CPR: REVIVING HARD-TASK LEARNING IN MULTI-TASK REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-task reinforcement learning (MTRL) aims to learn multiple tasks with a shared model, yet hard tasks often make limited progress when trained alongside easier ones. We identify weak temporal-difference (TD) learning signals as a key characteristic of this failure and introduce expectile temporal-difference error (XTD), an upper expectile of signed TD errors that reveals informative positive value corrections obscured by averaging. Building on XTD, we propose Cross-level Progress Recovery (CPR), which revives hard-task learning through complementary inter-task and intra-task mechanisms. Inter-task dropout uses relative return and declining task-level XTD to identify sufficiently learned tasks and mitigate shared-critic over-specialization, while intra-task prioritization emphasizes high-XTD states to strengthen informative learning signals within hard tasks. Experiments on Meta-World show that CPR substantially improves overall performance, particularly on hard tasks, without relying on parameter resets. Trajectory and component analyses show that XTD highlights critical states and support the complementary roles of both mechanisms in reducing critic interference and improving hard-task learning.

1 INTRODUCTION

Multi-task reinforcement learning (MTRL) aims to learn multiple behaviors within a single agent by sharing representations and parameters across tasks (Thrun & Schwartz, 1994). A straightforward extension of single-task RL can condition a shared policy and critic on task identifiers, but joint training often suffers from imbalance across tasks with different difficulties and learning speeds. Easy tasks can produce strong reward signals and dominate shared optimization, while hard tasks struggle to develop sufficient learning signals and may remain completely unsolved. Existing approaches mitigate this problem by learning task-conditioned representations (Sodhani et al., 2021; Yang et al., 2020) or balancing task-level quantities such as returns, gradients, and sampling distributions (Hessel et al., 2019; Yu et al., 2020a; Liu et al., 2021a; Cho et al., 2024). While effective to some extent, these approaches still leave challenging tasks poorly learned in MTRL settings.

Recent methods have achieved substantial improvements by periodically resetting the actor or critic networks (Cho et al., 2025; 2024; He et al., 2026). Resetting can alleviate the dominance of already learned tasks and provide stalled tasks with an opportunity to restart learning, but the performance often depends heavily on this mechanism. Without resets, hard-task performance can deteriorate substantially. Moreover, abruptly discarding learned parameters can temporarily degrade previously acquired behaviors and introduce instability due to the resulting distribution shift, making repeated resets undesirable in practical deployment. Resetting is also a coarse intervention that does not distinguish tasks that have sufficiently converged from those that still require learning. This raises a fundamental question: *How can we enable hard tasks to learn without forcing network resets?*

Specifically, we introduce *expectile temporal-difference error* (XTD), which captures the upper tail of TD errors and thereby highlights transitions that remain underestimated by the critic. By emphasizing upper-tail TD errors rather than their average, XTD can identify learning signals from a small fraction of underestimated samples that would otherwise be obscured, making it particularly suitable for hard tasks with sparse informative experience. Unlike return alone, XTD provides a direct signal of how much useful value correction remains within a task. Figure 1 illustrates this distinction on Meta-World MT10 (Yu et al., 2020b). Under single-task soft actor-critic (SAC) (Haarnoja

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.

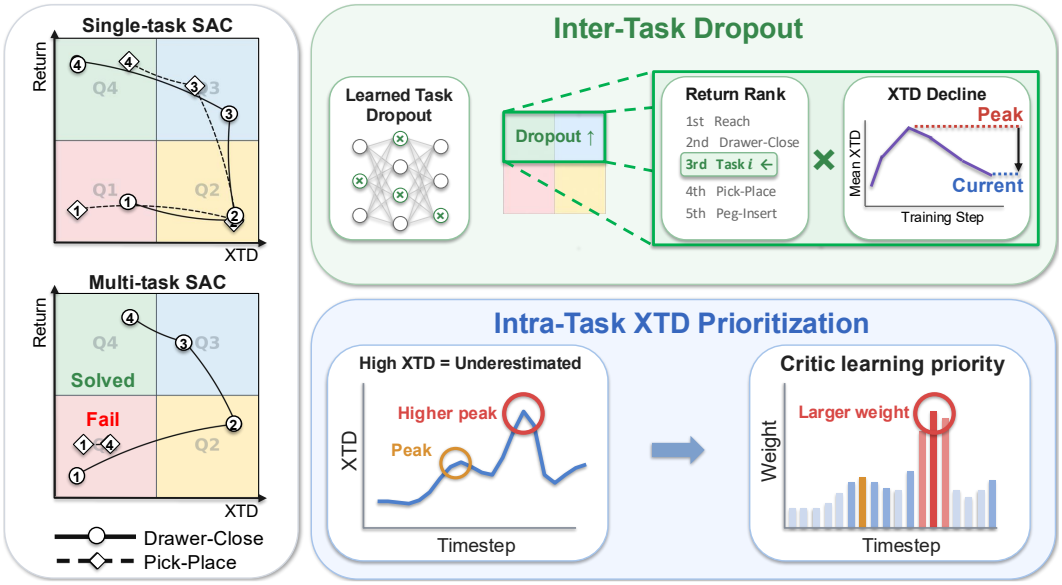

Figure 1: **Overview of CPR.** *Left:* Return–XTD trajectories for Drawer-Close (easy) and Pick-Place (hard). Under single-task SAC, the tasks progress toward convergence, whereas under multi-task SAC, the hard task stalls at low return. *Top right:* Inter-task dropout regularizes sufficiently learned tasks. *Bottom right:* Intra-task XTD prioritization emphasizes high-XTD samples.

et al., 2018), both an easy Drawer-Close task and a harder Pick-Place task follow a common learning progression: starting from low return and low XTD (Q1), XTD first emerges as useful transitions are discovered (Q2), return subsequently increases (Q3), and XTD decreases as learning converges (Q4). Under vanilla MT-SAC, however, the easy task follows this progression from Q1 to Q4, while the hard task remains trapped in Q1. This suggests that shared multi-task training can suppress the learning signal required for hard tasks to make progress, even when tasks are learnable in isolation.

Based on this observation, we propose **Cross-level Progress Recovery (CPR)**, an MTRL framework that revives hard-task learning through complementary inter-task and intra-task mechanisms. At the inter-task level, CPR uses XTD and return to identify sufficiently learned tasks and applies rank-based dropout to prevent shared critic specialization toward them, reducing interference with hard-task value learning. At the intra-task level, CPR prioritizes high-XTD transitions to amplify informative but weak learning signals within hard tasks. Together, these mechanisms enable stalled tasks to recover successful learning progression without relying on parameter resets. Experiments on Meta-World demonstrate that CPR substantially improves multi-task performance, particularly on hard tasks, and remains effective with and without resets.

2 Related Work

2.1 Multi-Task Reinforcement Learning

Multi-task RL jointly learns multiple tasks by sharing knowledge while mitigating negative transfer (Teh et al., 2017; Sodhani et al., 2021), whereas meta-learning (Finn et al., 2017; Kim et al., 2025; Lee et al., 2026) leverages cross-task knowledge to enable rapid adaptation to new tasks. Architectural methods use policy distillation or modularization (Andreas et al., 2017; Devin et al., 2017; Yang et al., 2020; Lan et al., 2023), parameter and representation composition (Sun et al., 2022), and mixture-of-experts or dynamic routing (Cheng et al., 2023; Hendawy et al., 2024; He et al., 2024b; Kong et al., 2025). Optimization methods reduce task interference via gradient manipulation and balancing (Yu et al., 2020a; Liu et al., 2021a; Chen et al., 2018; Liu et al., 2023) and multi-objective optimization (Xiao et al., 2023; Navon et al., 2022; Liu et al., 2021b). Training-dynamics methods address task difficulty via reward scaling (Hessel et al., 2019; Cho et al., 2025), scheduling and curricula (Huang et al., 2023; Cho et al., 2024), and phased policy evolution (He et al., 2026). Others use policy guidance (He et al., 2024a) or reward transfer (Ma et al., 2025).

2.2 Regularization and Learning Progress in RL

Regularization in RL stabilizes learning and preserves model capacity under bootstrapping and non-stationarity (Kumar et al., 2021; Lyle et al., 2022; Kumar et al., 2022; He et al., 2024c). Existing approaches regularize parameters and representations (Gogianu et al., 2021; Bjorck et al., 2021; Elsayed et al., 2024), maintain plasticity through parameter or feature adaptation (Nikishin et al., 2022; Sokar et al., 2023; Nikishin et al., 2023), and regularize value estimates to mitigate out-of-distribution overestimation in offline RL (Kumar et al., 2020b; Yeom et al., 2024; Chae et al., 2026). Entropy regularization encourages policy diversity and exploration (Haarnoja et al., 2018; Han & Sung, 2021a;b). Curriculum and goal-selection methods adapt training based on learning progress (Colas et al., 2019; Portelas et al., 2020), while learnability-based sampling prioritizes tasks with useful learning opportunities (Rutherford et al., 2024). We instead use task-wise learning progress to selectively regularize the shared critic, reducing over-specialization toward sufficiently learned tasks while preserving hard-task learning.

2.3 Temporal-Difference Errors as Learning Signals

Temporal-difference (TD) errors are widely used for value learning and experience prioritization (Schaul et al., 2016; De Bruin et al., 2018; Daley & Amato, 2019; Liu et al., 2021c), with reliability improved through correction and adaptive reweighting (Kumar et al., 2020a; Pan et al., 2022; Sujit et al., 2023; Pleiss et al., 2026). TD-derived signals have also guided exploration and curricula (Gehring & Precup, 2013; Jiang et al., 2021; Zhao et al., 2025), as well as adaptive value learning and update control (Li et al., 2024; Omura et al., 2024; Sharifnassab et al., 2026). Unlike individual or average TD errors, we use the upper expectile of signed TD errors to capture informative positive corrections that can be obscured when most TD errors are small. This is particularly important for hard tasks with weak learning signals, where a small fraction of underestimated transitions can drive further learning.

3 Preliminaries

3.1 Multi-Task Reinforcement Learning and Temporal-Difference Error

MTRL considers N tasks $\mathcal{T} = \{0, \dots, N-1\}$, where each task $i \in \mathcal{T}$ is modeled as an MDP $\mathcal{M}_i = (\mathcal{S}, \mathcal{A}, P_i, r_i, \gamma)$. Here, $\mathcal{S}$ and $\mathcal{A}$ are shared state and action spaces, $P_i(s' \mid s, a)$ and $r_i(s, a)$ are the transition dynamics and reward function, and $\gamma \in [0, 1)$ is the discount factor. At the beginning of each episode, a task i is given to the agent along with its identifier z_i. At each time step, the agent observes s_t, selects $a_t \sim \pi(\cdot \mid s_t, z_i)$, and receives $r_i(s_t, a_t)$ and $s_{t+1} \sim P_i(\cdot \mid s_t, a_t)$. The shared policy $\pi(a \mid s, z_i)$ is trained across tasks to maximize the expected discounted return $\frac{1}{N}\sum_{i=0}^{N-1} \mathbb{E}_{\pi, P_i}[\sum_{t=0}^{T-1} \gamma^t r_i(s_t, a_t)]$, where T denotes the episode horizon.

For Q-learning-based methods, the critic $Q(s, a, z_i)$ represents the action value and is learned toward the Bellman optimality target. Transitions collected by the policy π from task i are stored in a replay buffer $\mathcal{B}^i$, from which a mini-batch $B^i \subset \mathcal{B}^i$ is sampled for critic updates. The critic is trained by

$$\min_{Q} \frac{1}{N} \sum_{i=0}^{N-1} \frac{1}{|B^i|} \sum_{(s,a,r,s') \in B^i} \delta_i(s, a, r, s')^2, \tag{1}$$

where $\delta_i(s, a, r, s') = r + \gamma \max_{a'} \bar{Q}(s', a', z_i) - Q(s, a, z_i)$ is the signed TD error and $\bar{Q}$ denotes the target critic. Positive and negative TD errors indicate underestimation and overestimation, respectively, relative to the bootstrapped target toward which the current value estimate is updated. The specific form of the TD error may vary depending on the underlying RL algorithm.

3.2 Expectile Regression

Expectile regression (Newey & Powell, 1987) generalizes mean regression by asymmetrically weighting positive and negative residuals. Consider a random target variable Y conditioned on a context c. Expectile regression seeks a scalar v that summarizes the conditional distribution of Y given c. The q-expectile is defined as

$$e_q(Y \mid c) = \arg\min_{v} \mathbb{E}\left[\, |q - \mathbb{I}(Y - v < 0)| \, (Y - v)^2 \,\middle|\, c \right], \qquad q \in (0, 1). \tag{2}$$

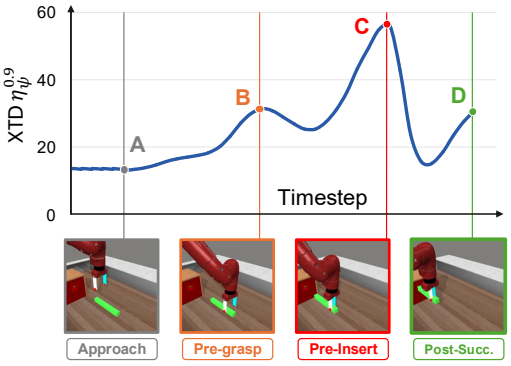

Figure 2: **XTD along a PEG-INSERT-SIDE trajectory.** XTD peaks near behaviorally critical interactions such as grasping and insertion.

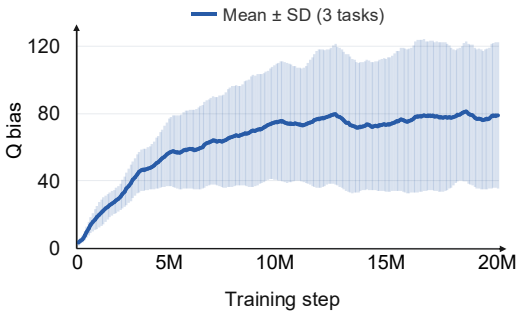

Figure 3: **Q bias under fixed random policies.** Mean Q bias across the three fixed-policy tasks increases overall as the other tasks are learned.

where $\mathbb{I}(\cdot)$ is the indicator function and q controls the asymmetry between positive and negative residuals: samples with $Y > v$ receive weight q, while those with $Y < v$ receive weight $1 - q$. When $q = 0.5$, the expectile reduces to the conditional mean, whereas increasing q toward 1 shifts it toward the upper endpoint of the conditional distribution, approaching its maximum under suitable conditions. Expectile regression has also been adopted in RL for value estimation, e.g., IQL (Kostrikov et al., 2022). In contrast, we leverage it to estimate upper TD errors conditioned on the state and task, allowing XTD to capture large TD errors that may be obscured by their average.

4 METHOD

4.1 EXPECTILE TEMPORAL-DIFFERENCE ERROR

As motivated in Section 1, we first define XTD as the learning signal for our inter-task and intra-task control. Given a transition (s, a, r, s') from task i, let $\delta_i = \delta_i(s, a, r, s')$ denote its signed TD error defined in Eq. (1). We define the XTD function $\eta_\psi^q(s, z_i)$ to estimate the upper expectile of δ_i conditioned on state s and task i. Specifically, in the expectile regression of Eq. (2), we set $Y = \delta_i$, $c = (s, z_i)$, and $v = \eta_\psi^q(s, z_i)$, which yields

$$\mathcal{L}_{\text{XTD}}(\psi) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{1}{|B^i|} \sum_{(s,a,r,s') \in B^i} \left| q - \mathbb{I}\Big(\delta_i - \eta_\psi^q(s, z_i) < 0\Big) \right| \Big(\delta_i - \eta_\psi^q(s, z_i)\Big)^2, \quad (3)$$

where $q \in (0.5, 1)$ and ψ denotes the parameters of the XTD function. At the optimum ψ^*, it estimates the conditional q-expectile of the signed TD error. As q increases from 0.5 toward 1, XTD shifts from the conditional mean $\mathbb{E}_{a\sim\pi,\, s'\sim P_i}[\delta_i \mid s, z_i]$ toward the maximum $\max_{a,s'} \delta_i(s, a, r_i(s, a), s')$ over transitions supported by π and P_i. Thus, unlike the average TD error, XTD captures large positive TD corrections even when they occur only in a small fraction of transitions. Since positive TD errors indicate underestimation relative to the bootstrapped target, a high XTD indicates substantial remaining learning signal at the corresponding state. This property is particularly relevant to hard tasks, where informative transitions can be sparse and otherwise obscured by the average TD error. We empirically validate this interpretation of XTD in Section 5.

Figure 2 further illustrates this property by plotting XTD over a successful trajectory of PEG-INSERT-SIDE, one of the challenging tasks in Meta-World. XTD is initially low and reaches its first peak just before the agent grasps the peg. It peaks again just before insertion into the hole and subsequently decreases. These peaks correspond to behaviorally critical stages where action quality can lead to substantially different value targets. Until the critic sufficiently captures these high-value behaviors, large positive TD corrections remain and are reflected by high XTD. XTD can therefore identify behaviorally critical states with meaningful remaining learning signals. We leverage this learning signal through two complementary mechanisms: inter-task dropout regulates sufficiently learned tasks, while intra-task prioritization emphasizes informative transitions within each task.

4.2 INTER-TASK DROPOUT: MITIGATING TASK OVER-SPECIALIZATION

To investigate why some tasks fail to progress beyond Q1 under multi-task training, as observed in Figure 1, we conduct a controlled MT-SAC experiment on three hard tasks that typically make limited learning progress: PUSH, PICK-PLACE, and PEG-INSERT-SIDE. We fix their policies to random policies throughout training, allowing us to isolate how their value estimates are affected as the other tasks are learned. We measure the initial-state value bias as $\mathbb{E}_{s_0,a\sim\pi(\cdot|s_0)}[|R - Q(s_0, a)|]$, where R is the rollout return from (s_0, a) under the fixed policy π and $Q(s_0, a)$ is the corresponding critic estimate. Figure 3 reports the mean and standard deviation across the three fixed-policy tasks. Overall, Q bias increases substantially as other tasks are learned. Although the critic continues to learn from these fixed-policy tasks, their random policies provide weak reward signals, making their value estimates more susceptible to interference from other tasks through the shared critic. This interference is particularly problematic for tasks with limited learning progress, whose weak learning signals may not counteract specialization toward sufficiently learned tasks. We term this phenomenon *task over-specialization*. We mitigate it with inter-task dropout that regularizes sufficiently learned tasks while minimally affecting actively learning tasks.

To identify sufficiently learned tasks and control the dropout strength, we jointly track return and XTD. For each task i, we maintain its average return and XTD, denoted by $\bar{R}^i$ and $\bar{\eta}^i$, respectively, using exponential moving averages (EMA) throughout training. As illustrated by the return–XTD trajectory in Figure 1, a task progresses from Q3 toward Q4 as it maintains high return while its XTD declines from its peak, indicating diminishing positive TD corrections. We capture this progression by tracking the peak positive XTD as $\bar{\eta}^i_{\max} \leftarrow \max(\bar{\eta}^i_{\max}, \bar{\eta}^i, 0)$. We then combine the relative return with the decline from this peak to determine the dropout probability.

Specifically, we define the task-specific dropout probability as

$$p^i_{\text{drop}} = p_{\max}\left(1 - \frac{\text{Rank}(\bar{R}^i)}{N-1}\right) \text{clip}\left(\frac{\bar{\eta}^i_{\max} - \bar{\eta}^i}{\bar{\eta}^i_{\max} + \epsilon}, 0, 1\right), \tag{4}$$

where $\text{Rank}(\bar{R}^i) \in \{0, \ldots, N-1\}$ ranks tasks in descending order of return, such that higher-return tasks receive smaller ranks. Here, $p_{\max}$ is the maximum dropout probability and $\epsilon > 0$ is a small constant. The first term favors tasks with relatively high return, while the second increases as XTD declines from its peak. Their product assigns strong dropout to tasks that both perform well and exhibit reduced XTD, corresponding to progression toward Q4. In contrast, tasks with low return or substantial remaining XTD receive little dropout, minimizing disruption to their ongoing learning.

We apply p^i_{drop} to the shared critic when updating task i. This selectively regularizes the critic for sufficiently learned tasks, limiting further specialization toward their value functions while largely preserving the learning of other tasks. We empirically examine whether this selective regularization mitigates task over-specialization and facilitates hard-task learning in Section 5.3.

4.3 INTRA-TASK PRIORITIZATION: XTD-BASED CRITIC LEARNING

While inter-task dropout mitigates interference across tasks, hard-task learning also requires strengthening informative learning signals within each task. As illustrated in Figure 2, XTD highlights behaviorally critical states where substantial TD corrections remain. This provides a direct signal for identifying where critic learning should be emphasized within a hard task. We therefore use XTD to prioritize critic updates at these states.

For each transition $(s, a, r, s') \in B^i$ of task i, we define the XTD-based weight as

$$w^i(s) = \frac{1 + \alpha\eta^q_\psi(s, z_i)}{\frac{1}{|B^i|}\sum_{(\tilde{s},\tilde{a},\tilde{r},\tilde{s}')\in B^i}\left(1 + \alpha\eta^q_\psi(\tilde{s}, z_i)\right)}, \tag{5}$$

where $\alpha \geq 0$ controls the strength of XTD-based prioritization. When $\alpha = 0$, all transitions receive equal weight, while increasing α places greater emphasis on transitions with higher XTD. The normalization keeps the weights at unit mean and preserves the overall scale of the critic loss. We use linear weighting to avoid excessive concentration on a few high-XTD transitions. For stable weighting, we additionally standardize and clip XTD-based priorities, as detailed in Appendix A.

Using these weights, we define the XTD-prioritized critic objective as

$$\mathcal{L}_Q = \frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{|B^i|}\sum_{(s,a,r,s')\in B^i} w^i(s)\left(\delta_i(s,a,r,s')\right)^2. \tag{6}$$

This objective allows informative TD corrections within hard tasks to contribute more strongly to value learning. The expectile level q controls the degree of prioritization. At $q = 0.5$, XTD reflects the conditional mean TD error, while increasing q shifts XTD toward larger positive TD corrections. As q approaches 1, critic learning increasingly focuses on states with the largest remaining corrections, resembling soft minimax behavior without directly optimizing the maximum TD error.

Finally, we propose CPR, a multi-task RL framework that combines inter-task dropout with intra-task XTD prioritization. While our approach can be incorporated into general multi-task RL algorithms, we instantiate CPR on MT-SAC, following the multi-task SAC framework commonly adopted in prior work. We additionally employ gradient normalization and episode-level return prioritization, which are commonly used to balance multi-task learning, and analyze their effects separately. By default, CPR does not use parameter resets. To assess reset effects, we additionally consider CPR (w/ Reset), which periodically resets network parameters during training and allows comparison with reset-based methods under the same setting. The overall training procedure is summarized in Algorithm 1, and further implementation details, including full actor and critic objectives, are provided in Appendix A.

Algorithm 1 CPR

Require: Tasks $\{\mathcal{T}_i\}_{i=0}^{N-1}$, expectile q, maximum dropout $p_{\max}$, priority strength α

Initialize policy π, critic Q, replay buffers $\mathcal{B}^i$, average return/XTD $\bar{R}^i, \bar{\eta}^i$, and peaks $\bar{\eta}^i_{\max} \leftarrow 0$
while training **do**
  Collect transitions and update average return $\bar{R}^i$
  Sample equal-sized minibatches $B^i \subset \mathcal{B}^i$
  Compute TD errors and update XTD estimator ψ via Eq. (3)
  Update average XTD $\bar{\eta}^i$ and peaks $\bar{\eta}^i_{\max}$
  Inter-Task Dropout: Compute p^i_{drop} via Eq. (4) and apply dropout to the shared critic
  Intra-Task Prioritization: Compute $w^i(s)$ via Eq. (5) and update critic via Eq. (6)
  Update policy via MT-SAC
end while

5 Experiments

5.1 Experimental Setup

Environment. To evaluate multi-task learning across tasks with varying learning difficulty, we consider Meta-World (Yu et al., 2020b), a widely used benchmark for multi-task robotic manipulation. A single task-conditioned agent jointly learns 10 tasks in MT10, or 50 tasks in MT50 using shared actor and critic networks. The tasks involve diverse manipulation skills, ranging from simple interactions such as opening a window or closing a drawer to multi-stage behaviors that require grasping, transporting, and inserting objects. Figure 4 visualizes representative tasks from MT10.

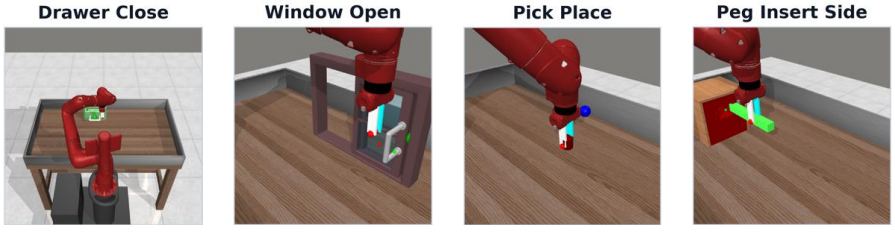

Figure 4: Visualization of representative tasks from Meta-World MT10.

Baselines and Implementation. We compare CPR with nine multi-task RL baselines, including reset-free and recent reset-based methods. Among reset-free methods, **MT-SAC** directly extends SAC (Haarnoja et al., 2018) using shared task-conditioned actor and critic networks, while

Table 1: **Performance on Meta-World MT10 and MT50 without parameter resets.**

	MT10	MT50				
	Success Rate (%)	Worst-k Tail-Task Success Rate (%)				
Algorithm	Success	$k = 10$	$k = 20$	$k = 30$	$k = 40$	$k = 50$ (All)
MT-SAC	73.7 ± 2.2	0.0 ± 0.0	0.0 ± 0.0	16.9 ± 2.8	36.8 ± 2.1	49.4 ± 1.7
MT-MH-SAC	78.3 ± 8.0	0.0 ± 0.0	0.0 ± 0.0	21.8 ± 2.6	40.8 ± 1.9	52.6 ± 1.5
PCGrad	78.6 ± 6.9	0.0 ± 0.0	15.5 ± 8.6	41.6 ± 6.5	56.0 ± 4.9	64.8 ± 4.0
Soft Modules	77.9 ± 3.0	0.0 ± 0.0	21.2 ± 8.3	44.5 ± 6.3	57.9 ± 4.8	66.2 ± 3.8
MOORE	70.0 ± 3.2	0.0 ± 0.0	0.0 ± 0.0	19.9 ± 3.0	38.6 ± 2.3	50.8 ± 1.9
PaCo	57.6 ± 2.1	0.0 ± 0.0	0.0 ± 0.0	15.8 ± 3.6	35.5 ± 3.0	48.3 ± 2.4
ARS (w/o Reset)	48.5 ± 2.5	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	13.0 ± 4.0	30.2 ± 3.5
HyMTRL (w/o Reset)	76.8 ± 3.8	0.0 ± 0.0	4.8 ± 3.6	31.9 ± 5.3	48.9 ± 3.9	59.1 ± 3.1
CPR	$\mathbf{97.0 \pm 0.1}$	$\mathbf{0.5 \pm 0.6}$	$\mathbf{34.5 \pm 8.4}$	$\mathbf{54.7 \pm 5.5}$	$\mathbf{65.8 \pm 4.1}$	$\mathbf{72.7 \pm 3.3}$

MT-MH-SAC uses task-specific heads. **PCGrad** (Yu et al., 2020a) mitigates conflicting task gradients, **Soft Modules** (Yang et al., 2020) learns task-conditioned routing over shared modules, **MOORE** (Hendawy et al., 2024) constructs orthogonal expert representations, and **PaCo** (Sun et al., 2022) represents task-specific parameters through shared parameters. Among reset-based methods, **SMT** (Cho et al., 2024) dynamically prioritizes difficult tasks and periodically resets network parameters, **ARS** (Cho et al., 2025) uses task histories for adaptive reward scaling together with periodic resets, and **HyMTRL** (He et al., 2026) combines phased policy evolution with parameter resets.

For a fair comparison, all methods use 20M transitions on MT10 and 100M on MT50, with aligned network sizes and training configurations. We evaluate both reset-free and reset settings. The former compares CPR with reset-free baselines and "w/o Reset" variants of reset-based methods, while the latter compares their original variants with CPR (w/ Reset). We reproduce all baselines except SMT using the implementations specified in Appendix B.2. For SMT, we report the original paper's results only in the reset setting, as its implementation is unavailable. For CPR, we use the configuration selected by hyperparameter search. Further experimental details and hyperparameter settings are in Appendices B and C, respectively.

5.2 Performance Comparison

We report success rates (%) at a predetermined final checkpoint as mean $\pm$ standard deviation across five seeds. Each evaluation uses 50 episodes. For MT10 and MT50, we report the average success rate across all tasks. To further evaluate performance on tasks that remain difficult to learn, we report Worst-k on MT50, defined as the average of each method's k lowest task-level success rates for $k \in \{10, 20, 30, 40\}$. Smaller k focuses on the lowest-performing tasks reflecting performance on tasks with limited learning progress. Per-task results are provided in Appendix D.

Table 1 compares the methods without parameter resets. CPR substantially outperforms all baselines on both MT10 and MT50, achieving 97.0 ± 0.1 on MT10 and 72.7 ± 3.3 on MT50. The improvement is particularly pronounced on hard tasks in MT50. While most baselines make little progress on their lowest-performing tasks, CPR begins to learn these tasks and achieves 34.5 ± 8.4 at $k = 20$, compared with 21.2 ± 8.3 for the strongest baseline. Nevertheless, the $k = 10$ performance remains low, indicating that the most challenging tasks still leave room for improvement.

Table 2 further compares methods designed to incorporate parameter resets. Even in this setting, CPR achieves strong performance on both benchmarks, reaching 99.1 ± 0.7 on MT10 and 74.6 ± 1.2 on MT50, although HyMTRL performs slightly better on MT50. More importantly, CPR exhibits a 2.1 and 1.9 percentage-point gap between training with and without resets on MT10 and MT50, respectively, substantially smaller than those of the reset-based baselines. This small gap supports our central finding that CPR can sustain strong multi-task performance without relying on parameter resets.

Table 2: **Performance on Meta-World MT10 and MT50 with parameter resets.** Δ_{reset} denotes the absolute performance gap from training without resets, where lower is better.

	MT10		MT50	
Algorithm	Success	$\Delta_{\text{reset}} \downarrow$	Success	$\Delta_{\text{reset}} \downarrow$
SMT	86.7 ± 8.6	–	56.0 ± 8.0	–
ARS	92.1 ± 2.9	43.6	65.1 ± 5.9	34.9
HyMTRL	82.2 ± 4.3	5.4	$\mathbf{77.6 \pm 2.8}$	18.5
CPR (w/ Reset)	$\mathbf{99.1 \pm 0.7}$	**2.1**	74.6 ± 1.2	**1.9**

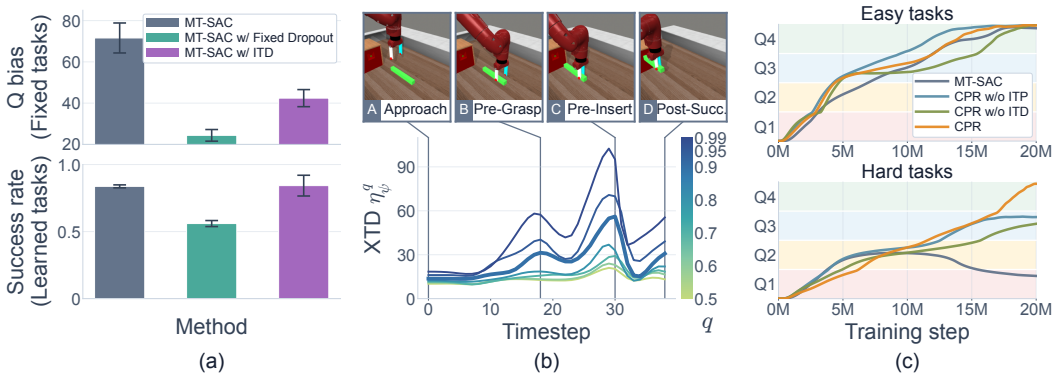

Figure 5: **Analysis of CPR.** (a) Q bias of the fixed-policy tasks (top) and success rate of the learned tasks (bottom). (b) XTD along a PEG-INSERT-SIDE trajectory for different expectile levels q. (c) Return–XTD trajectories of easy (top) and hard (bottom) tasks.

5.3 IN-DEPTH ANALYSIS OF CPR

We further analyze CPR on MT10, where hard tasks refer to PUSH, PEG-INSERT-SIDE, and PICK-PLACE, and easy tasks to the remaining seven. We examine whether inter-task dropout (ITD) mitigates task over-specialization, why upper-expectile XTD is important for intra-task prioritization (ITP), and how they jointly recover hard-task learning.

Mitigating Task Over-Specialization. We first examine whether ITD reduces Q bias caused by task over-specialization. Following Figure 3, we fix the policies of the three hard tasks to random policies and learn those of the seven easy tasks, while training the critic on all tasks. Figure 5(a) reports the Q bias of the fixed hard tasks (top) and the success rate of the learned easy tasks (bottom) under MT-SAC, MT-SAC with fixed dropout (w/ Fixed Dropout), and MT-SAC with ITD (w/ ITD), with dropout applied to the easy tasks. Both dropout variants substantially reduce Q bias, showing that regularizing the easy tasks mitigates their over-specialization in the shared critic. However, w/ Fixed Dropout considerably reduces their success rate, indicating that indiscriminate regularization can disrupt learning. In contrast, w/ ITD regularizes sufficiently learned tasks based on learning progress, achieving comparable Q-bias reduction while maintaining their success rate. Thus, ITD *mitigates task over-specialization while preserving learning performance*.

Why Upper-Expectile XTD for ITP? We next analyze XTD under standard MT10 training with CPR, focusing on PEG-INSERT-SIDE. We train XTD predictors η_ψ^q with different expectile levels q and examine their predictions along the trajectory in Figure 5(b), where A–D denote key stages of the trajectory. At higher q, XTD clearly increases around B (Pre-grasp) and C (Pre-insert), where substantial positive TD corrections remain for critical grasping and insertion behaviors. As q approaches 0.5, these increases become less pronounced because sparse positive TD errors are obscured by averaging. Since ITP prioritizes transitions according to XTD, q determines how strongly these critical states are emphasized. Consistently, Appendix E.1 examines the effect of q on performance, showing improvements from $q = 0.5$ to $q = 0.9$ followed by a slight decrease at $q \geq 0.95$, as increasingly extreme expectiles rely on fewer upper-tail samples. We therefore use $q = 0.9$, which distinguishes underestimated transitions while retaining enough samples for reliable XTD estimation. Thus, upper-expectile XTD enables ITP to *identify and emphasize otherwise obscured learning signals*.

Recovering Hard-Task Learning Progression. Finally, Figure 5(c) compares task learning progression under CPR, CPR w/o ITP, CPR w/o ITD, and MT-SAC. The seven easier tasks consistently progress toward Q4, whereas clear differences emerge for the three hard tasks. Under MT-SAC, these tasks initially move from Q1 toward Q2 as XTD increases, but fail to translate these signals into higher return and eventually return toward Q1. ITD mitigates Q bias from task over-specialization, while ITP emphasizes high-XTD transitions with informative value corrections. Either component enables progress toward Q3, but neither reaches Q4 within the training budget. Combining both in CPR allows the hard tasks to reach Q4 while preserving the convergence of the easier tasks. Overall, their complementary roles enable CPR to *recover hard-task learning*.

5.4 Ablation Study and Further Analysis

Ablation Study. Table 3 evaluates the contributions of ITD and ITP through variants. For ITD, we consider removing ITD, applying a fixed dropout rate regardless of learning progress (w/ Fixed Dropout), and determining dropout solely from the return rank without XTD (w/ Return-based Dropout). Removing ITD substantially reduces performance, while both alternatives perform worse than CPR. Fixed Dropout cannot selectively regularize tasks based on learning progress, whereas Return-based Dropout cannot determine whether high-return tasks have sufficiently converged without XTD. For ITP, removing ITP and replacing upper-expectile XTD with mean TD ($q = 0.5$) both substantially reduce performance, as mean TD can obscure sparse positive corrections. These results confirm the importance of XTD for both selective inter-task dropout and informative intra-task prioritization.

Table 3: **Component Analysis of CPR.**

Variant	Success Rate
CPR	97.0 ± 0.1
Inter-Task Dropout	
CPR w/o Inter-Task Dropout	88.9 ± 7.8
CPR w/ Fixed Dropout	79.2 ± 7.7
CPR w/ Return-based Dropout	82.8 ± 1.3
Intra-Task Prioritization	
CPR w/o Intra-Task Prioritization	81.5 ± 9.0
CPR w/ Mean TD	85.3 ± 2.6
Auxiliary Components	
CPR w/o Return Prioritization	95.8 ± 1.5
CPR w/o Gradient Normalization	90.2 ± 0.4
MT-SAC	73.7 ± 2.2

We also ablate two auxiliary components: return prioritization, which samples more high-return episodes, and gradient normalization, which normalizes task-wise gradient magnitudes. Removing return prioritization causes only a minor performance drop, suggesting that XTD-based transition prioritization is more important for selecting informative experience. Gradient normalization has a larger effect by balancing gradient scales across tasks, although its contribution remains smaller than those of ITD and ITP. Overall, these results show that *ITD and ITP provide the core complementary mechanisms of CPR*, while the auxiliary components provide additional improvements.

Additional Analysis. We provide further analyses in the appendix. In addition to the expectile-level analysis in Appendix E.1, Appendices E.2 and E.3 examine sensitivity to the maximum dropout rate $p_{\max}$ and prioritization strength α, respectively. Performance varies across these values but remains robust over a broad range. Appendix E.4 further visualizes XTD along trajectories from other MT10 tasks, showing similar patterns at behaviorally meaningful states. Finally, Appendix F analyzes the computational overhead of CPR and its individual components relative to MT-SAC, showing that the overall overhead remains modest at approximately 5%.

6 Limitations

ITD uses relative task returns to identify sufficiently learned tasks, which assumes comparable return scales across tasks. When return scales differ substantially, this limitation can be addressed using scale-independent measures such as success rates or by normalizing returns with predefined maximum rewards. In addition, despite substantial improvements on hard tasks, performance remains low on the lowest-performing tasks in MT50 (Worst-10). Since CPR primarily strengthens existing learning signals without relying on parameter resets, tasks with extremely sparse or insufficient signals may require additional exploration or task-specific mechanisms.

7 Conclusion

We show that hard-task learning in MTRL can be hindered by task over-specialization in the shared critic and weak learning signals within hard tasks. To address these issues, we introduce XTD, the upper expectile of signed TD errors, which reveals informative positive corrections that can be obscured by their average. Building on XTD, CPR combines ITD to mitigate task over-specialization with ITP to emphasize high-XTD transitions within hard tasks. Experiments on MT10 and MT50 show that these complementary mechanisms substantially improve hard-task learning and overall performance without relying on parameter resets. More broadly, our findings highlight the importance of jointly addressing inter-task over-specialization and intra-task learning signals, with XTD providing an effective signal for both ITD and ITP.

AI USE STATEMENT

Large language models were used only for translation, wording refinement, and readability improvement during manuscript preparation. They were not used for problem formulation, method development, experimental design or execution, or result analysis and interpretation. All scientific content and technical decisions were developed by the authors, who reviewed and revised all AI-assisted text.

ETHICS STATEMENT

This work is methodological and evaluates multi-task reinforcement learning entirely in simulated environments using publicly available benchmarks. It involves no human subjects, personally identifiable information, or sensitive data, and introduces no substantial ethical concerns beyond those generally associated with reinforcement learning research.

REPRODUCIBILITY STATEMENT

Section 4 presents the objectives and training procedure of CPR. Appendix A provides implementation details, while Appendices B and C describe the experimental settings, baselines, and hyperparameters. We also provide anonymized source code in the supplementary material to facilitate reproduction of our results.

REFERENCES

Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 166–175. PMLR, 2017.

Nils Bjorck, Carla P Gomes, and Kilian Q Weinberger. Towards deeper deep reinforcement learning with spectral normalization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8242–8255, 2021.

Jongseong Chae, Jongeui Park, Yongjae Shin, Gyeongmin Kim, Seungyul Han, and Youngchul Sung. Flow actor-critic for offline reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 794–803. PMLR, 2018.

Guangran Cheng, Lu Dong, Wenzhe Cai, and Changyin Sun. Multi-task reinforcement learning with attention-based mixture of experts. *IEEE Robotics and Automation Letters*, 8(6):3812–3819, 2023.

Myungsik Cho, Jongeui Park, Suyoung Lee, and Youngchul Sung. Hard tasks first: Multi-task reinforcement learning through task scheduling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8556–8577. PMLR, 2024.

Myungsik Cho, Jongeui Park, Jeonghye Kim, and Youngchul Sung. ARS: Adaptive reward scaling for multi-task reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10583–10601. PMLR, 2025.

Cédric Colas, Pierre Fournier, Mohamed Chetouani, Olivier Sigaud, and Pierre-Yves Oudeyer. CURIOUS: Intrinsically motivated modular multi-goal reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1331–1340. PMLR, 2019.

Brett Daley and Christopher Amato. Reconciling λ-returns with experience replay. *Advances in Neural Information Processing Systems*, 32, 2019.

Tim De Bruin, Jens Kober, Karl Tuyls, and Robert Babuška. Experience selection in deep reinforcement learning for control. *Journal of Machine Learning Research*, 19(9):1–56, 2018.

Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2169–2176. IEEE, 2017.

Mohamed Elsayed, Qingfeng Lan, Clare Lyle, and A. Rupam Mahmood. Weight clipping for deep continual and reinforcement learning. In *Reinforcement Learning Conference*, 2024.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Clement Gehring and Doina Precup. Smart exploration in reinforcement learning using absolute temporal difference errors. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 1037–1044, 2013.

Florin Gogianu, Tudor Berariu, Mihaela C Rosca, Claudia Clopath, Lucian Busoniu, and Razvan Pascanu. Spectral normalisation for deep reinforcement learning: An optimisation perspective. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3734–3744. PMLR, 2021.

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.

Seungyul Han and Youngchul Sung. Diversity actor-critic: Sample-aware entropy regularization for sample-efficient exploration. In *International conference on machine learning*, pp. 4018–4029. PMLR, 2021a.

Seungyul Han and Youngchul Sung. A max-min entropy framework for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:25732–25745, 2021b.

Jinmin He, Kai Li, Yifan Zang, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. Efficient multi-task reinforcement learning with cross-task policy guidance. In *Advances in Neural Information Processing Systems*, volume 37, pp. 117997–118024, 2024a. doi: 10.52202/079017-3747.

Jinmin He, Kai Li, Yifan Zang, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. Not all tasks are equally difficult: Multi-task deep reinforcement learning with dynamic depth routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12376–12384, 2024b.

Jinmin He, Kai Li, Xiaoyi Dong, Yifan Zang, Yuheng Jing, Yifan Zhang, Junliang Xing, and Jian Cheng. HyMTRL: A hybrid multi-task reinforcement learning framework via phased policy evolution. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Qiang He, Tianyi Zhou, Meng Fang, and Setareh Maghsudi. Adaptive regularization of representation rank as an implicit constraint of Bellman equation. In *The Twelfth International Conference on Learning Representations*, 2024c.

Ahmed Hendawy, Jan Peters, and Carlo D'Eramo. Multi-task reinforcement learning with mixture of orthogonal experts. In *International Conference on Learning Representations*, 2024.

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with PopArt. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3796–3803, 2019. doi: 10.1609/aaai.v33i01.33013796.

Hanchi Huang, Deheng Ye, Li Shen, and Wei Liu. Curriculum-based asymmetric multi-task reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6): 7258–7269, 2023. doi: 10.1109/TPAMI.2022.3223872.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pp. 4940–4950. PMLR, 2021.

Jeongmo Kim, Yisak Park, Minung Kim, and Seungyul Han. Task-aware virtual training: Enhancing generalization in meta-reinforcement learning for out-of-distribution tasks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30720–30748. PMLR, 2025.

Yilun Kong, Guozheng Ma, Qi Zhao, Haoyu Wang, Li Shen, Xueqian Wang, and Dacheng Tao. Mastering massive multi-task reinforcement learning via mixture-of-expert decision transformer. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31379–31396. PMLR, 2025.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Abhishek Gupta, and Sergey Levine. Discor: Corrective feedback in reinforcement learning via distribution correction. *Advances in neural information processing systems*, 33: 18560–18572, 2020a.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020b.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations*, 2021.

Aviral Kumar, Rishabh Agarwal, Tengyu Ma, Aaron Courville, George Tucker, and Sergey Levine. DR3: Value-based deep reinforcement learning requires explicit regularization. In *International Conference on Learning Representations*, 2022.

Siming Lan, Rui Zhang, Qi Yi, Jiaming Guo, Shaohui Peng, Yunkai Gao, Fan Wu, Ruizhi Chen, Zidong Du, Xing Hu, Xishan Zhang, Ling Li, and Yunji Chen. Contrastive modules with temporal attention for multi-task reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 36507–36523, 2023. doi: 10.52202/075280-1585.

Sanghyeon Lee, Sangjun Bae, Yisak Park, and Seungyul Han. Self-improving skill learning for robust skill-based meta-reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Pengyi Li, HAO Jianye, Hongyao Tang, Yan Zheng, and Fazl Barez. Value-evolutionary-based reinforcement learning. In *Forty-first International Conference on Machine Learning*, 2024.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18878–18890, 2021a.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. FAMO: Fast adaptive multitask optimization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 57226–57243, 2023. doi: 10.52202/075280-2500.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International Conference on Learning Representations*, 2021b.

Xu-Hui Liu, Zhenghai Xue, Jingcheng Pang, Shengyi Jiang, Feng Xu, and Yang Yu. Regret minimization experience replay in off-policy reinforcement learning. *Advances in neural information processing systems*, 34:17604–17615, 2021c.

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. In *International Conference on Learning Representations*, 2022.

Haozhe Ma, Zhengding Luo, Thanh Vinh Vo, Kuankuan Sima, and Tze-Yun Leong. Centralized reward agent for knowledge sharing and transfer in multi-task reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 163080–163103, 2025. doi: 10.52202/085713-5444.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16428–16446. PMLR, 2022.

Whitney K. Newey and James L. Powell. Asymmetric least squares estimation and testing. *Econometrica*, 55(4):819–847, 1987. ISSN 00129682, 14680262.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16828–16847. PMLR, 2022.

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto. Deep reinforcement learning with plasticity injection. In *Advances in Neural Information Processing Systems*, volume 36, pp. 37142–37159, 2023. doi: 10.52202/075280-1614.

Motoki Omura, Takayuki Osa, Yusuke Mukuta, and Tatsuya Harada. Symmetric q-learning: Reducing skewness of bellman error in online reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14474–14481, 2024.

Yangchen Pan, Jincheng Mei, Amir-massoud Farahmand, Martha White, Hengshuai Yao, Mohsen Rohani, and Jun Luo. Understanding and mitigating the limitations of prioritized experience replay. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pp. 1561–1571. PMLR, 2022.

Leonard S Pleiss, Tobias Sutter, and Maximilian Schiffer. Reliability-adjusted prioritized experience replay. In *International Conference on Learning Representations*, volume 2026, pp. 59206–59232, 2026.

Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep RL in continuously parameterized environments. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 835–853. PMLR, 2020.

Alex Rutherford, Michael Beukman, Timon Willi, Bruno Lacerda, Nick Hawes, and Jakob Foerster. No regrets: Investigating and improving regret approximations for curriculum discovery. In *Advances in Neural Information Processing Systems*, volume 37, pp. 16071–16101, 2024.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

Arsalan Sharifnassab, Mohamed Elsayed, Kris De Asis, A. Rupam Mahmood, and Richard S Sutton. Intentional updates for streaming reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026.

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9767–9779. PMLR, 2021.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32145–32168. PMLR, 2023.

Shivakanth Sujit, Somjit Nath, Pedro Braga, and Samira Ebrahimi Kahou. Prioritizing samples in reinforcement learning with reducible loss. *Advances in Neural Information Processing Systems*, 36:23237–23258, 2023.

Lingfeng Sun, Haichao Zhang, Wei Xu, and Masayoshi Tomizuka. PaCo: Parameter-compositional multi-task reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1562.

Yee Whye Teh, Victor Bapst, Wojciech M. Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4496–4506, 2017.

Sebastian Thrun and Anton Schwartz. Finding structure in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 7, pp. 385–392. MIT Press, 1994.

Peiyao Xiao, Hao Ban, and Kaiyi Ji. Direction-oriented multi-objective learning: Simple and provable stochastic algorithms. In *Advances in Neural Information Processing Systems*, volume 36, pp. 4509–4533, 2023. doi: 10.52202/075280-0200.

Ruihan Yang, Huazhe Xu, Yi Wu, and Xiaolong Wang. Multi-task reinforcement learning with soft modularization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4767–4777, 2020.

Junghyuk Yeom, Yonghyeon Jo, Jungmo Kim, Sanghyeon Lee, and Seungyul Han. Exclusively penalized q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:113405–113435, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020a.

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, 2020b.

Wenshuai Zhao, Zhiyuan Li, and Joni Pajarinen. Learning progress driven multi-agent curriculum. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 77572–77588. PMLR, 13–19 Jul 2025.

A Implementation Details

MT-SAC Losses. Soft Actor-Critic (SAC) is an off-policy actor–critic method that augments the expected return with an entropy objective (Haarnoja et al., 2018). MT-SAC extends SAC to multiple tasks by conditioning the shared actor π_θ and critic Q_ϕ on the state and one-hot task identifier z_i. In the notation of the main text, the signed TD error is

$$\delta_i = \mathrm{sg}\big[r + \gamma(1-d)\left(Q_{\bar{\phi}}(s', a', z_i) - \lambda_i \log \pi_\theta(a' \mid s', z_i)\right)\big] - Q_\phi(s, a, z_i), \quad a' \sim \pi_\theta(\cdot \mid s', z_i), \tag{A.1}$$

where d indicates termination, λ_i is the task-specific entropy temperature, and sg denotes stop-gradient. For equal-sized per-task minibatches B^i, the critic and actor losses are

$$\begin{aligned} \mathcal{L}_Q(\phi) &= \frac{1}{N}\sum_{i=0}^{N-1} \frac{1}{|B^i|} \sum_{(s,a,r,s',d)\in B^i} \delta_i^2, \\ \mathcal{L}_\pi(\theta) &= \frac{1}{N}\sum_{i=0}^{N-1} \frac{1}{|B^i|} \sum_{(s,a,r,s',d)\in B^i} \mathbb{E}_{\tilde{a}\sim\pi_\theta(\cdot|s,z_i)}\left[\lambda_i \log \pi_\theta(\tilde{a} \mid s, z_i) - Q_\phi(s, \tilde{a}, z_i)\right]. \end{aligned} \tag{A.2}$$

We use twin critics: their minimum defines both the Bellman target and the actor objective, and both critics are fitted to the same target. Actor updates backpropagate through sampled actions but not through critic parameters. Each λ_i is initialized to 1 and optimized independently with Adam (learning rate 3×10^{-4}) toward a target entropy of -4. Target critics are updated by Polyak averaging, as in SAC.

Inter-Task Dropout. For each task i, we maintain EMAs of completed-episode returns $\bar{R}^i$ and minibatch-mean XTD predictions $\bar{\eta}^i$. Their decay coefficients are 0.9 and 0.99, respectively, so new observations receive weights 0.1 and 0.01. Both EMAs and the historical XTD peak $\bar{\eta}^i_{\max}$ are initialized to zero. We track the peak as described in Section 4.2, setting the decline factor in Eq. (4) to zero until $\bar{\eta}^i_{\max} > \epsilon$.

We disable dropout for the first 100,000 critic updates on MT10 and 50,000 on MT50, then linearly increase it to the prescribed rate over the next 100,000 updates. The task-specific mask is applied to the online and target critics during critic optimization, including target computation, and to the actor during policy optimization.

Intra-Task Prioritization. We train $\eta^q_\psi(s, z_i)$ with Eq. (3). Its targets are the detached, signed, and unweighted TD errors δ_i computed before applying dropout; this loss updates only ψ.

To construct priorities, we process the XTD predictions separately within each task minibatch. We first lower-clip them at $\epsilon = 10^{-6}$, then standardize them using the within-task mean and standard deviation (with an ϵ floor), and retain only the positive part. Substituting the processed values into Eq. (5), we clip the linear priorities $1 + \alpha\eta^q_\psi(s, z_i)$ to $[1, 2]$ and normalize them to unit mean. Consequently, the final weights $w^i(s)$ can be below one, while $\alpha = 0$ or constant predictions produce uniform weights.

The weights are detached and applied only to the critic loss in Eq. (6); the actor and temperature losses remain unweighted. This preprocessing is used only for prioritization and does not alter XTD regression or task-level XTD statistics.

Reset Variant. The reset variant restores the actor and online critics to their initial parameters and then synchronizes the target critics. It also resets each task-specific entropy temperature to 1 and reinitializes the actor, critic, and temperature optimizers. We retain the XTD estimator and its optimizer, the return and XTD statistics (including historical XTD peaks), and the replay buffers. The global training schedule continues, without repeating warm-up.

B EXPERIMENTAL SETUP

This section provides the experimental details needed to interpret and reproduce our comparisons on the Meta-World benchmarks. In Section B.1, we describe the MT10 and MT50 task suites, including their observation and action spaces. In Section B.2, we detail the baseline implementations used to ensure controlled comparisons with our method.

B.1 ENVIRONMENTAL DETAILS

Meta-World MT10 and MT50 (Yu et al., 2020b) contain 10 and 50 Sawyer manipulation tasks with shared observation and action spaces. The observation $s \in \mathbb{R}^{39}$ combines current and previous end-effector, gripper, and object information with the goal position;[1] appending z_i gives 49- and 89-dimensional inputs, respectively. The action $a \in [-1, 1]^4$ controls Cartesian end-effector displacement and the gripper. Task-specific dense rewards $r_i(s, a)$, typically in $[0, 10]$, encourage reaching, grasping, and goal completion as applicable; evaluation uses a separate success indicator. Figure B.1 shows the MT10 environments. Tables B.1 and B.2 list task names and indices in each benchmark's dictionary order.

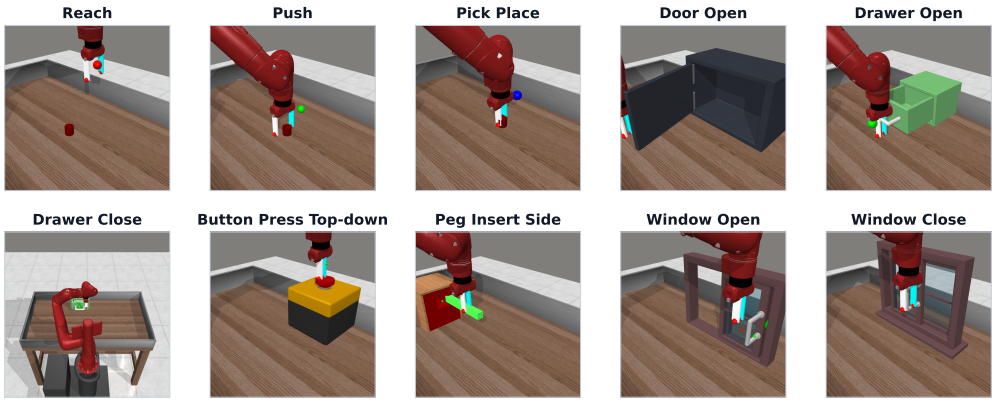

Figure B.1: The Meta-World MT10 environments.

Table B.1: Task indices and names in Meta-World MT10.

Task ID	Task name	Task ID	Task name
0	reach-v2	5	drawer-close-v2
1	push-v2	6	button-press-topdown-v2
2	pick-place-v2	7	peg-insert-side-v2
3	door-open-v2	8	window-open-v2
4	drawer-open-v2	9	window-close-v2

Table B.2: Task indices and names in Meta-World MT50.

ID	Task name	ID	Task name	ID	Task name
0	assembly-v2	17	hand-insert-v2	34	pick-place-v2
1	basketball-v2	18	drawer-close-v2	35	plate-slide-v2
2	bin-picking-v2	19	drawer-open-v2	36	plate-slide-side-v2
3	box-close-v2	20	faucet-open-v2	37	plate-slide-back-v2
4	button-press-topdown-v2	21	faucet-close-v2	38	plate-slide-back-side-v2
5	button-press-topdown-wall-v2	22	hammer-v2	39	peg-unplug-side-v2
6	button-press-v2	23	handle-press-side-v2	40	soccer-v2
7	button-press-wall-v2	24	handle-press-v2	41	stick-push-v2
8	coffee-button-v2	25	handle-pull-side-v2	42	stick-pull-v2
9	coffee-pull-v2	26	handle-pull-v2	43	push-wall-v2
10	coffee-push-v2	27	lever-pull-v2	44	reach-wall-v2
11	dial-turn-v2	28	peg-insert-side-v2	45	shelf-place-v2
12	disassemble-v2	29	pick-place-wall-v2	46	sweep-into-v2
13	door-close-v2	30	pick-out-of-hole-v2	47	sweep-v2
14	door-lock-v2	31	reach-v2	48	window-open-v2
15	door-open-v2	32	push-back-v2	49	window-close-v2
16	door-unlock-v2	33	push-v2		

[1] V2 implementation: https://github.com/Farama-Foundation/Metaworld/tree/v2.0.0/metaworld/envs/mujoco/sawyer_xyz.

B.2 Baseline Descriptions

We refer to `metaworld-algorithms`[2] for the baselines except ARS, HyMTRL, and SMT, and use the authors' implementations for ARS[3] and HyMTRL[4]. For controlled comparisons, ARS and HyMTRL use MT-SAC without additional baseline-specific architectures, allowing their training mechanisms to be compared on the same base learner.

MT-SAC and MT-MH-SAC (Haarnoja et al., 2018; Yang et al., 2020). The shared-network baseline conditions its actor and critic on observations and one-hot task identifiers, averaging SAC losses across tasks (Appendix A). Its multi-head variant adds independent task-specific output heads to a shared trunk, retaining shared representations while separating task-specific predictions.

PCGrad (Yu et al., 2020a). To reduce gradient interference, per-task actor and critic gradients are processed in randomized order, projecting out components that conflict with other task gradients before aggregation. Non-conflicting pairs remain unchanged. This modifies the optimizer update without introducing task-specific network components.

Soft Modules (Yang et al., 2020). A state- and task-conditioned routing network softly combines shared modules in both actor and critic. Jointly learning the modules and routing permits different sharing patterns across states and tasks. Actor and critic task losses are additionally weighted by a softmax over negative task-specific SAC temperatures.

MOORE (Hendawy et al., 2024). Shared expert outputs are orthonormalized via Gram–Schmidt and combined with task-dependent coefficients in both actor and critic. This encourages diversity in feature space while allowing each task to use a different combination of shared expert representations.

PaCo (Sun et al., 2022). Task-specific actor and critic parameters are linear combinations of shared parameter sets, with a learned composition vector for each task. For stability, exploding task losses are excluded and their composition vectors are reinitialized from non-exploding tasks. This task-local recovery differs from resetting entire actor or critic networks.

SMT (Cho et al., 2024). This reset-based baseline combines difficulty-aware task scheduling with periodic network reinitialization to mitigate simplicity bias. Scheduling allocates more training resources to harder tasks, reducing interference from easier ones. We use the authors' reported results rather than retraining this method under our matched settings.

ARS (Cho et al., 2025). This reset-based baseline combines adaptive reward scaling with scheduled actor and critic reinitialization. Each task's rewards are scaled by the largest replay-buffer mean reward across tasks divided by its own mean, amplifying smaller reward signals in the Bellman target. Scales are recomputed at reset points and fixed between them, while replay buffers are retained. Our no-reset variant omits only network reinitialization, preserving the reward-scaling rule and update schedule.

HyMTRL (He et al., 2026). This reset-based framework combines phased policy evolution with critic reinitialization while preserving the policy and replay buffer. Among its variants, we use the MT-SAC-based variant for a fair comparison with our setting. Mastered tasks transition from RL to imitation learning and are excluded from critic optimization, consolidating their policies while focusing value learning on the remaining tasks. Our no-reset variant removes only critic reinitialization while retaining phased policy evolution and all other settings.

[2] `https://github.com/rainx0r/metaworld-algorithms`
[3] `https://github.com/ms-cho/Adaptive-Reward-Scaling-ARS-for-MTRL`
[4] `https://openreview.net/forum?id=Jo7Mt1XbuQ`

C HYPERPARAMETER SETUP

Table C.1 summarizes the shared training settings and method-specific hyperparameters used in our experiments. Training budgets count the total environment transitions collected across tasks, while batch sizes are specified per task. Method-specific settings include modular architectures, expert counts, parameter sets, and reset configurations. For our method, we use $q = 0.9$ and $p_{\max} = 0.05$ on both benchmarks, with prioritization strength $\alpha = 0.1$ on MT10 and $\alpha = 0.5$ on MT50. Our method does not reset network parameters by default; the listed reset count applies only to its reset variant. For ARS on MT50, we use a replay buffer of 25M transitions, as the shared 5M setting yielded substantially lower performance.

Table C.1: Hyperparameter settings for all methods in the Meta-World environments

Group	Hyperparameter	MT10	MT50
Shared Hyperparameters	Baseline algorithm	MT-SAC	MT-SAC
	Training steps	20M	100M
	Buffer size	10M	5M
	Batch size	100 per task	
	Episode length	500	
	Discount factor	0.99	
	Network depth (critic, actor)	4	
	Network width (critic, actor)	400	
	Learning rate (critic, actor, temperature λ)	3×10^{-4}	
Soft Modules	# of modules	2	
	# of layers per module	2	
	Module width	256	
	Embedding dimension	400	
MOORE	# of experts	4	6
	# layers per expert	3	
	Layer width	400	
PaCo	# of parameter sets	5	
ARS	Buffer size	10M	25M
	# of resets	4	6
HyMTRL	# of resets	4	6
	Mastery threshold	0.9	
CPR	Intra-task prioritization strength α	0.1	0.5
	Expectile level q	0.9	
	XTD estimator learning rate	3×10^{-4}	
	XTD estimator network depth	4	
	XTD estimator network width	400	
	Maximum dropout probability $p_{\max}$	0.05	
	# of resets (w/ reset variant)	4	

D PER-TASK PERFORMANCE ON MT10 AND MT50

Figures D.1 and D.2 present per-task success-rate curves on MT10 and MT50, respectively, complementing the aggregate results in Section 5.2. ARS, HyMTRL, and our method (CPR) are all evaluated without parameter resets in these figures. On MT10, CPR improves learning on challenging tasks such as PUSH and PICK-PLACE, while maintaining high success rates on easier tasks. On MT50, gains are particularly evident on BOX-CLOSE, PICK-PLACE-WALL, and STICK-PUSH.

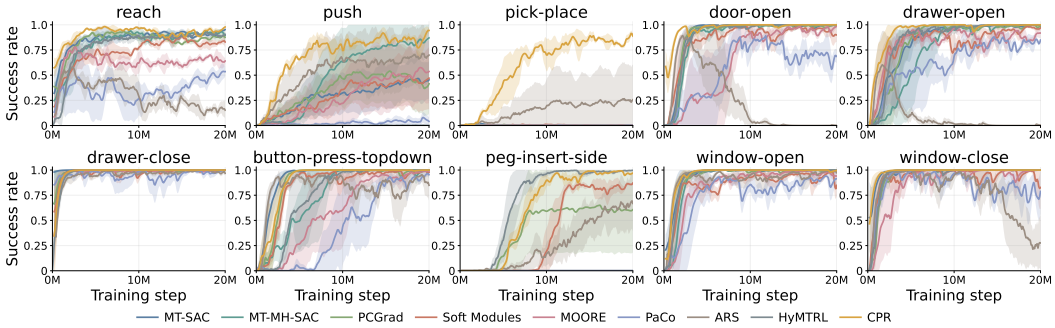

Figure D.1: Per-task performance on Meta-World MT10. ARS, HyMTRL, and CPR are evaluated without parameter resets.

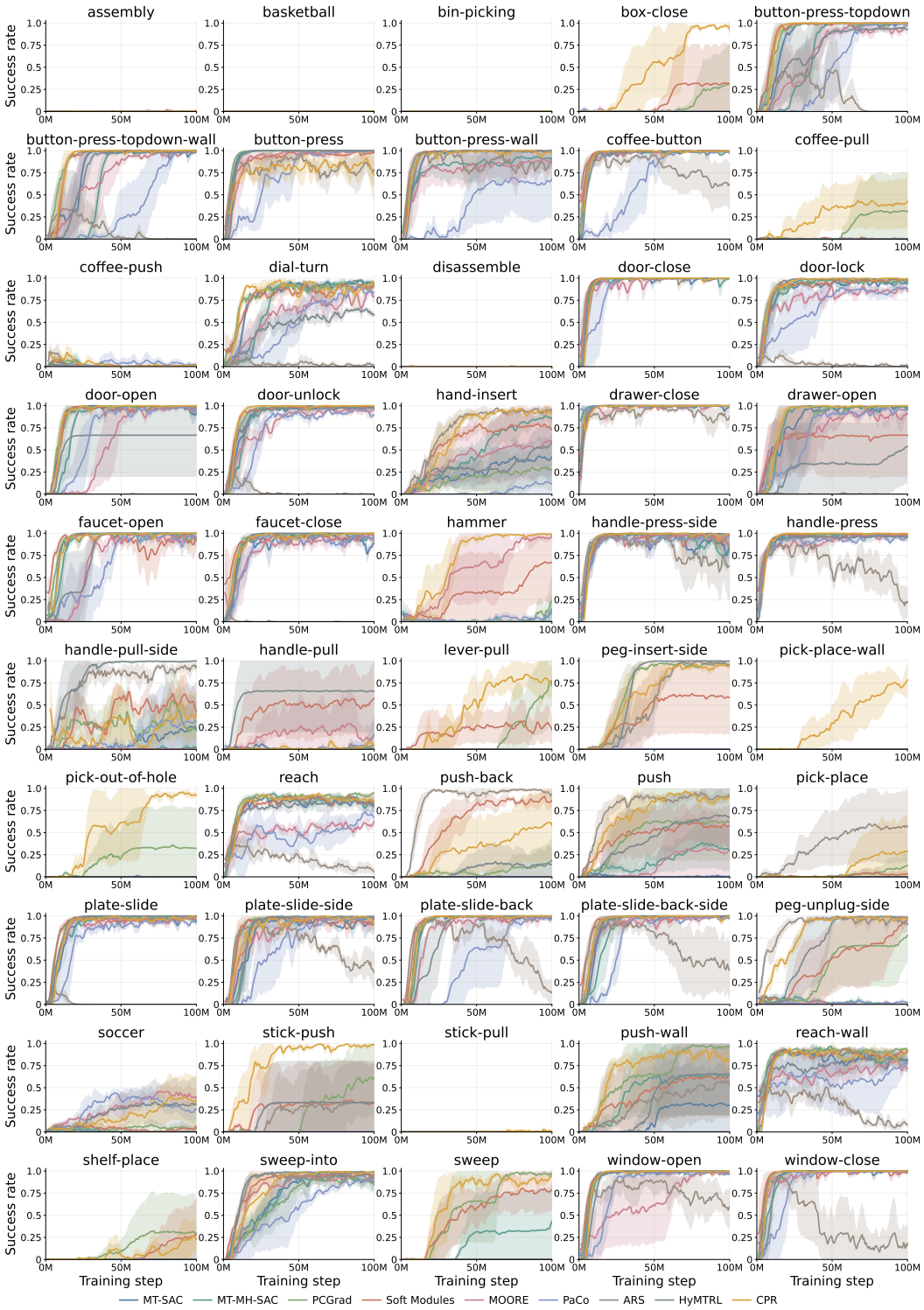

Figure D.2: Per-task performance on Meta-World MT50. ARS, HyMTRL, and CPR are evaluated without parameter resets.

E Further Analysis Results

This section provides additional analyses of CPR. Appendix E.1 studies the expectile level q, Appendix E.2 analyzes the effect of the maximum dropout rate $p_{\max}$, Appendix E.3 examines the XTD prioritization strength α, and Appendix E.4 examines XTD trajectories across MT10 tasks.

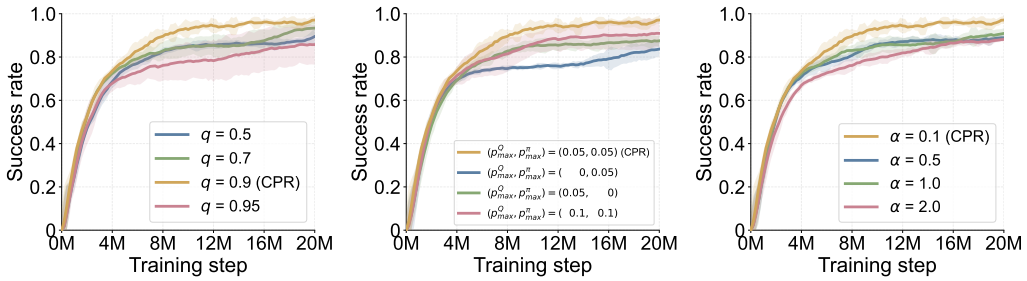

Figure E.1: Hyperparameter sensitivity on MT10. *Left:* Expectile level q. *Middle:* Maximum dropout rate $p_{\max}$. *Right:* XTD prioritization strength α.

E.1 Effect of the Expectile Level q

We compare $q \in \{0.5, 0.7, 0.9, 0.95\}$ to assess how the expectile level affects learning. Figure E.1 (left) shows that $q = 0.9$ achieves the highest final success rate among the tested values. The mean-based setting $q = 0.5$ and the lower upper-expectile setting $q = 0.7$ learn more slowly, while increasing q to 0.95 also reduces performance. This pattern complements the trajectory analysis in Figure 5 (middle): upper expectiles reveal positive TD corrections near critical interaction stages that are less apparent in the conditional mean. However, emphasizing increasingly extreme residuals does not necessarily produce a more useful learning signal. The results support $q = 0.9$, which highlights informative positive corrections in the upper tail while achieving the best performance among the tested levels. Since XTD informs both inter-task dropout and intra-task prioritization, this sweep measures the effect of q on the combined framework.

E.2 Effect of the Dropout Rate

Figure E.1 (middle) compares four dropout configurations, where $p^Q_{\max}$ and $p^\pi_{\max}$ denote the maximum dropout rates for the critic and actor, respectively. The task-specific rate increases when a task ranks higher in return and its XTD declines further from its peak, assigning stronger regularization to sufficiently learned tasks. Among $(p^Q_{\max}, p^\pi_{\max}) \in \{(0.05, 0.05), (0, 0.05), (0.05, 0), (0.1, 0.1)\}$, the configuration $(0.05, 0.05)$ achieves the highest final success rate. Removing critic dropout ($p^Q_{\max} = 0$) causes the largest performance reduction, consistent with the analysis in Section 5.3 showing that dropout mitigates task over-specialization. Removing actor dropout ($p^\pi_{\max} = 0$) also reduces performance, suggesting that actor regularization provides an additional benefit alongside critic dropout. Increasing both maximum rates to 0.1 yields lower final performance than 0.05. Together with Section 5.3, these results highlight the importance of controlling the application and strength of dropout: stronger regularization does not necessarily improve learning.

E.3 Effect of the XTD Prioritization Strength α

We vary $\alpha \in \{0.1, 0.5, 1.0, 2.0\}$ to examine the strength of intra-task XTD prioritization. Figure E.1 (right) shows that $\alpha = 0.1$ achieves the highest final success rate, while larger values generally slow learning and lead to lower final performance. This suggests that a modest emphasis on high-XTD states is sufficient to improve critic learning, and that stronger reweighting is not necessarily beneficial. As detailed in Appendix A, priorities are constructed from standardized XTD predictions, clipped, and normalized to unit mean within each task. Thus, α controls the relative weighting of transitions rather than the overall scale of the critic loss, and its effect is bounded by clipping. Together with the lower performance obtained without intra-task prioritization in Table 3, these results support using moderate XTD-based weighting.

E.4 PER-TASK XTD TRAJECTORIES ON MT10

Following the trajectory-based analysis in Figure 2, we examine all ten MT10 tasks to assess whether high XTD identifies behaviorally meaningful states beyond PEG-INSERT-SIDE. Figure E.2 shows four stages of a successful trajectory for each task: (1) early, (2) high-XTD, (3) post-peak, and (4) success. For most tasks, the high-XTD state (2) captures a critical configuration just before grasping or initiating the key object interaction. This pre-grasp pattern is particularly evident in PICK-PLACE and PEG-INSERT-SIDE; tasks that do not require grasping instead highlight preparation for task-relevant contact. These qualitative examples support XTD as a signal of meaningful interaction stages across tasks, rather than only along the trajectory in the main text.

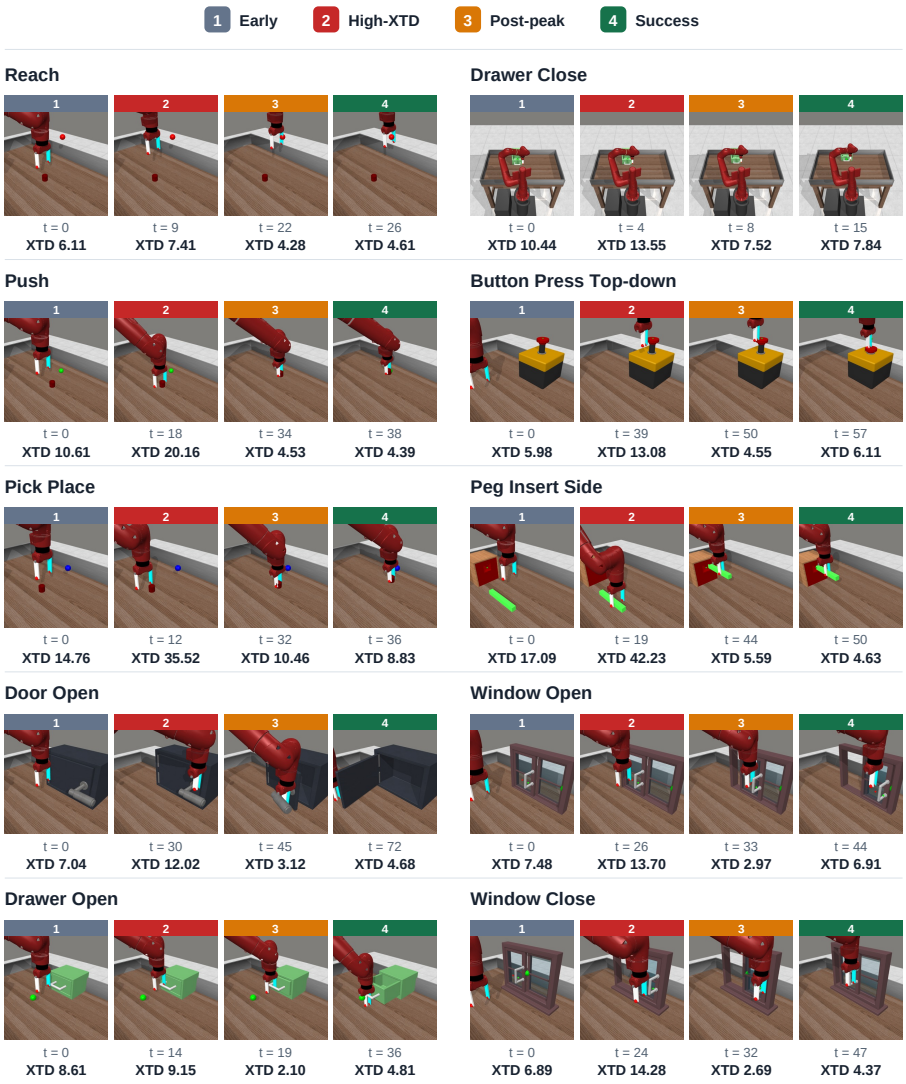

Figure E.2: **High-XTD states across all MT10 tasks.** Each task shows (1) early, (2) high-XTD, (3) post-peak, and (4) success states, with the timestep and XTD below each frame.

F COMPUTATIONAL COMPLEXITY

We compare performance and computational cost on MT10 in Table F.1. All methods were measured using the same hardware configuration. In addition to the final success rate, we report wall-clock training time to quantify the overhead of CPR.

Table F.1: Computational-cost comparison on MT10.

Method	Success Rate (%)	Wall-Clock Time (hours)
MT-SAC	73.7 ± 2.2	18.98
CPR w/o Intra-Task Prioritization	81.5 ± 9.0	19.32
CPR w/o Inter-Task Dropout	88.9 ± 7.8	19.43
CPR	97.0 ± 0.1	19.90

CPR requires 19.90 hours of training, compared with 18.98 hours for MT-SAC, corresponding to an increase of only approximately 4.8% in wall-clock time. The two ablated variants also remain close to the MT-SAC training time, requiring 19.32 and 19.43 hours. Thus, the auxiliary XTD estimator and the two control mechanisms introduce only modest computational overhead. In contrast, the full method improves the final success rate from 73.7% to 97.0%, indicating that the substantial performance gain is achieved with little increase in overall training time.