our simulation evaluation to additional Robomimic tasks (Lift, Transport, and Toolhung) to thoroughly compare REACH against baselines under different training conditions (Tab 7).

Action Steering and Hyperparameter S: Our Action Steering Analysis illustrates the distinct qualitative effects of negative versus positive prompts. Specifically, we ablate the classifier-free guidance scale (S) during sampling. As shown in Tab 7, different S values yield varying success rates despite using the same error-detector and threshold. The final performance under optimal S generally surpasses what is achievable by merely resampling actions. This strongly suggests that the negative steering property of the guidance is actively and effectively steering the policy away from false actions.

Policy	Lift(MH)	Transport(MH)	Toolhung
LSTM/BC-RNN	1.00/0.93	0.60/0.20	0.67/0.31
IBC/BC	0.15/0.02	0.00/0.00	0.00/0.00
Diffusion	1.00/0.97	0.68/0.46	0.50/0.30
Cond. DP	1.00/0.98	0.69/0.48	0.52/0.34
PH-S0	1.00/0.97	0.70/0.49	0.51/0.32
PH-S0.2	1.00/0.98	0.72/0.50	0.55/0.33
PH-S2.5	1.00/0.98	0.74/0.52	0.54/0.32
PH-S4.5	1.00/0.98	0.71/0.51	0.55/0.35
MH-S0	1.00/0.97	0.69/0.49	0.51/0.33
MH-S0.2	1.00/0.98	0.72/0.49	0.54/0.32
MH-S2.5	1.00/0.98	0.75/0.53	0.54/0.33
MH-S4.5	1.00/0.98	0.70/0.50	0.55/0.35

Table 7. Simulation comparison across additional Robomimic tasks and guidance scale (S) ablations.

E. Comparisons to Other Methods and Limitations

Compared to alternative robustness methods that rely heavily on additional data collection or external VLM-based perception models, **REACH** achieves enhanced robustness internally. It uses only the native policy-training dataset and a lightweight AE-based detector, making it highly preferable when external supervision is unavailable or computationally prohibitive.

Limitations: A fundamental assumption of our framework is environment reversibility. While certain dynamic actions may irreversibly disturb the environment upon execution, a significant portion of these failures is rooted in erroneous action samples being planned. REACH mitigates this limitation by detecting these failure-inducing actions early in the sampling phase, steering the policy to resample safer actions before an irreversible physical state is reached.