REWORLD: ADAPTIVE CORRECTIVE SIMULATION FROM WORLD MODELS FOR EMBODIED POLICY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Robotic policies trained from successful demonstrations receive little supervision for recovering from off-nominal states, allowing execution errors to compound into task failure. Physical corrective collection is costly, and task-specific simulators may be unavailable. We introduce ReWorld, which uses an action-conditioned world model as a local simulator to synthesize adaptive corrective experience around demonstrations. ReWorld probes the base policy at predicted perturbation endpoints and retains regions from which it repeatedly fails to recover. For retained cases, a motion planner returns the robot toward the source demonstration and the expert suffix completes the corrective action chunk. The predicted observation and constructed action form corrective training pairs, while competence-aware allocation concentrates supervision on weaker tasks. The world model d planner are used only offline, leaving deployment unchanged. Across 50 RoboTwin 2.0 tasks, ReWorld improves SmolVLA from 38.72% to 47.76% and $\pi_{0.5}$ from 75.64% to 83.32%. Controlled studies show that selected corrections are more data-efficient than unfiltered corrections, local recovery probes track execution difficulty, and recovery transfers to held-out perturbation magnitudes. Code will be released.

1 INTRODUCTION

World models promise to become more than generators of plausible futures. As Li and colleagues emphasize, one of their most consequential roles is to serve as simulators in which embodied agents learn through action and consequence (Li et al., 2025). Coupled with an action model, a world model could form a closed perception–action loop. The action model decides what the robot should do, the world model predicts how the environment responds, and the predicted observation informs the next decision. Recent video world models have moved toward this vision by generating realistic robot interactions, predicting action-conditioned futures, and synthesizing trajectories for policy learning (Yang et al., 2023; Zhu et al., 2025; Jang et al., 2025; Liu et al., 2026). Yet generating plausible video is not the same as providing a world in which a robot can reliably learn. Once predictions are fed back into the policy, small inconsistencies can alter subsequent actions and push later predictions farther from realistic interaction.

This problem becomes more pronounced over complete manipulation episodes. A useful simulator must preserve geometry, contact, task progress, and action consequences across repeated interactions, including those that depart from demonstrated behavior. Existing world models have made encouraging progress, but reliably replacing physical interaction with long-horizon generated experience remains challenging. This raises the question that motivates our work. Must a world model faithfully simulate an entire task before it can provide useful experience for policy learning? We argue that it does not. Between generating isolated videos and replacing the physical environment lies a more focused opportunity. A world model may already be useful if it can simulate short, task-relevant interactions around trajectories known to succeed.

Such local interactions address a structural blind spot in robot learning. Successful demonstrations show how a task proceeds when earlier actions have gone well, but provide little guidance once execution departs from the demonstrated path. A small error in end-effector pose, gripper position, or contact timing may leave the observation nearly unchanged while making the nominal action

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.

inappropriate. The next action can then produce a larger deviation, allowing a minor error to grow into task failure. More nominal demonstrations improve coverage of successful execution, while label-preserving augmentations diversify appearance. Neither directly provides the missing supervision that connects off-nominal observations to actions that keep the task feasible.

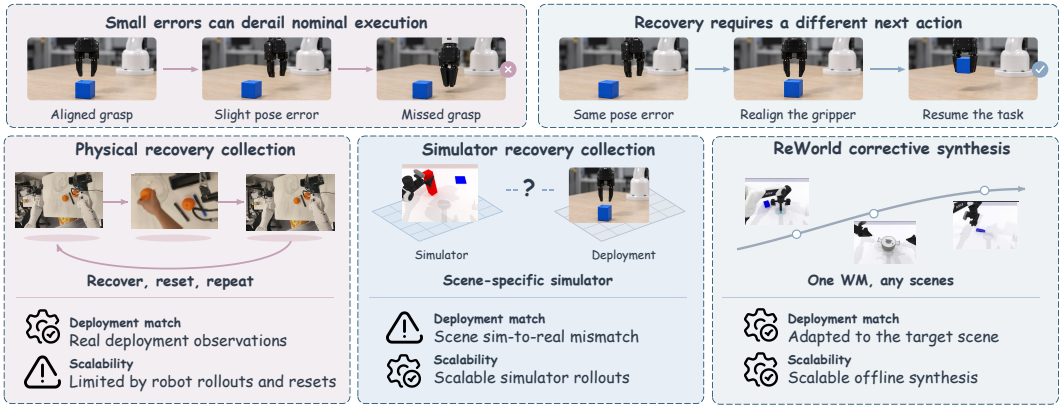

Figure 1. Motivation of ReWorld. Small execution deviations can require a different next action. Physical collection provides deployment-matched observations but requires repeated rollouts and resets, while task-specific simulators may be unavailable or mismatched to the target domain. ReWorld uses a frozen action-conditioned world model to synthesize local perturbed observations and pairs them with automatically constructed recovery actions for direct policy post-training. The world model and planner are used only during offline data construction, while deployment retains the original policy.

Figure 1 summarizes this motivation. Physical interaction provides deployment-matched corrections but requires repeated rollouts and resets, while task-specific simulation may be unavailable or mismatched (Kelly et al., 2019; Hu et al., 2026; Physical Intelligence et al., 2025a; Mandlekar et al., 2023). ReWorld instead uses a world model as a local simulator around successful behavior, where short counterfactual interactions expose the recovery supervision missing from demonstrations.

We introduce **ReWorld** (Recovery from World Model), a framework that turns a world model into an adaptive generator of corrective experience for embodied policy post-training. Rather than using the world model to reproduce complete task executions, ReWorld asks what the robot would observe if its execution deviated from a successful demonstration and what it should do next to keep the task feasible. Starting from successful demonstrations, an action-conditioned world model predicts the visual consequences of structured action perturbations, exposing the policy to off-nominal states without physically executing the corresponding mistakes. ReWorld then constructs corrective action chunks through motion planning and demonstration reuse. The resulting examples retain the observation–action interface of ordinary policy training data while adding supervision precisely where successful demonstrations are silent.

Crucially, ReWorld does not generate corrections indiscriminately. It diagnoses the base policy's local recovery capability by building a policy-conditioned recoverability table. At each predicted perturbation endpoint, the base policy is queried once, and its action chunk is assessed using local kinematic and gripper-command criteria. Perturbations from which the policy repeatedly fails to recover are prioritized during data generation. Their predicted observations are paired with motions toward demonstration reference states or with gripper-reopening commands, followed by reused demonstration actions that continue the task. During multitask post-training, task-level competence further controls corrective-data sampling and loss weighting, concentrating supervision on weaknesses without allowing low-performing tasks to dominate training. This creates a closed improvement cycle in data space. The world model reveals deviations, the policy identifies which ones expose recovery gaps, and automatic action construction supplies the missing corrections.

At deployment, the policy acts directly from incoming observations using its original architecture, without world-model rollouts, explicit failure detection, or a separate recovery controller. We evaluate ReWorld on RoboTwin 2.0 (Chen et al., 2025) with simple SmolVLA and $\pi_{0.5}$ (Shukor et al., 2025; Physical Intelligence et al., 2025b), two policy backbones with substantially different capacities and baseline performance. ReWorld improves overall task success from **38.72% to 47.76%** for SmolVLA

and from **75.64% to 83.32%** for $\pi_{0.5}$, yielding absolute gains of **9.04** and **7.68** percentage points. The consistent gains across both backbones suggest that locally simulated corrective experience can improve closed-loop execution even when the underlying policy begins from markedly different levels of competence.

Our contributions are threefold. **(i)** First, we introduce **ReWorld**, a framework that combines world-model synthesis of structured execution deviations with automatic corrective-action construction, while probing the base policy to target its local recovery gaps. **(ii)** Second, we propose a policy-conditioned recoverability diagnosis that identifies local recovery gaps at synthesized perturbation endpoints and uses them to target subsequent corrective-data generation, with competence-aware sampling and weighting balancing supervision across tasks. **(iii)** Third, ReWorld distills corrective experience directly into embodied policies without modifying their deployment architecture, and consistently improves closed-loop task success across SmolVLA and $\pi_{0.5}$ on RoboTwin 2.0.

2 Related Work

Embodied Policies and Behavior Learning. Embodied policies combine expressive action modeling with large-scale multimodal pretraining. Diffusion and flow-matching policies support continuous visuomotor control (Chi et al., 2025; Team et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025b; Shukor et al., 2025), while RT-2 and OpenVLA use multimodal autoregressive prediction (Brohan et al., 2023; Kim et al., 2024). DreamZero further couples future observations and actions (Ye et al., 2026). ReWorld is architecture-agnostic, adding off-nominal observation and corrective-action pairs to the training distribution.

Corrective Learning, Recovery, and Data Selection. Corrective learning addresses covariate shift through paradigms. DAgger, DART, HG-DAgger, and RaC query experts on policy-induced or perturbed executions (Ross et al., 2011; Laskey et al., 2017; Kelly et al., 2019; Hu et al., 2026). RECAP and RePO-VLA incorporate recovery experience into policy optimization (Physical Intelligence et al., 2025a; Liufu et al., 2026). Offline methods synthesize or curate corrections, including CCIL, MimicGen, DemoGen, and CUPID (Ke et al., 2024; Mandlekar et al., 2023; Xue et al., 2025; Agia et al., 2025), while RACER uses an online vision-language supervisor (Dai et al., 2025). ReWorld complements these methods by probing recoverability before constructing corrective actions.

World Models for Policy Training. World-model methods expand robot experience without executing every trajectory physically. DreamGen and Ctrl-World generate imagined experience for policy learning (Jang et al., 2025; Guo et al., 2026), while Hi-WM and Dream2Fix introduce intervention or structured recovery supervision within imagined interaction (Li et al., 2026b;a). ReWorld uses the world model locally and conditionally, predicts observations after structured perturbations, and constructs the corresponding actions through planning and demonstration reuse. The resulting pairs post-train the original policy without world-model inference at deployment.

3 Method

3.1 Problem Formulation

We study corrective post-training of an embodied policy from demonstrations. At time t, the policy maps a visual observation o_t, robot state s_t, and task instruction ℓ to an H-step action chunk

$$A_t = (a_t, \dots, a_{t+H-1}), \qquad A_t \sim \pi_\theta(\cdot \mid o_t, s_t, \ell), \tag{1}$$

where each a_{t+k} is one robot action at time $t+k$, and s_t collects the robot pose, joint, and gripper state available to the policy. Given nominal demonstrations $\mathcal{D}_N = \{(o_t^N, s_t^N, \ell, A_t^N)\}$ and a pretrained policy π_{θ_0}, our goal is to construct a corrective dataset $\mathcal{D}_C$ of off-nominal observations paired with recovery actions and use it to improve closed-loop recovery without modifying the policy architecture. We assume a frozen action-conditioned world model W_ϕ and a motion planner $\mathcal{P}$, both used only for offline data construction. Because the world model, planner, and probing policy are fixed during corrective-data construction, the offline cache is distributionally equivalent to executing the same fixed construction procedure online and recording the resulting pairs. We provide the formal equivalence argument in the supplementary material. And Figure 2 summarizes the three stages of ReWorld, from perturbation synthesis to competence-aware post-training.

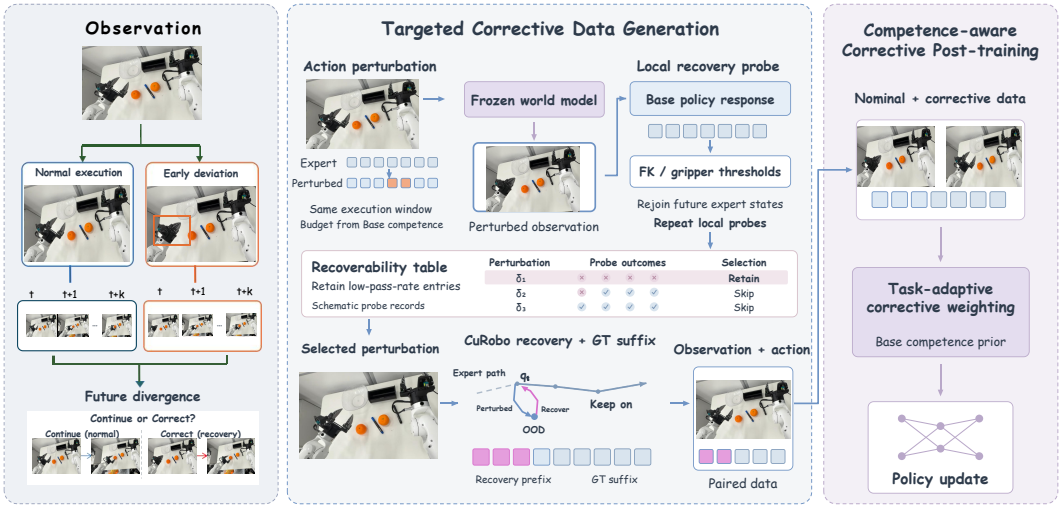

Figure 2. Overview of ReWorld. Panel (a) identifies early deviations by comparing nominal and perturbed executions. Panel (b) generates corrective experience by rendering structured perturbations with a frozen action-conditioned world model, probing the base policy, retaining low-recoverability cases, and pairing each retained observation with a planned recovery prefix and expert continuation. Panel (c) applies competence-aware post-training to the resulting corrective pairs, while the world model and planner remain offline components.

3.2 Identifying Useful Deviations

The formulation above leaves open which deviations should enter $\mathcal{D}_C$. Some perturbations are already recoverable, so correcting them wastes the data budget. Because recoverability depends on manipulation context as well as perturbation type and magnitude, we use the current policy to identify local recovery gaps.

We probe controlled deviations around successful demonstrations. A perturbation unit u specifies its task context, manipulation phase, active arm, perturbation type, direction, and magnitude. Algorithm 1 summarizes this policy-specific probing procedure. We consider end-effector translation, end-effector rotation, and premature gripper closure. For a demonstration segment beginning at t, the nominal action chunk A_t^N is replaced by a perturbed chunk A_t^δ. The perturbed command gives the resulting robot state $s^\delta = \mathrm{State}(s_t^N, A_t^\delta)$, while the frozen world model predicts the visual endpoint $\hat{o}^\delta = W_\phi(o_t^N, s_t^N, A_t^\delta)$. We then query the base policy with $\widetilde{A}^\delta \sim \pi_{\theta_0}(\cdot \mid \hat{o}^\delta, s^\delta, \ell)$.

For translation and rotation, forward kinematics compares the predicted endpoint with a short expert window. We mark a response recoverable when position error is at most $0.03\,\mathrm{m}$ or rotation error is at most 10°. For premature closure, recovery requires reopening the affected gripper to at least 0.8. These criteria measure local state recovery rather than eventual task success.

Each perturbation unit is evaluated on $N_u = 4$ demonstration segments. Let $r_{u,j} \in \{0, 1\}$ indicate whether response j satisfies the local criterion. Its empirical recovery rate is

$$\hat{\rho}_u = \frac{1}{N_u} \sum_{j=1}^{N_u} r_{u,j}. \tag{2}$$

We retain units with $\hat{\rho}_u \leq 0.25$ and assign failure weight $w_u = 1 - \hat{\rho}_u$. The table therefore concentrates synthesis on regions the base policy rarely resolves.

With a finite probing budget, we allocate more candidate perturbations to tasks on which the base policy is less competent. Competence is estimated on an independent calibration set that is disjoint from the final benchmark. For task i with n_i^+ successes among n_i trials, we use

$$p_i = \frac{n_i^+ + 1}{n_i + 2}, \qquad B_i = \mathrm{round}\left(B_{\min} + \Delta B(1 - p_i)\right). \tag{3}$$

Task competence determines *where to probe*, while $\hat{\rho}_u$ determines *what to retain*. The table is fixed after probing and identifies where supervision is needed, not how to recover.

Algorithm 1 Policy-Specific Recovery Probing

Require: Nominal data $\mathcal{D}^N$, base policy π_{θ_0}, frozen world model W_ϕ, task budgets $\{B_i\}$
Ensure: Recoverability table $\mathcal{R}$

$\mathcal{R} \leftarrow \varnothing$
for each task i **do**
  for each perturbation unit u sampled within budget B_i **do**
    for $j = 1, \ldots, N_u$ **do**
      Sample a nominal demonstration segment $(o_t^N, s_t^N, A_t^N, \ell)$
      Construct the perturbed action chunk A_t^δ from A_t^N
      $s^\delta \leftarrow \mathrm{State}(s_t^N, A_t^\delta)$
      $\hat{o}^\delta \leftarrow W_\phi(o_t^N, s_t^N, A_t^\delta)$
      Query the base policy and obtain $\widetilde{A}^\delta \sim \pi_{\theta_0}(\cdot \mid \hat{o}^\delta, s^\delta, \ell)$
      $r_{u,j} \leftarrow \mathrm{LocalRecoverable}(\widetilde{A}^\delta, s^\delta)$
    end for
    $\hat{\rho}_u \leftarrow \frac{1}{N_u} \sum_{j=1}^{N_u} r_{u,j}$
    if $N_u \geq 4$ and $\hat{\rho}_u \leq 0.25$ **then**
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{(u, 1 - \hat{\rho}_u)\}$
    end if
  end for
end for
return $\mathcal{R}$

3.3 Constructing Corrective Experience

The recoverability table identifies where correction is needed, but not the corrective action. We construct each label from the source demonstration and a motion planner. For each retained perturbation pattern, we regenerate candidate endpoints $(\hat{o}^\delta, s^\delta)$ and query the base policy again. Because the table aggregates multiple demonstrations, a particular candidate may remain recoverable. We retain it only when it fails the probing criterion.

For end-effector translation and rotation, the source demonstration state at the beginning of the perturbed segment is the recovery target. Starting from s^δ, the planner $\mathcal{P}$ generates a recovery prefix P^δ that reconnects the robot to the demonstrated trajectory. We append the corresponding expert continuation $A_t^{N,\mathrm{suffix}}$ and align the sequence to the policy horizon according to

$$A^C = T_H \left(P^\delta \oplus A_t^{N,\mathrm{suffix}} \right), \tag{4}$$

where $\oplus$ denotes temporal concatenation and T_H truncates or pads the sequence to H steps. We use CuRobo for the recovery prefix. Returning to the source state preserves the local task context, so the expert suffix remains executable. We retain a constructed sample only when the planner reaches the target within the pose and gripper tolerances, the returned state can execute the expert suffix under the same local validity checks, and the concatenated trajectory passes collision, joint-limit, and action-validity checks. Otherwise, the candidate is discarded. For premature gripper closure, the prefix instead reopens the affected gripper while keeping the arm pose fixed before appending the expert continuation. The current implementation therefore covers Cartesian arm motions and gripper actions, while the same construction can be extended to other end effectors with suitable state and motion-planning interfaces.

The predicted perturbation observation and constructed corrective action form the corrective dataset

$$\mathcal{D}_C = \left\{ (\hat{o}^\delta, s^\delta, \ell, A^C) \right\}. \tag{5}$$

These samples preserve the policy interface and can be mixed with nominal demonstrations. The world model and planner are removed at deployment.

3.4 Competence-Aware Corrective Post-Training

Once corrective pairs are constructed, we allocate them across tasks. Weak tasks benefit from more recovery examples, whereas excessive correction on reliable tasks can interfere with nominal behavior. We reuse the competence estimate p_i from Section 3.2.

For task i, we define a target corrective share

$$c_i = \text{clip}\left(c_{\min} + \alpha(1 - p_i),\, c_{\min},\, c_{\max}\right), \tag{6}$$

where α controls the dependence on competence and $c_{\min}, c_{\max}$ bound the amount of corrective supervision. Lower-competence tasks receive larger corrective shares while every task retains nominal supervision.

We sample nominal tasks uniformly and sample corrective tasks according to the odds of their target corrective shares

$$\omega_i = \frac{c_i}{1 - c_i}, \qquad P_N(i) = \frac{1}{K}, \qquad P_C(i) = \frac{\omega_i}{\sum_{j=1}^{K} \omega_j}. \tag{7}$$

Both sample types use the native action-learning objective, with a corrective coefficient chosen to match the target share c_i. Calibration, probing, data construction, and final evaluation use disjoint scene-seed pools, so no seed used for competence estimation or correction selection appears in the final benchmark. All allocation decisions are fixed before testing. Deployment retains the original architecture and requires no world-model rollout or recovery controller.

4 Experiments

We evaluate the claims in the same order as the method. First, adaptive corrective experience should improve closed-loop performance across policy backbones. Next, policy-conditioned probing and competence-aware allocation should focus supervision where it is most useful. Finally, learned recovery should extend beyond the exact perturbations used for synthesis. We additionally evaluate physical transfer separately because the world model and planner are removed before deployment.

4.1 Experimental Setup

We evaluate 50 manipulation tasks in RoboTwin 2.0 using SmolVLA and $\pi_{0.5}$, which differ substantially in capacity and base performance. For both backbones, ReWorld uses a frozen GE-Sim world model (Liao et al., 2026), adapted to the target robot and visual domain, to synthesize perturbed observations during offline corrective-data construction. GE-Sim is never queried during policy inference. We compare ReWorld with CCIL (Ke et al., 2024), CUPID (Agia et al., 2025), Ctrl-World (Guo et al., 2026), and RACER (Dai et al., 2025). RACER uses Qwen3-VL-8B (Bai et al., 2025) as an online vision-language supervisor. The main benchmark evaluates each method on 50 tasks with 50 seeds per task, totaling 2,500 task trials per method. Controlled studies use the matched protocols described with each analysis and start from the same base policy with matched optimization settings. Competence calibration, recovery probing, corrective-data construction, and final benchmark evaluation use disjoint scene-seed pools. Here Base denotes the corresponding policy after standard RoboTwin task adaptation and before ReWorld or any comparison method is applied. Additional implementation and evaluation details are provided in the supplement.

Table 1. Mean task success on 50 RoboTwin 2.0 tasks with 50 evaluation seeds per task. Success rates are percentages and gains are absolute percentage points over the corresponding base policy. RACER uses online Qwen3-VL-8B guidance, while ReWorld retains the original deployment architecture.

	SmolVLA		$\pi_{0.5}$	
Method	Success	Gain	Success	Gain
Base	38.72	N/A	75.64	N/A
CCIL	42.48	+3.76	74.64	−1.00
CUPID	41.92	+3.20	77.72	+2.08
Ctrl-World	42.04	+3.32	76.92	+1.28
RACER + Qwen	43.00	+4.28	78.60	+2.96
ReWorld	**47.76**	**+9.04**	**83.32**	**+7.68**

4.2 Main Closed-Loop Performance

Table 1 compares ReWorld with corrective-label synthesis, policy-aware data curation, world-model-based experience generation, and online recovery supervision. ReWorld improves SmolVLA from 38.72% to 47.76% and $\pi_{0.5}$ from 75.64% to 83.32%, corresponding to gains of 9.04 and 7.68

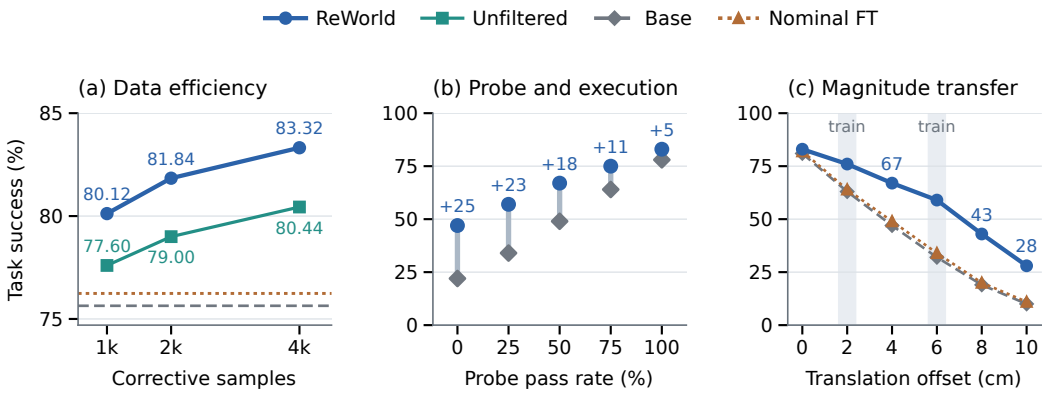

Figure 3. Controlled evaluation of ReWorld. Panel (a) compares selected and unfiltered corrective data at matched budgets. Panel (b) relates local recovery-probe pass rates to actual simulator execution success. Panel (c) evaluates transfer to perturbation magnitudes held out during corrective-data construction.

percentage points. It exceeds the strongest compared baseline, RACER, by 4.76 points on SmolVLA and 4.72 points on $\pi_{0.5}$. Ctrl-World improves both base policies only modestly, with gains of 3.32 and 1.28 points. These results show that adaptive corrective experience provides larger benefits than generic imagined experience or online recovery supervision under the same evaluation protocol. CCIL, CUPID, and Ctrl-World improve at least one backbone, but less consistently. RACER adds an online Qwen supervisor, whereas ReWorld obtains the largest gains while preserving the original deployment interface. This supports selecting states according to the policy's recovery gaps and supplying the missing action supervision offline.

The main results establish that corrective post-training is effective, but they do not show whether the gain comes from generating more off-nominal data or from selecting deviations that expose genuine recovery gaps. We isolate policy-conditioned selection through matched corrective-data budgets and an independent validation of the local recovery probe.

Adaptive corrections provide greater value per sample. Figure 3(a) compares ReWorld-selected corrections with an unfiltered variant using the same demonstrations, candidates, world model, action construction, and validity checks. Only policy-based selection differs. ReWorld outperforms unfiltered correction at every matched budget. One thousand selected samples reach 80.12%, nearly matching four thousand unfiltered samples at 80.44%, while two thousand selected samples reach 81.84%. Selection therefore increases the value of each corrective sample rather than adding arbitrary off-nominal states. The comparison measures data efficiency, since probing adds computation.

The local probe identifies execution difficulty. Because the table uses a local criterion rather than full-episode execution, we test whether its signal tracks closed-loop difficulty. We group units by the fraction of four probes that pass and evaluate Base and ReWorld from matched simulator states on disjoint scenes. Figure 3(b) shows that Base success increases from 22% to 78% as probe pass rate rises from 0% to 100%. ReWorld improves every bin, with gains of 25 and 23 points in the two least recoverable bins compared with 5 points in the fully recoverable bin. The probe thus identifies both execution difficulty and high-value correction regions.

4.3 Why Competence-Aware Correction?

Uniform correction creates a cross-task trade-off. Table 2(b) shows that increasing correction helps weak tasks but can interfere with reliable ones. Increasing uniform corrective strength from $\gamma = 0.5$ to $\gamma = 2$ raises Low-group success from 60.00% to 68.00% while reducing High-group success from 92.00% to 82.00%. This pattern shows that corrective supervision should not be increased uniformly.

At matched strength, allocation preserves Low-group success at 68.00% and restores High-group success to 91.00%, yielding 83.08% overall. Table 2(a) shows that using competence in exploration

Table 2. Competence-aware corrective post-training on $\pi_{0.5}$. Panel (a) studies adaptive allocation during failure-region exploration and corrective supervision. Panel (b) studies corrective-supervision strength under a fixed corrective-data pool. Low, Mid, and High groups are defined from the base-policy competence prior before post-training. All entries report task success rates in percent.

(a) Competence-aware allocation

Explore	Supervise	All	Low	Mid	High
Uniform	Uniform	79.00	61.00	76.00	88.00
Adaptive	Uniform	79.88	64.00	77.00	88.00
Uniform	Adaptive	81.44	65.00	79.00	89.48
Adaptive	Adaptive	**83.08**	**68.00**	**80.00**	**91.00**

(b) Corrective-supervision strength

Supervision	γ	All	Low	Mid	High
Zero correction	0	76.32	44.00	71.00	92.48
Uniform	0.5	81.08	60.00	77.00	92.00
Uniform	1	79.88	64.00	77.00	88.00
Uniform	2	78.00	68.00	78.00	82.00
Adaptive	Matched	**83.08**	**68.00**	**80.00**	**91.00**

Table 3. Real-robot closed-loop evaluation across four manipulation tasks. Each entry reports task success over 50 evaluation trials. Base + 10k continues training on the original demonstrations for the same additional 10k steps used by ReWorld. Gain is the absolute improvement of ReWorld over Base in percentage points.

Task	Base	Base + 10k	ReWorld	Gain
Object sorting	60	60	**68**	+8
Shampoo bottle handover	90	88	**96**	+6
Battery insertion	58	64	**76**	+18
Bowls and cups stacking	30	28	**34**	+4
Average	59.5	60.0	**68.5**	+9.0

and supervision reaches the highest score. Recoverability determines which deviations deserve correction, while task competence determines where probing and supervision should be concentrated.

4.4 Generalization and Physical Transfer

Recovery transfers beyond the training magnitudes. Figure 3(c) tests whether learned recovery is tied to the exact perturbation magnitudes used during corrective synthesis. Training uses only 2 cm and 6 cm translations at a fixed pre-grasp stage and direction, while 4, 8, and 10 cm are held out. ReWorld reaches 67%, 43%, and 28% success on these held-out offsets compared with 47%, 19%, and 10% for Base, producing gains of 20, 24, and 18 points. Performance declines as deviations grow, so the experiment supports interpolation and bounded extrapolation in magnitude under the tested stage and direction rather than generalization to unseen types, directions, or contact conditions.

Physical closed-loop transfer. We finally evaluate transfer to a real robot on four tasks, object sorting, shampoo bottle handover, battery insertion, and bowls and cups stacking. Each task uses 50 evaluation trials. The Base policy is trained on 200 demonstrations per task for five epochs, corresponding to 40k optimization steps across the tasks. ReWorld starts from this checkpoint and trains for 10k additional steps using corrective data synthesized offline from the real-robot demonstrations with a frozen GE-Sim model trained on Tianji-domain data and a motion planner. No corrective trajectories are collected through physical robot interaction. To isolate the effect of the added corrective data from additional optimization, Base + 10k continues training on the original demonstrations for the same 10k steps. All three policies use the same deployment interface.

Table 3 reports the results, while Figure 4 visualizes the physical evaluation and the task-wise gains of ReWorld over Base and Base + 10k. ReWorld improves over Base by 8, 6, 18, and 4 percentage points on the four tasks, respectively. The Base + 10k control changes the results by 0, -2, $+6$, and -2 points, showing that extra optimization alone does not explain the gains. ReWorld also exceeds this control by 8, 8, 12, and 6 points. The largest improvement occurs on battery insertion, where recovery from alignment errors is especially important, while the lower gain on stacking is consistent with the longer multi-object sequence and its accumulated contact sensitivity.

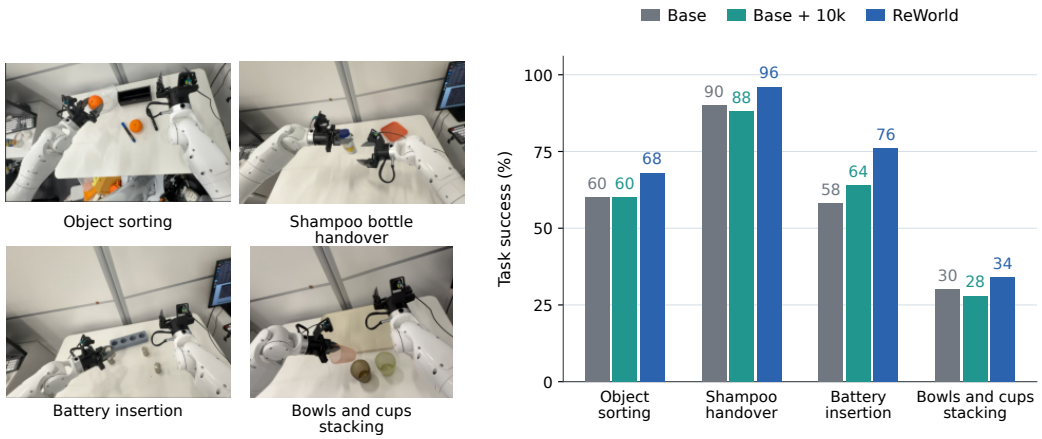

Figure 4. Real-robot evaluation across four manipulation tasks. The left side shows representative deployment scenes, while the right side compares success over 50 trials per task for Base, Base + 10k, and ReWorld. The bar chart highlights consistent task-wise improvements from ReWorld beyond both the original policy and additional training on nominal data.

5 Discussion

ReWorld is motivated by a basic imbalance in demonstration-based robot learning. Successful trajectories provide abundant supervision for nominal execution, while recovery states are rare and costly to collect even though they often determine whether small errors remain recoverable or grow into failure. Repeatedly inducing such failures on physical robots is difficult to scale, and requiring a complete task simulator limits applicability. This motivates using a learned world model for a narrower role. Our results suggest that useful recovery supervision can already be obtained from local counterfactual interactions around successful behavior, without requiring full task simulation.

The quality of corrective supervision is still limited by the underlying ACWM. Inaccurate predictions of geometry, contact, or object state may produce corrective observations that do not faithfully match the intended perturbations, especially farther from the demonstration distribution. Future work should improve ACWM adaptation and filtering for corrective data generation. Our current perturbations also focus on end-effector translation, rotation, and gripper actions, leaving richer contact disturbances, object-state errors, and broader physical-robot evaluation for future study.

6 Conclusion

We presented ReWorld, which uses local world-model counterfactuals to supply recovery supervision missing from successful demonstrations. ReWorld identifies policy-specific recovery gaps, pairs predicted off-nominal observations with corrective actions from planning and demonstration reuse, and allocates supervision according to task competence. Across 50 RoboTwin 2.0 tasks, this adaptive corrective experience improves both SmolVLA and $\pi_{0.5}$, while controlled studies validate recovery-based selection, probe difficulty, competence-aware allocation, and transfer to held-out perturbation magnitudes. The world model and planner are removed after training, leaving the original policy architecture unchanged. These results support a focused use of world models in which short, policy-adaptive counterfactual experience supplies missing recovery supervision without requiring long-horizon world-model rollout or an additional deployment-time recovery system.

AI Use Statement

Generative AI tools were used to assist with language editing and improving the readability of the manuscript. They were not used to generate experimental results, formulate the proposed method, or interpret the empirical findings. All AI-assisted text was reviewed and edited by the authors. The authors take full responsibility for the content of this paper.

REFERENCES

Christopher Agia, Rohan Sinha, Jingyun Yang, Rika Antonova, Marco Pavone, Haruki Nishimura, Masha Itkina, and Jeannette Bohg. CUPID: Curating data your robot loves with influence functions. *arXiv preprint arXiv:2506.19121*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

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

Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. RACER: Rich language-guided failure recovery policies for imitation learning. In *2025 IEEE international conference on robotics and automation (ICRA)*, pp. 15657–15664. IEEE, 2025.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026.

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

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. DreamGen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, and Guanghui Ren. EnerVerse-AC: Envisioning embodied environments with action condition. *arXiv preprint arXiv:2505.09723*, 2025.

Liyiming Ke, Yunchu Zhang, Abhay Deshpande, Siddhartha Srinivasa, and Abhishek Gupta. CCIL: Continuity-based data augmentation for corrective imitation learning. In *International Conference on Learning Representations*, volume 2024, pp. 35813–35831, 2024.

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

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. DART: Noise injection for robust imitation learning. In *Conference on robot learning*, pp. 143–156. PMLR, 2017.

Dayou Li, Jiuzhou Lei, Hao Wang, Lulin Liu, Yunhao Yang, Zihan Wang, Bangya Liu, Minghui Zheng, and Zhiwen Fan. Learning actionable manipulation recovery via counterfactual failure synthesis. *arXiv preprint arXiv:2603.13528*, 2026a.

Xinqing Li, Xin He, Le Zhang, Min Wu, Xiaoli Li, and Yun Liu. A comprehensive survey on world models for embodied AI. *arXiv preprint arXiv:2510.16732*, 2025.

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*, 2026b.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan Luo, Liliang Chen, et al. Genie Envisioner: A unified world foundation platform for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 88446–88463, 2026.

Xiaokang Liu, Zechen Bai, Hai Ci, Kevin Yuchen Ma, and Mike Zheng Shou. World-VLA-Loop: Closed-loop learning of video world model and VLA policy. *arXiv preprint arXiv:2602.06508*, 2026.

Weijia Liufu, Xiaoyu Guo, Ruiyi Chen, Jingzhi Liu, Kaidong Zhang, Xiwen Liang, Jianqi Lin, Dawei Sun, Yuze Wang, Rongtao Xu, et al. RePO-VLA: Recovery-driven policy optimization for vision-language-action models. *arXiv preprint arXiv:2605.09410*, 2026.

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. *arXiv preprint arXiv:2310.17596*, 2023.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, et al. $\pi^*_{0.6}$: A VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a.

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

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. cuRobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. DemoGen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

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

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9834–9844. IEEE, 2025.

SUPPLEMENTARY MATERIAL

This supplement provides the formal offline construction argument and the implementation and evaluation details referenced by the main paper. It also reports an additional experiment that replaces GE-Sim with EVAC to test whether the corrective synthesis pipeline transfers across world models.

A OFFLINE CONSTRUCTION EQUIVALENCE

The main paper constructs corrective samples before policy post-training rather than invoking the world model and motion planner inside every optimization step. We formalize why these two schedules target the same training distribution when the generation components and the base policy used for probing are frozen.

The recoverability table $\mathcal{R}$ and task-competence prior are computed once before corrective-data construction and remain fixed throughout post-training. Conditioned on these fixed quantities, let $\xi \sim q(\cdot \mid \mathcal{R})$ denote a raw generation configuration. It contains the task, source demonstration, manipulation phase, selected timestep, retained perturbation pattern, and every random choice used during candidate construction. Let the frozen generation components be the world model W_ϕ, motion planner $\mathcal{P}$, and base policy π_{θ_0}. Their complete candidate-generation procedure is written as

$$\mathcal{G}(\xi) = \mathcal{G}(W_\phi, \mathcal{P}, \pi_{\theta_0}, \mathcal{R}, \xi). \tag{8}$$

The procedure instantiates a candidate from $\mathcal{R}$, predicts the perturbed observation, rechecks the local recovery criterion for that candidate, constructs a corrective action for retained cases, and applies the same validity checks used during data construction. Let $I(\xi) \in \{0, 1\}$ indicate whether the candidate is retained after these checks. For an accepted configuration, the generated training tuple is

$$z(\xi) = \left(\hat{o}^\delta, s^\delta, \ell, A^C\right). \tag{9}$$

The candidate-level rejection rule induces a distribution over accepted configurations

$$p_C(\xi \mid \mathcal{R}) = \frac{q(\xi \mid \mathcal{R}) I(\xi)}{\mathbb{E}_{\xi' \sim q(\cdot \mid \mathcal{R})}[I(\xi')]}. \tag{10}$$

An online version of the same construction would sample an accepted configuration and immediately optimize the policy on the generated tuple. For policy parameters θ and the policy's native action-learning loss ℓ_θ, its population objective is

$$\mathcal{L}_{\text{online}}(\theta) = \mathbb{E}_{\xi \sim p_C(\cdot \mid \mathcal{R})}\left[\ell_\theta\big(z(\xi)\big)\right]. \tag{11}$$

Offline construction instead draws M accepted configurations independently from the same distribution, stores $z_i = z(\xi_i)$, and minimizes

$$\widehat{\mathcal{L}}_{\text{off}}(\theta) = \frac{1}{M} \sum_{i=1}^{M} \ell_\theta(z_i), \qquad \xi_i \overset{\text{i.i.d.}}{\sim} p_C(\cdot \mid \mathcal{R}). \tag{12}$$

Taking expectation over the random cache gives

$$\mathbb{E}\left[\widehat{\mathcal{L}}_{\text{off}}(\theta)\right] = \mathcal{L}_{\text{online}}(\theta). \tag{13}$$

Thus the offline objective is an unbiased Monte Carlo estimator of the objective obtained by invoking the same frozen generation pipeline during optimization. Under independent sampling and finite expected loss, the strong law of large numbers further gives

$$\widehat{\mathcal{L}}_{\text{off}}(\theta) \xrightarrow[M \to \infty]{\text{a.s.}} \mathcal{L}_{\text{online}}(\theta). \tag{14}$$

The equivalence is distributional rather than sample-wise. A finite cache does not reproduce every sample that could be generated online. It instead contains independent realizations from the same accepted corrective distribution conditioned on the fixed table $\mathcal{R}$. This argument relies on the generation pipeline and selection state being fixed during post-training, which holds in ReWorld because W_ϕ, $\mathcal{P}$, π_{θ_0}, $\mathcal{R}$, and the competence prior are not updated while constructing or consuming corrective data.

Table 4. Key corrective-data construction settings used by ReWorld.

Setting	Value
GE-Sim adaptation	20k steps
Translation directions	$\pm x, \pm y, \pm z$
Rotation directions	$\pm x, \pm y, \pm z$ axes
Maximum rotation magnitude	15°
Premature-close command	≤ 0.10
Nominal open reference	≥ 0.90
Probes per perturbation unit N_u	4
Retention threshold	$\hat{\rho}_u \leq 0.25$
Position recovery threshold	$0.03\,\mathrm{m}$
Rotation recovery threshold	10°
Gripper reopening threshold	≥ 0.8
Expert-window start	$t + H$
Maximum expert-window extension	16 demonstration steps
$B_{\min}$	48
ΔB	96

B Implementation Details

B.1 World model adaptation

The main experiments use GE-Sim as the action-conditioned world model. We initialize GE-Sim from its official checkpoint and adapt it to the RoboTwin 2.0 robot and visual domain for 20k optimization steps. The adaptation uses the same RoboTwin video and training domain for both policy backbones. The resulting checkpoint is frozen during probing and corrective-data construction and is never queried during policy inference.

For the physical experiments, we use a separate GE-Sim model trained on real-robot data collected with the Tianji platform. This Tianji-domain world model is frozen before corrective-data construction and is likewise never queried during policy inference.

B.2 Structured perturbations and recovery probing

Translation perturbations are defined along the six Cartesian directions $\pm x$, $\pm y$, and $\pm z$. Rotation perturbations use positive and negative rotations around the x, y, and z axes, with a maximum rotation magnitude of 15°. Premature gripper closure sets the affected gripper to a closed command with value at most 0.10, while the nominal open reference is at least 0.90.

Each perturbation unit is evaluated on $N_u = 4$ demonstration segments. A unit is retained when its empirical recovery rate satisfies $\hat{\rho}_u \leq 0.25$. For end-effector translation and rotation, the endpoint of the base policy's predicted action chunk is compared with the successful demonstration using forward kinematics. Position recovery uses a threshold of $0.03\,\mathrm{m}$ and rotation recovery uses a quaternion geodesic threshold of 10°. The expert reference window begins at $t + H$ and extends forward by at most 16 demonstration steps. For premature gripper closure, a response is recoverable when the predicted chunk reopens the affected gripper to at least 0.8.

The multitask probing budget is

$$B_i = \mathrm{round}\left(B_{\min} + \Delta B(1 - p_i)\right), \tag{15}$$

with $B_{\min} = 48$ and $\Delta B = 96$. Task competence therefore controls where probes are spent, while the local recovery rate controls which perturbation units are retained.

For pose deviations, CuRobo (Sundaralingam et al., 2023) plans a recovery prefix from the perturbed robot state toward the source demonstration state. The corresponding expert continuation is appended and aligned to the native action horizon. For premature closure, the recovery prefix reopens the affected gripper while holding the arm pose fixed before appending the expert continuation. Intermediate world-model frames are not expanded into additional policy-training samples.

B.3 COMPETENCE-AWARE POST-TRAINING

For task i, the main paper defines a target corrective share c_i and odds $\omega_i = c_i/(1 - c_i)$. Nominal tasks are sampled uniformly as $P_N(i) = 1/K$, while corrective tasks use

$$P_C(i) = \frac{\omega_i}{\sum_{j=1}^{K} \omega_j}. \tag{16}$$

Let each update use b_N nominal samples and b_C corrective samples. A single global corrective coefficient can realize the target share for every task by choosing

$$\lambda_C = \frac{b_N}{K b_C} \sum_{j=1}^{K} \omega_j. \tag{17}$$

The expected unnormalized nominal and corrective coefficients for task i are b_N/K and $\lambda_C b_C P_C(i)$. Their corrective share is therefore

$$\frac{\lambda_C b_C P_C(i)}{b_N/K + \lambda_C b_C P_C(i)} = \frac{\omega_i}{1 + \omega_i} \tag{18}$$

$$= c_i. \tag{19}$$

This equality concerns the expected loss coefficients rather than realized gradient magnitudes. The competence prior, task-sampling distribution, and global corrective coefficient remain fixed throughout post-training. Both nominal and corrective samples use the native action-learning objective of the corresponding policy backbone.

C EXPERIMENTAL DETAILS

C.1 BASELINE IMPLEMENTATION AND COMPARISON FAIRNESS

CCIL, CUPID, Ctrl-World, and RACER are reproduced by adapting their official implementations and released procedures to the same RoboTwin 2.0 setting used by ReWorld. Every comparison starts from the same backbone-specific Base checkpoint used in the main experiment. Additional optimization steps are matched to ReWorld, and data budgets are kept comparable according to the requirements of each method. Policy architecture, task demonstrations, and the final evaluation protocol are otherwise unchanged unless the baseline itself requires an additional component. RACER uses Qwen3-VL-8B as its online vision-language supervisor, while ReWorld retains the original policy interface at deployment.

C.2 REAL-ROBOT SETUP AND SUCCESS CRITERIA

The physical experiments use a Tianji dual-arm robot with two 7-DoF arms and two independent 1-DoF grippers. Gripper commands use the range from 0 to 100. Visual observations are provided by three RGB cameras placed at the head, left hand, and right hand, each operating at 640×480 resolution. The robot is controlled at 30 Hz.

Each of the four tasks uses 200 physical demonstrations. The Base policy is trained for five epochs, corresponding to 40k optimization steps across the four-task multitask dataset. ReWorld starts from this checkpoint and is post-trained for 10k additional steps with task-dependent corrective data. The Base + 10k control continues training on the same nominal demonstrations for the same number of additional steps. Each task is evaluated over 50 physical trials.

D GENERALIZATION ACROSS WORLD MODELS

The main experiments use GE-Sim for corrective observation synthesis. We additionally replace GE-Sim with EVAC (Jiang et al., 2025) to examine whether the ReWorld pipeline remains effective with a different world model. EVAC is initialized from its official checkpoint and adapted to RoboTwin 2.0 for 40k optimization steps. ACT and SmolVLA Base policies are each adapted on RoboTwin 2.0 for 20k steps. We then apply the same ReWorld probing, corrective-action construction, and

Table 5. Success criteria for the real-robot evaluation.

Task	Success criterion
Object Sorting	All objects are placed in their designated target regions.
Shampoo Bottle Handover	The bottle is transferred from the left arm to the right arm and placed on the target plate.
Battery Insertion	All four batteries are inserted into their designated slots.
Bowls and Cups Stacking	All three bowls and three cups are correctly stacked at their target locations.

Table 6. Generalization to EVAC on five RoboTwin 2.0 tasks. Entries report task success rates in percent.

Method	Open laptop	Burger fries	Handover block	Dual bottles	Bottles dustbin	Avg.
ACT Base	70	65	60	**56**	27	55.6
ACT + ReWorld	**85**	**70**	**83**	53	**47**	**67.6**
SmolVLA Base	86	68	24	78	**54**	62.0
SmolVLA + ReWorld	**88**	**86**	**62**	**84**	46	**73.2**

post-training pipeline with EVAC providing the perturbed observations. The table reports closed-loop task success under the same evaluation procedure for Base and ReWorld within each backbone.

ReWorld improves ACT from 55.6% to 67.6% and SmolVLA from 62.0% to 73.2%, corresponding to gains of 12.0 and 11.2 percentage points. Both policy backbones improve on four of the five tasks. These results support that the benefit of targeted corrective synthesis transfers beyond the GE-Sim world model used in the main experiments.