# WHEN LEARNED REWARDS MISLEAD ROBOTIC REINFORCEMENT LEARNING: DIAGNOSING AND CALIBRATING REWARDS UNDER DISTRIBUTION SHIFT

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Dense rewards provide informative feedback for robotic reinforcement learning (RL), but their reliability can deteriorate once online policy optimization drives the state visitation beyond the reward model's training distribution. In this work, we first diagnose this failure mode and reveal that online RL induces substantial out-of-distribution (OOD) state visitation, where reward prediction errors are markedly larger than on in-distribution states and can misguide policy optimization. Our key insight is to calibrate rewards according to their estimated reliability, preserving high-confidence predictions while suppressing unreliable ones. We therefore propose ***Confidence-Aware Reward Calibration (CRC)***, which estimates reward confidence from feature-space distances to the reward model's training data and calibrates low-confidence predictions accordingly. In this way, CRC treats the reward model as a frozen black box and can be seamlessly integrated with different learning-based reward models and RL algorithms without retraining it. Across diverse reward models and RL algorithms, CRC consistently improves online policy learning in both simulation and real-world settings, demonstrating its effectiveness and generality under distribution shift. Code and videos are publicly available at: `https://crc.k7m2.workers.dev/`.

## 1 INTRODUCTION

Reinforcement learning (RL) enables robots to acquire and refine manipulation skills through interaction with the environment, allowing pretrained policies to improve beyond their initial demonstrations and adapt to task objectives (Nair et al., 2018; Ren et al., 2025; Physical Intelligence et al., 2025). Sparse terminal rewards provide a simple supervision signal by indicating task completion, but their limited feedback often leads to inefficient exploration and policy optimization, particularly in long-horizon manipulation tasks (Ball et al., 2023; Escoriza et al., 2025; Mao et al., 2025). To alleviate this limitation, recent work has explored learning-based reward models that map high-dimensional observations to dense, task-relevant feedback (Christiano et al., 2017; Lee et al., 2021; Ma et al., 2023b;a; Zhang et al., 2026). Such models can be trained from diverse supervision, including expert demonstrations, human preferences, and task annotations, while stage- or process-aware variants further estimate intermediate progress throughout task execution (Zhang et al., 2025; Tan et al., 2026). By providing dense, step-wise feedback, learned rewards can substantially improve credit assignment and facilitate efficient online policy optimization.

Despite their advantages, learning-based reward models can become unreliable during online RL due to policy-induced distribution shift. The reward model is typically trained on a fixed offline dataset and remains frozen during policy optimization, while the evolving policy may visit states that are poorly represented in its training data (Yue et al., 2023; Fluri et al., 2024; Xu et al., 2026). We investigate this issue on four Meta-World tasks: `door-lock`, `handle-press-side`, `door-unlock`, and `sweep-into` (Yu et al., 2020). As seen in Fig. 1 (**a**,**b**), online rollouts frequently leave the reward model's training distribution, with OOD states accounting for 68.3% of visited states on average. Fig. 1 (**c**) further shows that reward prediction errors generally increase as states move farther from the training data in feature space. To examine whether these errors directly affect policy learning, we perform an oracle calibration study in which learned rewards are replaced, on ID states only, OOD states only, or both, by rewards computed from the simulator's handcrafted task rules,



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.

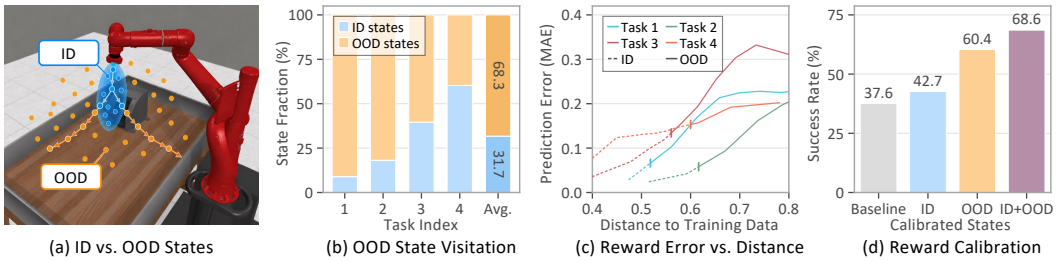



Figure 1: **Motivations. (a)** In-distribution (ID) vs. out-of-distribution (OOD) states during online RL. ID states remain close to the training distribution, whereas OOD states extend beyond it. **(b)** During policy optimization, OOD states account for **68.3**% of visited states on average across four Meta-World tasks. **(c)** Reward prediction error increases with feature-space distance to the training data. **(d)** OOD reward calibration yields substantially greater gains than ID reward calibration.

denoted as the rule-based reward model (RM-rule, see §5.1). As shown in Fig. 1 (**d**), calibrating OOD rewards yields substantially larger gains than correcting ID rewards and nearly matches the improvement obtained by correcting both groups. This suggests that unreliable reward predictions on OOD states are a key source of misguidance during online policy optimization, motivating the following question:

> *How can we identify and calibrate unreliable rewards on OOD states during online reinforcement learning without retraining the reward model?*

Our answer to the above question in this work is **Confidence-Aware Reward Calibration (CRC)**, a *training-free* and *plug-and-play* approach for calibrating unreliable rewards under policy-induced distribution shift. Specifically, CRC consists of two complementary components: *Reliability-based Reward Reweighting* and *Neighbor-guided Reward Compensation*. The former estimates the reliability of each reward prediction from its feature-space deviation relative to the reward model's training data, preserving high-confidence predictions while progressively downweighting unreliable ones. The latter leverages neighboring training samples to compensate low-confidence rewards using local reward evidence, while accounting for uncertainty through label disagreement. Notably, CRC reuses the existing reward-model training data and operates directly on reward outputs, requiring neither additional trajectory collection nor reward-model retraining. It can be seamlessly integrated with different learning-based reward models and RL algorithms. Across three reward models and two RL algorithms, CRC consistently improves online policy learning, achieving absolute gains of **18.7**% on Meta-World (Yu et al., 2020) and **11.7**% on Robomimic (Mandlekar et al., 2022). On three real-world tasks, CRC further improves the average success rate from **77.9**% to **92.9**%.

Our contributions are threefold. **1)** We identify reward unreliability under policy-induced distribution shift as a key failure mode in online robotic RL, and show that errors on OOD states can significantly hinder policy optimization. **2)** We propose Confidence-Aware Reward Calibration (CRC), a training-free and plug-and-play approach that estimates state-wise reward reliability from local feature-space structure and adaptively calibrates unreliable predictions without additional trajectory collection or reward-model retraining. **3)** We validate CRC across multiple reward models and RL algorithms in both simulation and real-world manipulation, demonstrating consistent improvements in task success and online policy learning.

## 2 Related Work

**Online Reinforcement Learning for Robotics.** Online RL enables robots to improve manipulation policies through continued interaction with the environment (Kalashnikov et al., 2018; Lee et al., 2022; Rafailov et al., 2023). To improve data efficiency, demonstration-augmented and offline-to-online methods leverage prior experience for policy initialization and subsequent online refinement (Rajeswaran et al., 2018; Nair et al., 2020). RLPD (Ball et al., 2023) mixes offline and online transitions during off-policy learning, while IBRL (Hu et al., 2024) uses imitation-policy actions to guide exploration and critic bootstrapping. Other works focus on stabilizing the offline-to-online transition, including calibrated value learning in Cal-QL (Nakamoto et al., 2023) and policy warmup

in WSRL (Zhou et al., 2025). For physical robots, ARIEL enables autonomous fine-tuning from prior multi-task data (Walke et al., 2023), while FARL incorporates safety-aware recovery to reduce costly failures (Li et al., 2026). These methods mainly improve policy initialization, value estimation, exploration, or safety during online adaptation; in contrast, our work focuses on the reliability of learned reward signals under policy-induced distribution shift.

**Reward Learning for Robotic RL.** Dense reward signals can improve exploration and credit assignment in robotic RL, but handcrafted rewards require substantial task-specific engineering and scale poorly to complex tasks (Tsao et al., 2026; Xie et al., 2024; Ma et al., 2024). Recent work therefore derives rewards from pretrained visual or vision-language models (Ma et al., 2023b;a; Sontakke et al., 2023; Rocamonde et al., 2024), or learns task-specific reward models from demonstrations, preferences, and progress annotations. ReWiND (Zhang et al., 2025) learns language-conditioned progress rewards for policy adaptation, RoboReward (Lee et al., 2026) develops large-scale vision-language reward models for robotic tasks, and Robometer (Liang et al., 2026) combines frame-level progress supervision with trajectory-level preferences. Recently, Robo-Dopamine 2.0 (Xu et al., 2026) introduces history-conditioned and OOD-aware supervision to improve process reward modeling, but requires dedicated OOD data during reward-model training. Despite these advances, learned rewards may still become unreliable when online policies visit states insufficiently supported by the reward model's training data (Fluri et al., 2024). Unlike methods that improve reward-model training itself, CRC performs training-free calibration of frozen reward models using their existing training data, without collecting additional trajectories or modifying the underlying RL algorithm.

## 3 Preliminaries and Motivation

In this section, we first formulate online robotic RL with learned rewards and then investigate how policy-induced distribution shift leads to unreliable reward predictions.

### 3.1 Preliminaries

**MDP Modeling.** We consider a robotic manipulation task specified by an instruction $\ell$. Its underlying environment is modeled as a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \rho_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $\mathcal{P}(\cdot \mid s, a)$ is the transition dynamics, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the task reward function, $\rho_0$ is the initial-state distribution, and $\gamma \in [0, 1)$ is the discount factor. At time $t$, the robot receives an observation $o_t$ of the underlying state $s_t$. Conditioned on $o_t$ and $\ell$, the policy selects an action $a_t \sim \pi_\theta(\cdot \mid o_t, \ell)$. The environment then transitions to $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$, and the robot receives a scalar reward $r_t = r(s_t, a_t)$. For an episode of horizon $L$, the policy aims to maximize the expected discounted return:

$$\theta^* = \arg\max_\theta \mathbb{E}_{\xi \sim p_\theta(\cdot|\ell)} \left[ \sum_{t=0}^{L-1} \gamma^t r(s_t, a_t) \right], \tag{1}$$

where $\xi$ denotes an interaction trajectory and $p_\theta(\cdot \mid \ell)$ is the trajectory distribution induced by the initial-state distribution $\rho_0$, the policy $\pi_\theta$, and the environment dynamics $\mathcal{P}$.

**Problem Setup.** We study online robotic RL guided by a reward model $\mathcal{R}_\phi$ trained on a fixed offline dataset $\mathcal{D}_{\mathrm{RM}}$ with task-related supervision. At each step, the model takes the observation $o_t$ and task instruction $\ell$ as the joint input $x_t = (o_t, \ell)$ and outputs a scalar reward score $\mathcal{R}_\phi(x_t)$. The frozen reward model may produce unreliable predictions as the online policy moves beyond its training distribution. We therefore calibrate its predictions using hidden representations and offline training data $\mathcal{D}_{\mathrm{RM}}$, requiring neither reward-model retraining nor additional trajectories.

### 3.2 Motivation: Unreliable OOD Rewards Hinder Policy Learning

Learning-based reward models provide dense supervision for policy optimization, but their reliability may degrade as the evolving policy shifts the visited-state distribution beyond the reward model's offline training support. In this part, we scrutinize whether such distribution shift causes unreliable reward predictions that hinder policy learning. Our diagnosis uses four Meta-World tasks: `door-lock`, `handle-press-side`, `door-unlock`, and `sweep-into`. We evaluate

the RM-rule (§ 5.1) reward model, trained on offline trajectories and frozen during online optimization, and compare its predictions with oracle rewards to quantify errors under distribution shift.

**OOD Scoring and Identification.** To investigate whether reward prediction errors are associated with distribution shift, we first measure how closely online states resemble those in the reward model's offline training data. Prior work has demonstrated that normalized deep representations support effective nearest-neighbor OOD detection across different network architectures and training objectives (Sun et al., 2022; Papernot & McDaniel, 2018). Motivated by these findings, we directly reuse the reward model's hidden representations to measure the similarity between online states and offline training states, without training an additional encoder or OOD detector. This allows CRC to treat the reward model as a frozen black box.

Given the input $x_t = (o_t, \ell)$, we extract the hidden feature $h_t = h_\phi(x_t) \in \mathbb{R}^d$ and normalize it as $\bar{h}_t = h_t / \max(\|h_t\|_2, \epsilon)$, where $\epsilon > 0$ ensures numerical stability. For each task, we apply the same procedure to the offline training dataset $\mathcal{D}_{\mathrm{RM}}$ to construct a reference bank $\mathcal{B}_{\mathrm{ref}} = \{(\bar{h}_j, y_j)\}_{j=1}^{N_{\mathrm{ref}}}$, where $N_{\mathrm{ref}}$ is the number of reference samples and $y_j$ is the scalar reference value associated with $\bar{h}_j$. We measure an online state's deviation from the reference data by averaging the Euclidean distances to its $k$ nearest reference representations:

$$S_{\mathrm{OOD}}(\bar{h}_t) = \frac{1}{k} \sum_{j \in \mathcal{N}_k(\bar{h}_t)} \|\bar{h}_t - \bar{h}_j\|_2, \tag{2}$$

where $\mathcal{N}_k(\bar{h}_t)$ denotes the indices of the $k$ nearest features in $\mathcal{B}_{\mathrm{ref}}$, with $1 \le k \le N_{\mathrm{ref}}$. A larger score indicates that the online state is farther from nearby offline states in the chosen feature space. For each task, we hold out a subset of offline samples for threshold estimation and construct $\mathcal{B}_{\mathrm{test}}$ from the remaining samples. We compute the held-out samples' OOD scores against $\mathcal{B}_{\mathrm{test}}$ and set the threshold to their $q$-quantile, $\tau_q = \mathrm{Quantile}_q\left(\left\{S_{\mathrm{OOD}}(\bar{h}_i)\right\}_{i=1}^{N_{\mathrm{cal}}}\right)$, where $q = 0.90$. An online state is classified as OOD if its score exceeds $\tau_q$, and as ID otherwise:

$$g(x_t) = \begin{cases} \mathrm{ID}, & S_{\mathrm{OOD}}(\bar{h}_t) \le \tau_q, \\ \mathrm{OOD}, & S_{\mathrm{OOD}}(\bar{h}_t) > \tau_q. \end{cases} \tag{3}$$

**Diagnostic Findings.** We conduct the diagnostic study on four Meta-World tasks using the RM-rule reward model, which is trained on offline data and kept frozen during online policy optimization. Online trajectories are partitioned into ID and OOD states using the feature-space scoring procedure above. Reward prediction error is measured as the mean absolute error (MAE) between RM-rule predictions and simulator reference rewards, after normalizing both to the same scale. As can be observed from Fig. 1 **(b)**, online policy optimization frequently drives state visitation beyond the reward model's offline support, with OOD states accounting for 68.3% of visited observations on average across the four tasks. Fig. 1 **(c)** further shows that reward prediction errors are substantially larger on OOD states than on ID states and generally increase with feature-space distance from the training data. These results indicate that policy-induced distribution shift is closely associated with degraded reward reliability. To determine whether such errors directly affect policy learning, we further perform an oracle intervention study. Specifically, we replace RM-rule predictions with simulator rewards on ID states only, OOD states only, or both, while leaving all other predictions unchanged. As shown in Fig. 1 **(d)**, correcting OOD rewards yields substantially larger success-rate gains than correcting ID rewards and nearly matches the performance obtained when both groups are corrected. Learning curves are provided in Fig. S4. Together, these findings identify **unreliable reward predictions on OOD states as a key source of error in online policy optimization**. This motivates a training-free method for selectively calibrating unreliable rewards without collecting additional trajectories or retraining the reward model.

## 4 Confidence-aware Reward Calibration

In this section, we present CRC, a training-free and plug-and-play approach for calibrating unreliable rewards under distribution shift. As illustrated in Figure 2, CRC combines *Reliability-based Reward Reweighting* to suppress low-confidence predictions on OOD states with *Neighbor-guided Reward Compensation* to preserve task-relevant reward signals.

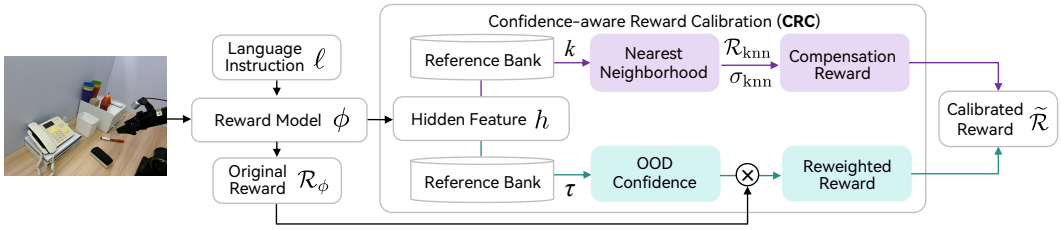



Figure 2: **Illustration of the CRC method.** CRC treats the reward model as a frozen *black box* and requires only its reward outputs and hidden features, without retraining or architectural modification.

### 4.1 Reliability-based Reward Reweighting

In Section 3.2, we reveal that reward prediction errors tend to increase as online states deviate from the offline training data. This motivates reducing the influence of unreliable reward predictions when the policy visits OOD states. We implement this principle through a weighting rule that preserves predictions within the offline distance threshold and progressively downweights those beyond it. Specifically, given the OOD score $S_{\text{OOD}}(\bar{h}_t)$ and the task-specific threshold $\tau_q > 0$, we compute the normalized excess distance $d_t$ and map it to a confidence weight $c_t$:

$$d_t = \max\left(\frac{S_{\text{OOD}}(\bar{h}_t) - \tau_q}{\tau_q}, 0\right), \qquad c_t = \exp(-\alpha d_t), \tag{4}$$

where $\alpha > 0$ controls the confidence decay rate. We then apply this weight to the frozen reward model's output to obtain the reweighted prediction:

$$\mathcal{R}_t^{\text{rw}}(x_t) = c_t\, \mathcal{R}_\phi(x_t). \tag{5}$$

For ID states, $d_t = 0$ and $c_t = 1$, so the original prediction is preserved. For OOD states, $c_t$ decreases smoothly with the normalized excess distance, progressively reducing the contribution of potentially unreliable predictions to the calibrated reward.

### 4.2 Neighbor-guided Reward Compensation

The confidence weight $c_t$ is derived from the feature-space distance between an online state and the offline reference bank $\mathcal{B}_{\text{ref}}$, which is constructed from the reward model's training data. As a result, its reliability estimate inevitably reflects the coverage of the offline dataset. States that are sparsely represented offline may receive low confidence even when they correspond to meaningful task progress during online interaction. In such cases, reward reweighting alone can overly suppress $\mathcal{R}_t^{\text{rw}}(x_t)$ and discard useful learning signals. To alleviate this limitation, CRC augments the reweighted prediction with a local reward reference estimated from neighboring offline samples.

Given the current state $x_t$, we retrieve the $k$ nearest reference features to $\bar{h}_t$ from $\mathcal{B}_{\text{ref}}$, with index set $\mathcal{N}_k(\bar{h}_t)$ and $1 \le k \le N_{\text{ref}}$. $y_j$ denotes the scalar reference value associated with the $j$-th neighbor. To assign greater influence to closer reference samples, we compute distance-based weights as

$$w_{t,j} = \frac{\exp\left(-\|\bar{h}_t - \bar{h}_j\|_2^2\right)}{\sum_{j' \in \mathcal{N}_k(\bar{h}_t)} \exp\left(-\|\bar{h}_t - \bar{h}_{j'}\|_2^2\right)}, \qquad j \in \mathcal{N}_k(\bar{h}_t). \tag{6}$$

We then compute the weighted mean and standard deviation of their reference values:

$$\mathcal{R}_{\text{knn}}(x_t) = \sum_{j \in \mathcal{N}_k(\bar{h}_t)} w_{t,j} y_j, \qquad \sigma_{\text{knn}}(x_t) = \sqrt{\sum_{j \in \mathcal{N}_k(\bar{h}_t)} w_{t,j} \left(y_j - \mathcal{R}_{\text{knn}}(x_t)\right)^2}. \tag{7}$$

The mean $\mathcal{R}_{\text{knn}}(x_t)$ aggregates local reward evidence from neighboring offline samples. However, when the retrieved neighbors correspond to different stages of task progress, their reference values can vary substantially, indicating weak local consensus in the reference set. In this case, $\mathcal{R}_{\text{knn}}(x_t)$ may be an unreliable estimate for the current state. We therefore use the weighted standard deviation to quantify this disagreement and define a conservative compensation term as

$$\mathcal{R}_t^{\text{Comp}}(x_t) = \mathcal{R}_{\text{knn}}(x_t) - \lambda_r \sigma_{\text{knn}}(x_t), \tag{8}$$

where $\lambda_r \geq 0$ controls the penalty for disagreement among neighboring reference values. When neighboring reference values are consistent, the disagreement penalty remains small, allowing the compensation term to recover useful signals suppressed by low confidence. In contrast, larger disagreement increases the penalty and reduces the influence of uncertain local evidence, preventing overly optimistic compensation.

### 4.3 Reward Calibration with CRC

Finally, CRC combines the two components to calibrate unreliable rewards on OOD states:

$$\widetilde{\mathcal{R}}_t = c_t \mathcal{R}_\phi(x_t) + \beta(1 - c_t)\mathcal{R}_t^{\text{Comp}}(x_t), \tag{9}$$

where $\beta \geq 0$ controls the strength of reward compensation. The first term suppresses potentially unreliable reward predictions according to their estimated confidence, while the second term restores task-relevant feedback that may be overly attenuated by reweighting alone. As confidence $c_t$ decreases, the factor $(1 - c_t)$ increases, gradually shifting CRC's reliance from the original reward prediction toward the conservative local reference. Illustrative examples of how the two terms complement each other are provided in § 5.4.

## 5 Experiments

In this section, we conduct extensive experiments in both simulation and real-world environments to demonstrate the effectiveness and generality of our proposed CRC approach. Specifically, we structure our experiments to answer the following questions:

*1) Can CRC improve online policy learning across various reward models and RL algorithms?*
*2) How do the key components and design choices of CRC affect its performance?*
*3) How CRC improves reward reliability under different OOD states?*

We answer the first question through simulation experiments on Meta-World Yu et al. (2020) and Robomimic Mandlekar et al. (2022) in Section 5.1, together with real-world manipulation experiments in Section 5.2. The second question is investigated through component ablations and hyperparameter analyses in Section 5.3. Finally, Section 5.4 provides qualitative analyses of how CRC identifies and calibrates unreliable reward predictions along real-world trajectories.

### 5.1 Simulation Experiments

We first evaluate CRC in simulation on Meta-World and Robomimic, covering multiple manipulation tasks and reward-model formulations. These experiments assess whether CRC can consistently improve online policy learning across different task settings and reward supervision signals.

**Evaluation Tasks.** We evaluate CRC employing tasks from **Meta-World** (Yu et al., 2020) and **Robomimic** (Mandlekar et al., 2022). For Meta-World, we use eight tasks covering diverse object interactions and control patterns: `door-lock`, `handle-press-side`, `door-unlock`, `sweep-into`, `faucet-close`, `hammer`, `lever-pull`, and `button-press-wall`. For Robomimic, we evaluate on the `Can` and `Lift` tasks, which provide scenarios with different interaction requirements. These tasks enable evaluation of CRC across diverse manipulation behaviors.

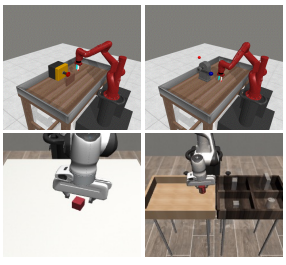


Figure 3: Example tasks of Meta-World (*top*) and Robomimic (*bottom*).

**Reward Models.** To evaluate CRC across reward formulations, we consider three learning-based reward models, each trained with a distinct type of supervision (**Appendix** B.1):

- **Rule-based Reward Model (RM-rule).** RM-rule is trained with dense reward labels generated by handcrafted rules, e.g. gripper-to-object and object-to-goal distances (Yu et al., 2020).
- **Preference-based Reward Model (RM-pre).** RM-pre follows the reward-learning objective of SORS (Memarian et al., 2021) and uses a Bradley–Terry preference model to learn scalar rewards that preserve trajectory rankings induced by observed discounted returns.

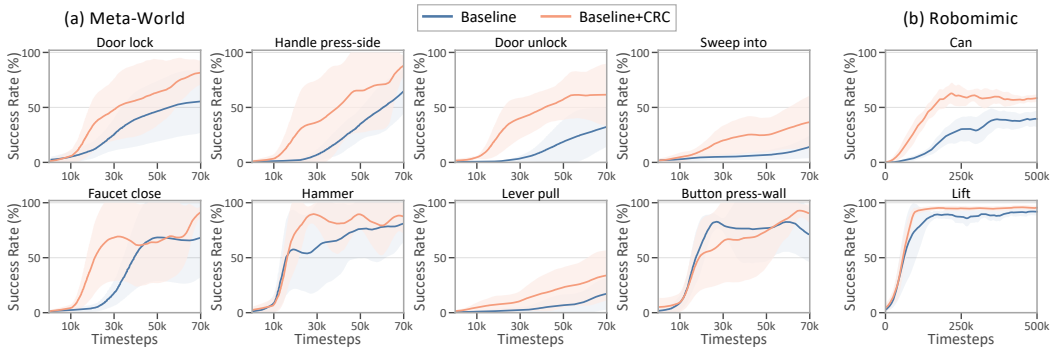



Figure 4: **Performance on Meta-World and Robomimic.** The baseline reward models for Meta-World and Robomimic are RM-rule and RM-pro, respectively. The RL algorithm is RLPD. Curves report mean results over four random seeds. Results on other reward models are in **Appendix** A.

- **Progress-based Reward Model (RM-pro).** RM-pro is trained to predict normalized task progress from successful demonstrations (Zhang et al., 2025), using temporal position as supervision with targets linearly increasing from 0 at the initial state to 1 at task completion.

**Experimental Setup.** We evaluate all three reward models on Meta-World. On Robomimic, we evaluate RM-pro and RM-pre only, as the benchmark does not provide the task-specific geometric supervision required by RM-rule. On Meta-World, we follow the ReWiND (Zhang et al., 2025) data preprocessing and offline-to-online RL protocol. Policies are first pretrained with IQL (Kostrikov et al., 2022) for 10k gradient updates and then fine-tuned online with RLPD Ball et al. (2023). The online stage begins with a critic warm-up, after which policy updates are enabled at 8k environment steps, under a total interaction budget of 70k steps. On Robomimic, following (Hu et al., 2024), we initialize the policy with imitation learning and subsequently perform online fine-tuning with RLPD for 500k steps. Since CRC is a plug-and-play reward calibration method that does not alter the underlying RL algorithm, each baseline and its CRC-augmented counterpart use the same policy initialization, training protocol, interaction budget, evaluation procedure, and metrics. Detailed training configurations and hyperparameter settings are provided in **Appendix** B.2.

**Results and Analysis.** Figure 4 reports the online learning performance of RLPD with and without our CRC across eight Meta-World tasks and two Robomimic tasks. As can be observed, CRC generally achieves higher success rates under the same interaction budget and, on several tasks, reaches comparable performance with fewer environment interactions, demonstrating improvements in both sample efficiency and final policy performance. Figure 5 further summarizes the task-averaged results across different reward models. Averaged over the final 10k environment steps and four random seeds, CRC yields absolute success-rate gains of **18.7**% on Meta-World and **11.7**% on Robomimic, demonstrating its effectiveness across different forms of reward supervision. Detailed results appear in Tables S1 and S2; supplementary learning curves are shown in Figs. S1–S2 (Meta-World) and Fig. S3 (Robomimic).

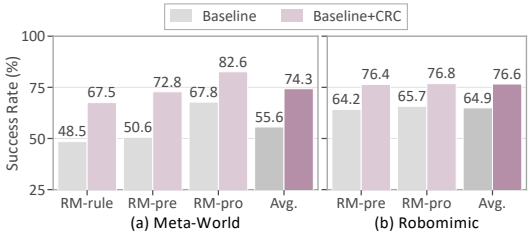



Figure 5: **Average success rates across different reward models.** Results are averaged over tasks and four random seeds.

## 5.2 Real-World Experiments

In this section, we further evaluate CRC on three real-world manipulation tasks to assess its effectiveness under real-world interaction and distribution shift.

**Experimental Setup.** We evaluate CRC on three real-world tasks: `Put Bread in Basket`, `Open Drawer & Store Lemon`, and `Place Stationery in Holder`. Experiments are conducted on an AGILEX PIPER 6-DoF robot arm with a 1-DoF gripper and two RGB cameras providing third-person and wrist views. For each task, we collect 30 demonstrations for policy

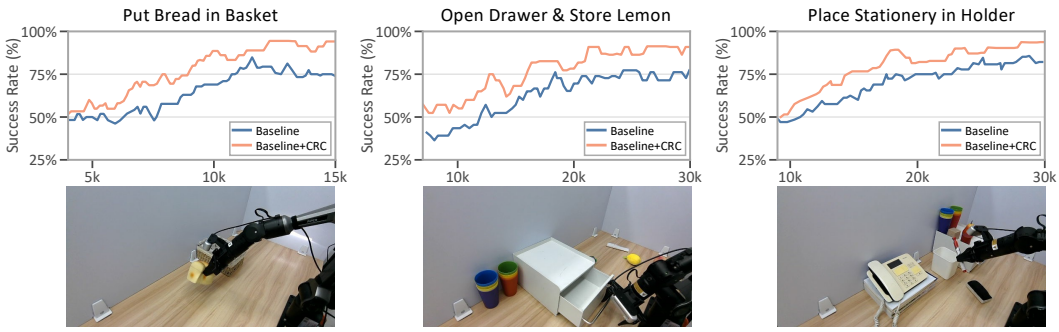



Figure 6: **Performance on real-world manipulation tasks.** The baseline reward model and RL algorithm are RM-pro and SAC, respectively.

initialization, of which 10 are used to train RM-pro from third-person observations and task instructions. The policy is initialized through imitation learning and subsequently fine-tuned with online SAC (Haarnoja et al., 2018), following an implementation adapted from DSRL (Wagenmaker et al., 2025). During online learning, the reward model remains frozen and CRC calibrates its predictions before they are used for policy updates. Additional details are provided in **Appendix** B.3.

**Results and Analysis.** Figure 6 shows the online learning performance on the three real-world manipulation tasks. Across all tasks, CRC consistently improves the learning curves over the uncalibrated baseline, achieving higher success rates under the same interaction budget and reaching comparable performance with fewer online interactions. The gains persist into later stages of training, indicating that CRC improves not only sample efficiency but also the final policy performance. Overall, CRC increases the average success rate from **77.9**% to **92.9**% across the three tasks. These improvements are obtained without retraining the reward model or modifying the underlying RL algorithm, supporting the effectiveness of CRC as a plug-and-play reward calibration module in physical robot learning. Detailed results are reported in Table S3.

### 5.3 Ablation Studies

We conduct ablation studies on four Meta-World tasks (`door-lock`, `handle-press-side`, `door-unlock`, and `sweep-into`) with RM-rule to evaluate the contributions of the CRC components and the sensitivity to hyperparameters. Detailed results are provided in Figs. S5–S7.

**Component Analysis.** We evaluate three variants to isolate the contributions of CRC's components. The *Baseline (Base)* directly uses the original reward prediction. *Reweighting (Rw)* applies Reliability-based Reward Reweighting using the confidence $c_t$ defined in Eq. (4). *Reweighting+Compensation (Rw+Comp)* corresponds to the full CRC formulation in Eq. (9), which further incorporates Neighbor-guided Reward Compensation. As observed in Fig. 7 (**a**), Reward Reweighting improves the average success rate from 37.6% to 58.5%, demonstrating the importance of adaptively suppressing unreliable reward predictions. Adding Reward Compensation further increases performance to 63.5%, showing that local reward evidence provides complementary guidance.

**Effect of $\tau$.** As defined in Eq.(3), the threshold $\tau$ determines whether an online state is regarded as ID or OOD. We vary the threshold quantile $q$ over $\{0.80, 0.90, 0.95\}$, yielding $\tau_{80}$, $\tau_{90}$, and $\tau_{95}$. As shown in Fig. 7 **(b)**, all three settings outperform the baseline, demonstrating that CRC remains effective across different threshold choices. Specifically, the average success rates are 51.2%, 63.5%, and 52.4% for $q = 0.80$, 0.90, and 0.95, respectively, with $q = 0.90$ achieving the best performance.

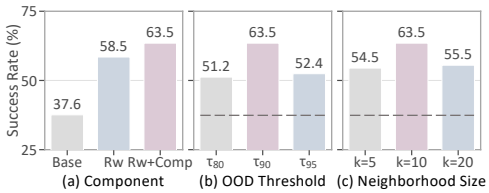



Figure 7: **Ablation studies.** Results are averaged across four Meta-World tasks. The black dotted lines in (b)(c) are the Baseline results.

**Effect of $k$.** The neighborhood size $k$ controls the amount of local evidence used for OOD scoring and reward compensation in CRC. A larger $k$ incorporates broader neighborhood information. We

vary $k$ over $\{5, 10, 20\}$ and report the obtained results in Fig. 7 **(c)**. As observed, the three settings consistently outperform the baseline, with $k = 10$ achieving the best performance.

### 5.4 How CRC improves reward reliability under different OOD states?

We further analyze how CRC calibrates reward reliability using two representative real-world OOD cases in Fig. 8. In the **(Left) Visual-scene OOD** case, an extreme bread placement leads to low confidence around $t = 17$. CRC suppresses the unreliable reward through reweighting, while neighbor-guided compensation restores part of the useful signal because the robot is still moving toward the bread. In the **(Right) Action-state OOD** case, the gripper becomes stuck on the holder edge near task completion. Around $t = 114$, CRC again downweights the low-confidence prediction, while local reward evidence provides positive compensation consistent with task progress. These cases show that CRC improves reward reliability by jointly suppressing unsupported predictions and preserving locally consistent progress signals, rather than simply reducing OOD rewards.

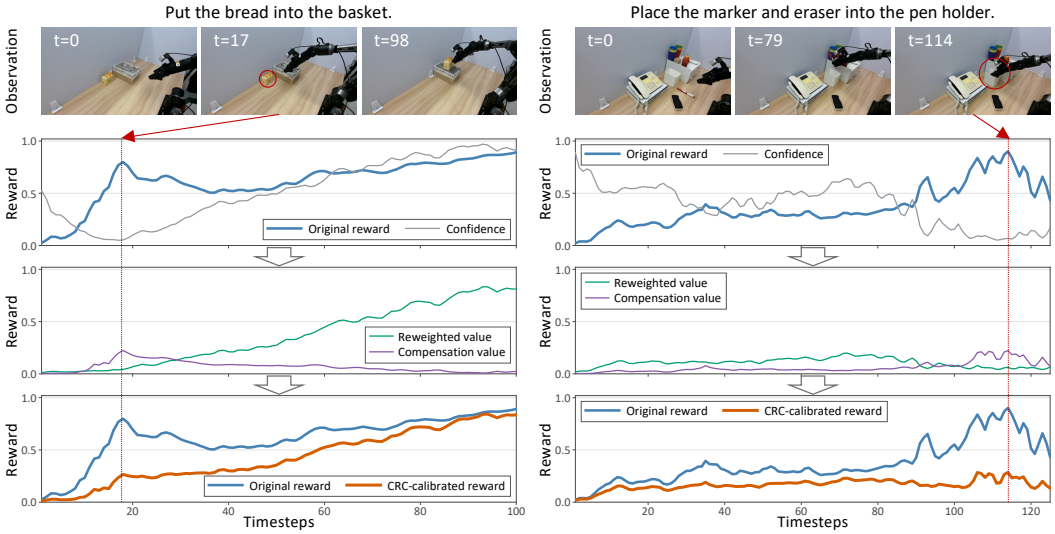



Figure 8: **CRC under different OOD states. (Left) Visual-scene OOD** caused by extreme bread placement. **(Right) Action-state OOD** caused by an abnormal gripper state. CRC assigns low confidence (gray curves) to unreliable reward predictions, suppresses their contribution through reward reweighting (green curves), and restores useful signals via reward compensation (purple curves). Ultimately, CRC calibrates the original reward (blue) into a more reliable reward signal (orange).

## 6 Conclusion, Limitations, and Future Work

In this work, we investigate the reliability of learned dense rewards in online robotic RL under distribution shift and identify unreliable OOD reward predictions as a key source of misguidance during policy optimization. To address this, we introduce CRC, a training-free and plug-and-play method that estimates reward reliability from feature-space deviation and calibrates low-confidence predictions using local reward evidence from neighboring training samples. CRC requires neither reward-model retraining nor additional trajectory collection and can be seamlessly integrated into existing RL pipelines without modifying the underlying actor or critic objectives. Experiments across simulation and real-world tasks demonstrate consistent gains across reward models and RL algorithms, validating the effectiveness and generality of CRC.

Despite these improvements, CRC remains a calibration mechanism rather than a method for recovering ground-truth rewards. Its calibrated output still depends on the original reward prediction and therefore cannot guarantee that erroneous rewards are fully corrected. Future work may explore stronger correction mechanisms or online reward-model adaptation to further reduce this dependence and improve robustness under more severe distribution shifts.

## AI USE STATEMENT

All experiments in this work were conducted by the authors, and all reported results reflect real experimental data. The manuscript was written entirely by the authors. Generative AI tools were used solely as auxiliary tools for grammar checking and language polishing. The authors reviewed the AI-assisted text, verified the reported calculations against the source logs, inspected the compiled output, and take full responsibility for the final content of the manuscript.

## REFERENCES

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient Online Reinforcement Learning with Offline Data. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1577–1594. PMLR, 2023. URL https://proceedings.mlr.press/v202/ball23a.html.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

Adrià López Escoriza, Nicklas Hansen, Stone Tao, Tongzhou Mu, and Hao Su. Multi-Stage Manipulation with Demonstration-Augmented Reward, Policy, and World Model Learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15542–15563. PMLR, 2025. URL https://proceedings.mlr.press/v267/escoriza25a.html.

Lukas Fluri, Leon Lang, Alessandro Abate, Patrick Forré, David Krueger, and Joar Skalse. The perils of optimizing learned reward functions: Low training error does not guarantee low regret. *arXiv preprint arXiv:2406.15753*, 2024. URL https://arxiv.org/abs/2406.15753.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Hengyuan Hu, Suvir Mirchandani, and Dorsa Sadigh. Imitation Bootstrapped Reinforcement Learning. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. doi: 10.15607/RSS.2024.XX.056. URL https://www.roboticsproceedings.org/rss20/p056.html.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. arXiv preprint arXiv:1806.10293, 2018. URL https://arxiv.org/abs/1806.10293.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Kimin Lee, Laura M Smith, and Pieter Abbeel. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6152–6163. PMLR, 2021. URL https://proceedings.mlr.press/v139/lee21i.html.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-Online Reinforcement Learning via Balanced Replay and Pessimistic Q-Ensemble. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1702–1712. PMLR, 2022. URL https://proceedings.mlr.press/v164/lee22d.html.

Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. RoboReward: General-Purpose Vision-Language Reward Models for Robotics. *arXiv preprint arXiv:2601.00675*, 2026. URL https://arxiv.org/abs/2601.00675.

Huanyu Li, Kun Lei, Sheng Zang, Kaizhe Hu, Yongyuan Liang, Bo An, Xiaoli Li, and Huazhe Xu. Failure-Aware RL: Reliable Offline-to-Online Reinforcement Learning with Self-Recovery for Real-World Manipulation. *arXiv preprint arXiv:2601.07821*, 2026. URL https://arxiv.org/abs/2601.07821.

Anthony Liang, Yigit Korkmaz, Jiahui Zhang, Minyoung Hwang, Abrar Anwar, Sidhant Kaushik, Aditya Shah, Alex S. Huang, Luke Zettlemoyer, Dieter Fox, Yu Xiang, Anqi Li, Andreea Bobu, Abhishek Gupta, Stephen Tu, Erdem Biyik, and Jesse Zhang. Robometer: Scaling General-Purpose Robotic Reward Models via Trajectory Comparisons. *arXiv preprint arXiv:2603.02115*, 2026. URL https://arxiv.org/abs/2603.02115.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-Image Representations and Rewards for Robotic Control. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23301–23320. PMLR, 2023a. URL https://proceedings.mlr.press/v202/ma23b.html.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2210.00030.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Jim Fan, and Anima Anandkumar. Eureka: Human-Level Reward Design via Coding Large Language Models. In *International Conference on Learning Representations*, volume 2024, pp. 26516–26560, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/70c26937fbf3d4600b69a129031b66ec-Abstract-Conference.html.

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*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 2022. URL https://proceedings.mlr.press/v164/mandlekar22a.html.

Xiaofeng Mao, Yucheng Xu, Zhaole Sun, Elle Miller, Daniel Layeghi, and Michael Mistry. Learning long-horizon robot manipulation skills via privileged action. *arXiv preprint arXiv:2502.15442*, 2025. URL https://arxiv.org/abs/2502.15442.

Farzan Memarian, Wonjoon Goo, Rudolf Lioutikov, Scott Niekum, and Ufuk Topcu. Self-supervised online reward shaping in sparse-reward environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2369–2375. IEEE, 2021. doi: 10.1109/IROS51168.2021.9636020. URL https://rudolf.intuitive-robots.net/publication/memarian-selfsupervisedonlinereward-2021/.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299. IEEE, 2018. doi: 10.1109/ICRA.2018.8463162. URL https://arxiv.org/abs/1709.10089.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating Online Reinforcement Learning with Offline Datasets. arXiv preprint arXiv:2006.09359, 2020. URL https://arxiv.org/abs/2006.09359.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning. In *Advances in Neural Information Processing Systems*, volume 36,

2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/c44a04289beaf0a7d968a94066a1d696-Abstract-Conference.html.

Nicolas Papernot and Patrick McDaniel. Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning. *arXiv preprint arXiv:1803.04765*, 2018. URL https://arxiv.org/abs/1803.04765.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a VLA That Learns From Experience. *arXiv preprint arXiv:2511.14759*, 2025. URL https://arxiv.org/abs/2511.14759.

Rafael Rafailov, Kyle Beltran Hatch, Victor Kolev, John D. Martin, Mariano Phielipp, and Chelsea Finn. MOTO: Offline Pre-training to Online Fine-tuning for Model-based Robot Learning. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3654–3671. PMLR, 2023. URL https://proceedings.mlr.press/v229/rafailov23a.html.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.049. URL https://www.roboticsproceedings.org/rss14/p49.html.

Allen Ren, Justin Lidard, Lars Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion Policy Policy Optimization. In *International Conference on Learning Representations*, volume 2025, pp. 77288–77329, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c0749c39aaff9e9e4c91f7118bf21b1e-Abstract-Conference.html.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-Language Models are Zero-Shot Reward Models for Reinforcement Learning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/7a7f6cc5dc2a84fb4edf0feb8e5cfd50-Abstract-Conference.html.

Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. RoboCLIP: One Demonstration is Enough to Learn Robot Policies. In *Advances in Neural Information Processing Systems*, volume 36, pp. 55681–55693. Curran Associates, Inc., 2023. doi: 10.52202/075280-2430. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ae54ce310476218f26dd48c1626d5187-Abstract-Conference.html.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-Distribution Detection with Deep Nearest Neighbors. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20827–20840. PMLR, 2022. URL https://proceedings.mlr.press/v162/sun22d.html.

Huajie Tan, Sixiang Chen, Yijie Xu, Zixiao Wang, Cheng Chi, Yuheng Ji, Yaoxu Lyu, Zhongxia Zhao, Xiansheng Chen, Peterson Co, Shaoxuan Xie, Guocai Yao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. General Process Reward Modeling for Robotic Reinforcement Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22412–22422, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Tan_General_Process_Reward_Modeling_for_Robotic_Reinforcement_Learning_CVPR_2026_paper.html.

Raymond Tsao, Andrew Wagenmaker, and Sergey Levine. Learning Process Rewards via Success Visitation Matching for Efficient RL. *arXiv preprint arXiv:2606.23640*, 2026. URL https://arxiv.org/abs/2606.23640.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025. URL https://arxiv.org/abs/2506.15799.

Homer Rich Walke, Jonathan Heewon Yang, Albert Yu, Aviral Kumar, Jędrzej Orbik, Avi Singh, and Sergey Levine. Don't Start From Scratch: Leveraging Prior Data to Automate Robotic Reinforcement Learning. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1652–1662. PMLR, 2023. URL https://proceedings.mlr.press/v205/walke23a.html.

Tianbao Xie, Siheng Zhao, Chen Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2Reward: Reward Shaping with Language Models for Reinforcement Learning. In *International Conference on Learning Representations*, volume 2024, pp. 35663–35699, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/9941833e8327910ef25daeb9005e4748-Abstract-Conference.html.

Yijie Xu, Haopeng Jin, Run Zhou, Shengbang Liu, Sixiang Chen, Hongyang Cheng, Sicheng Hu, Peterson Co, Jinwen Luo, Huajie Tan, and Shanghang Zhang. Robo-Dopamine 2.0: History-Conditioned and OOD-Aware Process Reward Modeling for Robotic Manipulation. *arXiv preprint arXiv:2608.15680*, 2026. URL https://arxiv.org/abs/2608.15680.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2020. URL https://proceedings.mlr.press/v100/yu20a.html.

Sheng Yue, Guanbo Wang, Wei Shao, Zhaofeng Zhang, Sen Lin, Ju Ren, and Junshan Zhang. CLARE: Conservative Model-Based Reward Learning for Offline Inverse Reinforcement Learning. *arXiv preprint arXiv:2302.04782*, 2023. URL https://arxiv.org/abs/2302.04782.

Jiahui Zhang, Yusen Luo, Abrar Anwar, Sumedh Anand Sontakke, Joseph J Lim, Jesse Thomason, Erdem Biyik, and Jesse Zhang. ReWiND: Language-Guided Rewards Teach Robot Policies without New Demonstrations. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 460–488. PMLR, 2025. URL https://proceedings.mlr.press/v305/zhang25a.html.

Jianshu Zhang, Keliang Wu, Haoran Lu, Anbang Liu, Ce Zhang, Weijie Yin, Chengxuan Qian, Xiyuan Yang, Zhenyu Pan, Guo Ye, and Han Liu. Progress Reward Modeling for Robotic Learning: A Comprehensive Survey. *arXiv preprint arXiv:2607.21655*, 2026. URL https://arxiv.org/abs/2607.21655.

Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient Online Reinforcement Learning Fine-Tuning Need Not Retain Offline Data. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/504491292cb71e7681eedfe0e602b72f-Abstract-Conference.html. Warm-start RL (WSRL).

# A Additional Experimental Results

## A.1 Meta-World Results

Table S1 reports the success rates for all eight Meta-World tasks under the three reward-model baselines: RM-rule, RM-pre, and RM-pro. To provide a more detailed view of the training process, Figures S1 and S2 show the learning curves for RM-pre and RM-pro, respectively. The corresponding learning curves for RM-rule are presented in Figure 4 of the main paper.

Table S1: **Meta-World results over different baseline reward models.** Results are averaged over the final 10k environment steps and four random seeds.

| Task | RM-rule | | RM-pre | | RM-pro | |
|---|---|---|---|---|---|---|
| | Baseline | CRC | Baseline | CRC | Baseline | CRC |
| Door lock | 0.544 | **0.788** | 0.874 | **0.910** | 0.887 | **0.939** |
| Handle press-side | 0.563 | **0.798** | 0.866 | **0.987** | 0.682 | **0.978** |
| Door unlock | 0.290 | **0.613** | 0.472 | **0.509** | 0.553 | **0.752** |
| Sweep into | 0.107 | **0.339** | 0.002 | **0.226** | 0.214 | **0.436** |
| Faucet close | 0.655 | **0.795** | 0.221 | **0.947** | 0.957 | **0.991** |
| Hammer | 0.797 | **0.848** | 0.522 | **0.837** | 0.994 | **0.997** |
| Lever pull | 0.142 | **0.312** | 0.355 | **0.471** | 0.380 | **0.577** |
| Button press-wall | 0.780 | **0.908** | 0.739 | **0.938** | 0.756 | **0.937** |
| **Average** | 0.485 | **0.675** | 0.506 | **0.728** | 0.678 | **0.826** |

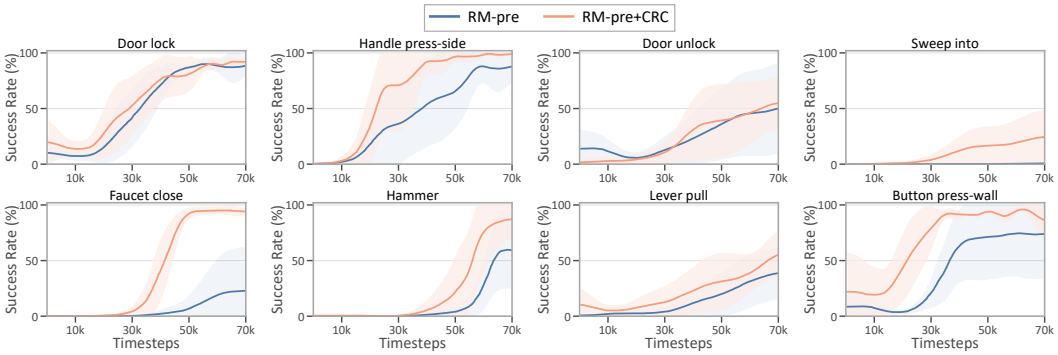



Figure S1: **Learning curves on eight Meta-World tasks with RM-pre.** Curves show mean success rates over four random seeds.

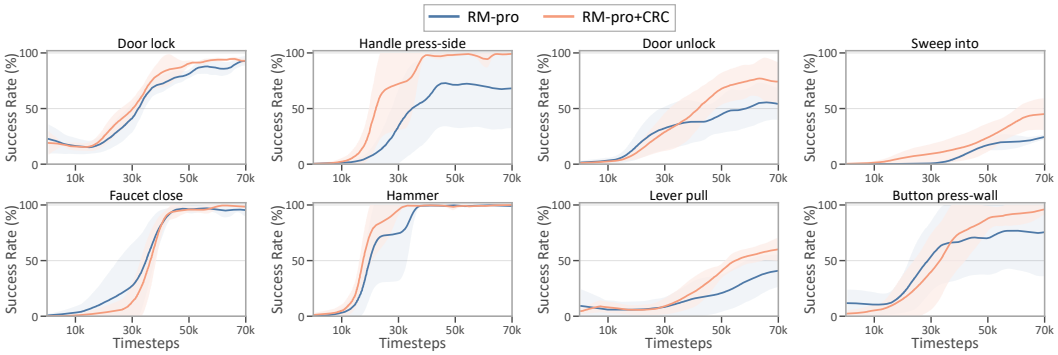



Figure S2: **Learning curves on eight Meta-World tasks with RM-pro.** Curves show mean success rates over four random seeds.

### A.2 Robomimic Results

Table S2 reports the success rates for both Robomimic tasks, Can and Lift, under the two reward-model baselines: RM-pre and RM-pro. To provide a more detailed view of the training process, Figure S3 shows the learning curves for RM-pre. The corresponding learning curves for RM-pro are presented in Figure 4 of the main paper.

Table S2: **Robomimic results over different baseline reward models.** Results are averaged over the final 10k environment steps and four random seeds.

| Method | Can | Lift | Avg. |
|---|---|---|---|
| RM-pre | 0.356 | 0.927 | 0.642 |
| RM-pre + CRC | **0.566** | **0.963** | **0.764** |
| RM-pro | 0.396 | 0.918 | 0.657 |
| RM-pro + CRC | **0.583** | **0.953** | **0.768** |

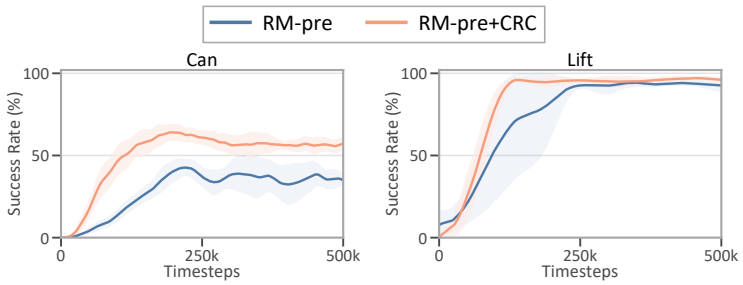



Figure S3: **Learning curves on two Robomimic tasks with RM-pre.** Curves show mean success rates over four random seeds.

### A.3 Real-world

Table S3 reports the real-world results from Figure 6. Figure S8 shows the setups for the three real-world manipulation tasks. Appendix B.3 provides further details of the experimental setup.

Table S3: **Real-world results over three manipulation tasks.**

| Method | Put Bread in Basket | Open Drawer & Store Lemon | Place Stationery in Holder | Avg. |
|---|---|---|---|---|
| RM-pro | 0.742 | 0.773 | 0.821 | 0.779 |
| RM-pro + CRC | **0.941** | **0.909** | **0.938** | **0.929** |

### A.4 Diagnostic Results

Figure S4 and Table S4 report the learning curves and final-window success rates, respectively, for the oracle interventions in Figure 1(d). Baseline uses the original RM-rule predictions. ID and OOD calibration replace predictions with simulator rewards only on ID and OOD states, respectively, whereas ID+OOD calibration replaces all predictions with oracle rewards.

Table S4: **Meta-World results under diagnostic reward interventions.** Success rates are averaged over the final 10k training steps and four random seeds.

| Intervention | Door lock | Handle press-side | Door unlock | Sweep into | Avg. |
|---|---|---|---|---|---|
| Baseline | 0.544 | 0.563 | 0.290 | 0.107 | 0.376 |
| ID calibration | 0.609 | 0.548 | 0.323 | 0.227 | 0.427 |
| OOD calibration | **0.830** | 0.750 | 0.599 | 0.237 | 0.604 |
| ID+OOD calibration | 0.818 | **0.924** | **0.640** | **0.362** | **0.686** |

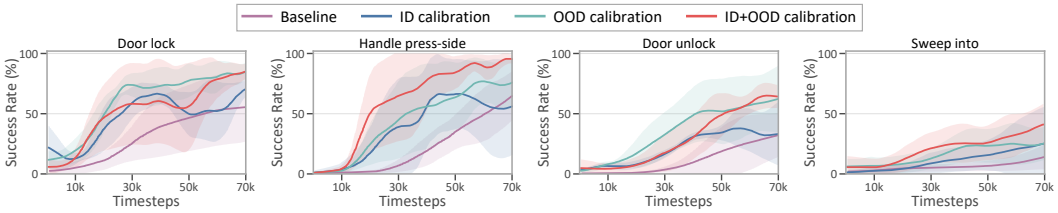



Figure S4: **Learning curves under diagnostic reward calibrations.** Results cover four Meta-World tasks with RM-rule as the baseline. Curves show mean success rates over four random seeds.

### A.5 Additional Ablation Results

Table S5 and Figures S5–S7 expand the ablations in Section 5.3 and Figure 7.

Table S5: **Meta-world Ablation results.** Success rates are averaged over the final 10k training steps and four random seeds.

| Variant | Door lock | Handle press-side | Door unlock | Sweep into | Avg. |
|---|---|---|---|---|---|
| **Components** | | | | | |
| Baseline | 0.544 | 0.563 | 0.290 | 0.107 | 0.376 |
| Reweighting | 0.678 | 0.744 | 0.597 | 0.319 | 0.585 |
| Reweighting+Compensation | **0.788** | **0.798** | **0.613** | **0.339** | **0.635** |
| **OOD threshold** | | | | | |
| $\tau_{80}$ | 0.604 | **0.831** | 0.332 | 0.281 | 0.512 |
| $\tau_{90}$ | **0.788** | 0.798 | **0.613** | **0.339** | **0.635** |
| $\tau_{95}$ | 0.604 | 0.705 | 0.607 | 0.182 | 0.524 |
| **Neighborhood size** | | | | | |
| $k = 5$ | 0.694 | 0.765 | 0.497 | 0.223 | 0.545 |
| $k = 10$ | **0.788** | 0.798 | **0.613** | **0.339** | **0.635** |
| $k = 20$ | 0.769 | **0.840** | 0.366 | 0.245 | 0.555 |

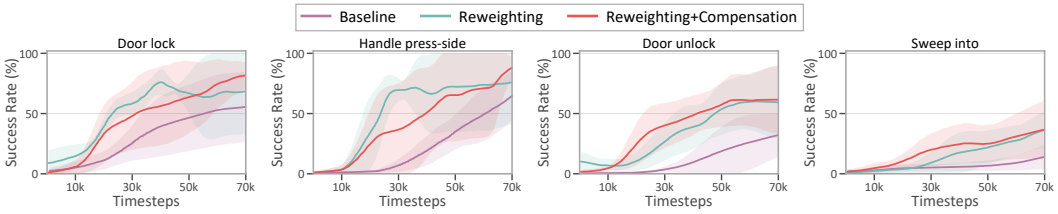



Figure S5: **Learning curves for component ablations.** Curves show mean success rates over four random seeds.

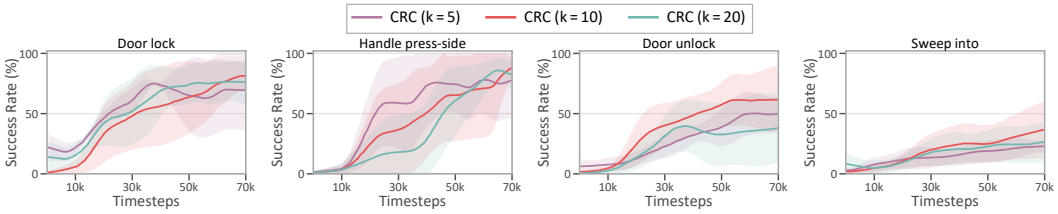



Figure S6: **Learning curves for different CRC neighborhood sizes.** Curves show mean success rates over four random seeds.

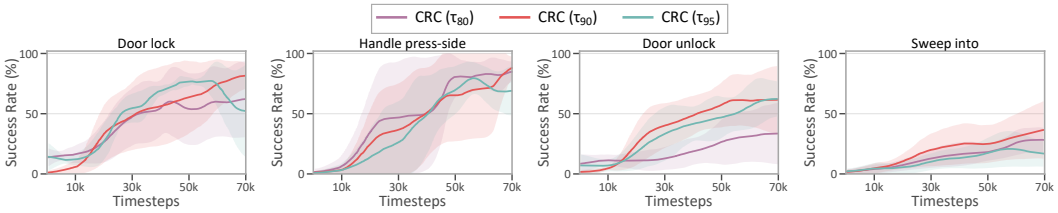



Figure S7: **Learning curves for different OOD threshold.** Curves show mean success rates over four random seeds.

# B Experimental Details

## B.1 Baseline Reward Models

We consider three learning-based reward models trained on offline trajectories with distinct supervision signals. Following Section 3.1, the joint input $x_t = (o_t, \ell)$ contains the observation $o_t$ and task instruction $\ell$, and the model predicts a scalar score $\mathcal{R}_\phi(x_t)$. For models trained with pointwise scalar supervision, we write $\mathcal{D}_{\mathrm{RM}} = \{(x_i, y_i)\}_{i=1}^{N_{\mathrm{RM}}}$, where $x_i$ is a joint observation and instruction input and $y_i$ is its target, such as a rule-based reward or task-progress annotation. RM-pre instead learns from same-task trajectory comparisons, as described below. The reward models remain frozen during online policy optimization.

**Rule-based Reward Model (RM-rule).** RM-rule learns a visual reward predictor from dense scalar labels generated by task-specific handcrafted rules, such as those based on gripper-to-object and object-to-goal distances (Yu et al., 2020). These rules encode geometric relations relevant to the task and provide graded supervision for intermediate observations along offline trajectories. Each joint input $x_i$ is paired with its corresponding rule-generated target $y_i$, so that the model learns to infer the prescribed reward from visual evidence and task context. Regression-based training minimizes the mean squared error between predicted rewards and scalar targets over labeled inputs:

$$\mathcal{L}_{\mathrm{MSE}}(\phi) = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{RM}}}\left[\left(\mathcal{R}_\phi(x) - y\right)^2\right]. \tag{10}$$

**Preference-based Reward Model (RM-pre).** RM-pre follows the reward-learning objective of SORS (Memarian et al., 2021) and uses a Bradley–Terry preference model to learn scalar rewards that preserve trajectory rankings induced by observed discounted returns. Preference-based reward learning infers reward functions from pairwise trajectory comparisons without requiring pointwise supervision. Let $\mathcal{D}_{\mathrm{pref}}$ contain same-task trajectory pairs $(\xi^+, \xi^-)$, where $\xi^+$ is preferred to $\xi^-$. For a trajectory $\xi$ with joint inputs $(x_0, \ldots, x_{L_\xi - 1})$, all containing the same task instruction, the predicted return is $G_\phi(\xi) = \sum_{t=0}^{L_\xi - 1} \gamma_{\mathrm{RM}}^t \mathcal{R}_\phi(x_t)$, where $L_\xi$ is the number of reward-model inputs and $\gamma_{\mathrm{RM}} \in [0, 1]$ is the reward-learning discount factor. Under the Bradley–Terry model, training minimizes the negative log-likelihood of the observed preferences:

$$\mathcal{L}_{\mathrm{pref}}(\phi) = -\mathbb{E}_{(\xi^+,\xi^-)\sim\mathcal{D}_{\mathrm{pref}}}\left[\log \frac{\exp\big(G_\phi(\xi^+)\big)}{\exp\big(G_\phi(\xi^+)\big) + \exp\big(G_\phi(\xi^-)\big)}\right], \tag{11}$$

which encourages higher predicted returns for preferred trajectories.

Since trajectory-level preference supervision does not provide per-state scalar reward labels, we construct pseudo-labels for the offline reference inputs using predictions from the frozen RM-pre. These pseudo-labels serve as the reference values $y_j$ in Equation (7) for computing the neighbor-based reward estimate and label disagreement. They are computed once when building the reference bank and remain fixed during online RL, without requiring additional reward annotations or reward-model retraining.

**Progress-based Reward Model (RM-pro).** RM-pro is trained to predict normalized task progress from successful demonstrations (Zhang et al., 2025), using temporal position as supervision. For a successful demonstration $\xi$ represented by $L_\xi \geq 2$ temporally ordered joint inputs $(x_0, \ldots, x_{L_\xi - 1})$, we assign the progress label

$$y_t^{\mathrm{pro}} = \frac{t}{L_\xi - 1}, \qquad t = 0, \ldots, L_\xi - 1. \tag{12}$$

The initial input has target $y_0^{\text{pro}} = 0$, and the final input at task completion has target $y_{L_\xi-1}^{\text{pro}} = 1$, with intermediate targets increasing linearly with temporal position. Normalizing by the demonstration length places these labels on a common $[0, 1]$ scale across demonstrations of different lengths. We then train $\mathcal{R}_\phi(x_t)$ to predict the progress target $y_t^{\text{pro}}$ from each joint input $x_t$.

### B.2 Simulation Experimental Details

**Meta-World.** We evaluate all three reward models on eight Meta-World tasks (Yu et al., 2020): `door-lock`, `handle-press-side`, `door-unlock`, `sweep-into`, `faucet-close`, `hammer`, `lever-pull`, and `button-press-wall`. We follow the ReWiND (Zhang et al., 2025) visual preprocessing and offline-to-online RL pipeline. Policies receive frozen DINOv2 image features and a task-instruction embedding. The RM-rule training pipeline uses five demonstration trajectories per task to learn from rule-generated reward labels. Offline demonstrations are relabeled by the reward model and used for IQL pretraining, followed by online optimization with RLPD (Ball et al., 2023) and mixed sampling from offline and online replay. Online learning starts after 4k environment steps; a further 4k steps of critic-only training precede actor updates.

**Robomimic.** We evaluate RM-pre and RM-pro on the `Can` and `Lift` tasks in Robomimic (Mandlekar et al., 2022), using a Panda arm. For each task, the first 10 demonstrations provide reward-model training data, and the next 30 train a behavior-cloning (BC) policy and populate demonstration replay. Policies use RGB observations together with three consecutive proprioceptive observations containing end-effector position, orientation, and gripper state. Based on the implementation of Hu et al. (2024), we collect 40 additional episodes with the BC policy to initialize replay, then train the online actor with the RLPD configuration for 500k environment steps.

Table S6 summarizes the simulation environment settings and Table S7 reports the hyperparameters.

Table S6: **Simulation environment settings.**

| Setting | Meta-World | Robomimic |
|---|---|---|
| Tasks | Eight manipulation tasks | `Can`, `Lift` |
| Reward models | RM-rule, RM-pre, RM-pro | RM-pre, RM-pro |
| Policy image input | $224 \times 224$ | $96 \times 96$ |
| Maximum episode length | 128 environment steps | 200 environment steps |
| Termination on success | Enabled | Enabled |

Table S7: **hyperparameters for the simulation training implementations.**

| Hyperparameter | Meta-World | Robomimic |
|---|---|---|
| Algorithm | RLPD (Ball et al., 2023) | RLPD (Ball et al., 2023) |
| Actor and critic learning rate | $10^{-4}$ | $10^{-4}$ |
| Batch size | 256 | 256 |
| Discount factor $\gamma$ | 0.99 | 0.99 |
| Number of critics | 5 | 2 |
| Target critics used online | 2 sampled critics | Both critics |
| Offline IQL updates | 10k gradient updates | - |
| Initial online collection | 4k environment steps | 40 additional BC episodes |
| Actor update start | After 8k environment steps | After BC collection |
| Online training budget | 70k environment steps | 500k environment steps |

### B.3 Real-World Experimental Details

We evaluate three real-world manipulation tasks using a PIPER arm with third-person and wrist RGB cameras. Figure S8 shows the setup for each task. Demonstrations are collected using a PIKA Sense handheld device tracked by HTC VIVE Base Stations. For each task, 30 demonstrations train a behavior-cloning policy, and the first 10 also train RM-pro. Policies combine frozen DINOv2 features with proprioception and predict chunks of 10 actions on a 10 Hz time grid. We initialize the

actor from BC and fine-tune it with SAC (Haarnoja et al., 2018). Baseline and CRC share the learner settings and initial data prefix. Table S8 summarizes the shared environment settings. Table S9 lists the complete language instructions and hyperparameters for each task.




Figure S8: **Real-world task setups.**

Table S8: **Shared real-world environment settings.**

| Setting | Value |
|---|---|
| Robot | PIPER 6-DoF arm and 1-DoF gripper |
| Cameras | Two Orbbec RGB cameras: third-person and wrist views |
| Camera stream | $640 \times 360$ pixels at 30 Hz |
| Image preprocessing | Resize to $398 \times 224$ pixels |
| Action chunk | 10 actions, each 7-dimensional, on a 10 Hz time grid |

Table S9: **Real-world task instructions and hyperparameters.**

| Hyperparameter | Put Bread in Basket | Open Drawer & Store Lemon | Place Stationery in Holder |
|---|---|---|---|
| Language instruction | `put the bread into the basket` | `open the drawer and put the lemon into it` | `place the marker and eraser into the pen holder` |
| Algorithm | SAC | SAC | SAC |
| Number of critics | 2 | 2 | 2 |
| Batch size | 64 | 64 | 64 |
| Discount factor $\gamma$ | 0.99 | 0.99 | 0.99 |
| Maximum episode length | 300 | 500 | 800 |
| Actor update start | 4k steps | 4k steps | 6k steps |
| Online training budget | 15k steps | 30k steps | 30k steps |
| Number of neighbors $k$ | 10 | 10 | 10 |
| Threshold quantile $q$ | 0.90 | 0.90 | 0.90 |
| Confidence decay coefficient $\alpha$ | 1.2 | 1.2 | 1.2 |
| Compensation coefficient $\beta$ | 0.3 | 0.3 | 0.3 |
| Uncertainty penalty coefficient $\lambda_r$ | 1.0 | 1.0 | 1.0 |