Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

DA-MRL: Dual-Arm Meta-Reinforcement Learning for Fast Adaptation in Dynamic Obstacle Avoidance

Lingxiao Hu

lingxiao.hu@chd.edu.cn

Chang'an University https://orcid.org/0009-0008-9366-4581

Hong Wang

Yang Zhang

Kai Ding

Research Article

Keywords:

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11389803/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

DA-MRL: Dual-Arm Meta-Reinforcement Learning for Fast Adaptation in Dynamic Obstacle Avoidance

Lingxiao Hu[1], Hong Wang[3,4], Yang Zhang[2*], Kai Ding[1]

[1] School of Mechanical Engineering and Intelligent Equipment, Chang'an University, Xi'an, 710064, Shaanxi, China .
[2] Department of Mechanical and Aerospace Engineering, University of Missouri, Columbia, 65211, MO, USA .
[3] School of Artificial Intelligence, Hubei University, Wuhan, 430062, Hubei, China .
[4] Key Laboratory of Intelligent Sensing System and Security (Ministry of Education), Hubei University, Wuhan, 430062, Hubei, China .

*Corresponding author(s). E-mail(s): zhangy1@missouri.edu;
Contributing authors: lingxiao.hu@chd.edu.cn; hong.wang@hubu.edu.cn; kding@chd.edu.cn;

Abstract

Dynamic obstacle avoidance in a shared dual-arm workspace requires coordinated motion and rapid adaptation to changing task conditions. This paper presents dual-arm meta-reinforcement learning (DA-MRL), a centralized framework that learns a task-generalized actor–critic initialization for adaptation to unseen target, obstacle, sensing, and dynamic conditions using limited support trajectories. MuJoCo experiments evaluate held-out in-distribution (ID) and out-of-distribution (OOD) tasks in both three-degree-of-freedom (3-DOF) and six-degree-of-freedom (6-DOF) systems. After ten adaptation updates, DA-MRL increases OOD collision-free success from 36.0% to 68.4% in 3-DOF and from 28.6% to 55.1% in 6-DOF, while reducing the corresponding collision rates. The adapted 6-DOF policy is then frozen and transferred to a physical dual-UR5 platform with red–green–blue and depth (RGB-D) perception. None of the six continuous 200-s deployment records, each containing 20,000 logged samples, encountered the defined collision-induced termination condition. These results demonstrate rapid policy adaptation in simulation and coordinated sim-to-real execution under the evaluated dynamic obstacle conditions.

Keywords: human-robot interaction, dual-arm robot system, meta-reinforcement learning, sim-to-real deployment

1 Introduction

Human–robot interaction (HRI) increasingly places robots, operators, and equipment within the same manufacturing workspace. In such environments, robots must maintain safe separation from surrounding objects while preserving task continuity. Dynamic safety zones update admissible robot motion according to human

movement [21], while vision-based tracking estimates obstacle states before planned motions are compromised [5]. Therefore, dynamic obstacle avoidance becomes a closed-loop control problem in which robot actions must be continuously adjusted according to workspace variations. Figure 1 illustrates the shared workspace considered in this study, where dual manipulators, operators, and surrounding equipment coexist. Each arm must approach its assigned target while avoiding moving obstacles, fixed structures, and potential inter-arm interference. A motion adjustment that improves one arm's clearance may reduce the maneuvering space of the other, requiring coordinated policies that consider target progress, safety margins, and changing task conditions.

The experimental task represents a class of dual-arm manufacturing operations, such as assembly, inspection, and pick-and-place, where two manipulators move toward separate working regions within a constrained workspace. The target regions represent task-dependent manipulation locations, whereas moving obstacles represent transient workspace occupancy requiring online motion adjustment. This study focuses on the collision-avoidance and rapid-adaptation layer rather than the end-effector process itself. To achieve reliable execution, target and obstacle states must be related to link-level clearance, joint constraints, and goal progress. Depth-based approaches maintain human–robot separation through range sensing [4], while collision detection and reaction mechanisms address physical contact events that require immediate responses [6]. These studies emphasize that safety should be integrated into the control loop rather than treated as a post-processing procedure.

Existing obstacle-avoidance methods mainly rely on model-based or learning-based strategies. Model-based approaches formulate geometric constraints through redundancy resolution, artificial potential fields, task-priority control, trajectory optimization, and predictive control [8, 9, 13, 15, 22, 27, 28]. However, their performance depends on accurate robot models, workspace descriptions, and assumptions about obstacle evolution. Learning-based methods learn observation–action mappings through interaction, with deep reinforcement learning and hybrid

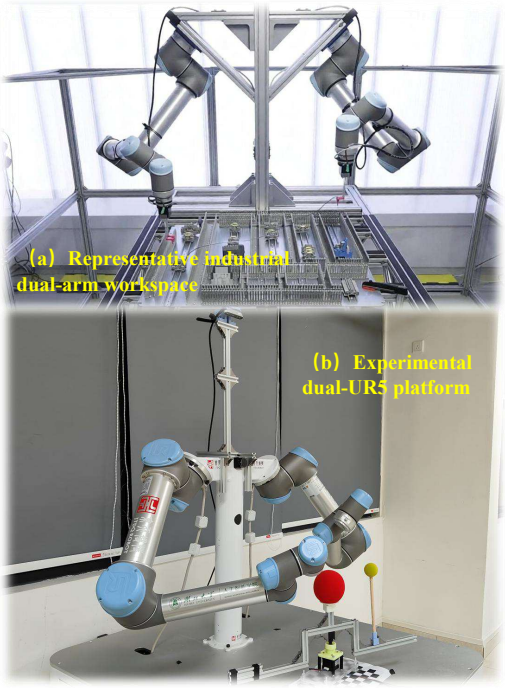


Fig. 1 Dual-arm manipulation settings considered in this study. (a) A representative industrial dual-arm workspace that motivates the dynamic obstacle-avoidance problem. (b) The dual-UR5 platform used for sim-to-real evaluation.

planning-control frameworks demonstrating adaptive collision-avoidance capabilities [1, 20, 25]. Nevertheless, most existing approaches optimize policies within fixed task distributions and provide limited mechanisms for rapidly adapting coordinated dual-arm behaviors under changing obstacle motions, target configurations, sensing conditions, delays, and dynamics. This motivates an adaptation-oriented framework capable of learning reusable initial policies and specializing them to new dual-arm tasks with limited interaction data.

DA-MRL formulates dynamic dual-arm obstacle avoidance as a task-adaptation problem over a distribution of related tasks. Inspired by gradient-based meta-learning principles, the framework learns task-generalized policy initializations that can be specialized to unseen conditions using limited support trajectories. A centralized actor–critic optimization scheme is adopted to generate coordinated velocity commands based on the states of both manipulators, target information, and obstacle conditions. The proposed framework

is evaluated on held-out ID and OOD task manifests using 3-DOF and 6-DOF dual-arm systems. Controlled comparisons investigate initialization strategies, meta-gradient formulations, and alternative learning paradigms, while the adapted 6-DOF policy is further validated on a physical dual-UR5 platform with RGB-D perception and closed-loop execution. The contributions of this work are summarized as follows:

1. A centralized task-adaptation framework (DA-MRL) is formulated for coordinated dual-arm dynamic obstacle avoidance. The framework learns task-generalized policy initializations from dynamic task distributions and enables adaptation to unseen conditions with limited support trajectories.
2. The proposed framework is validated across robotic configurations ranging from reduced-order 3-DOF systems to full 6-DOF dual-UR5 platforms, demonstrating consistent performance under different kinematic complexities and task variations.
3. A sim-to-real evaluation is conducted on a physical dual-UR5 platform using the simulation-adapted policy with fixed parameters during execution. RGB-D perception and closed-loop control are integrated to evaluate target completion, obstacle clearance, and collision-related execution outcomes.

2 Related work

2.1 Robot collision avoidance and dual-arm reinforcement learning

Geometric planning, optimization, and reactive control maintain manipulator clearance through explicit robot and obstacle models [9, 13, 15]. In physical human–robot interaction, collision detection and reaction provide an additional response layer [4, 6]. A dual-arm system introduces interdependent feasible regions because the motion of one manipulator changes the space and timing available to the other. Learning-based formulations represent this coupling within the state and optimize the two arm commands jointly.

RL studies have represented coordinated manipulation through both single-policy and multi-agent formulations. Liu et al. [12] coordinate two arms through collaborative DDPG, whereas other work assigns joint groups or task-space variables to separate agents [2, 16, 24]. TDQN provides a team-control formulation for a surgical continuum manipulator [7]. DA-MRL uses a centralized policy organization: one observation contains both arm states, object information, and safety-related distances, and one policy produces the complete joint command. During training, centralized dual-arm information is used for policy optimization, while the learned policy generates coordinated joint commands during execution.

Sim-to-real transfer adds discrepancies in perception and execution to those already present in the simulated task. Rusu et al. [19] transferred a pixel-driven reinforcement-learning policy to a physical manipulator, and Lin et al. [11] studied bimanual transfer with tactile feedback. These examples make the sensing interface and the boundary between learned and conventional control important parts of the experimental description. In the present deployment, target and obstacle states estimated by the camera are combined with centralized dual-UR5 proprioception, and the adapted policy remains fixed during hardware execution.

2.2 Reinforcement learning-based methods

Reinforcement learning methods for robot control differ principally in their action representation, policy-update regime, and treatment of multiple decision variables. The comparison methods considered here span discrete value-based learning, continuous off-policy actor–critic learning, and continuous on-policy policy optimization. This range is relevant to dual-arm avoidance because the joint command must coordinate both manipulators while responding to changes in obstacle motion and task geometry.

Deep Q-Networks (DQN) approximate an action-value function with a neural network and select actions from a discrete action set [14]. A high-dimensional joint-velocity controller requires discretization of each action component, causing the associated Cartesian action set to grow rapidly with the number of joints. Team Deep Q-Learning (TDQN) extends value-based learning to team settings by incorporating team information into the learning design; Ji et al. [7] demonstrate this idea for surgical-continuum-manipulator control.

In the comparison, DQN and TDQN represent discrete-action value-based methods, whereas DA-MRL uses a continuous joint-velocity policy.

Deep Deterministic Policy Gradient (DDPG) combines a deterministic continuous-action actor with a critic [10]. Its continuous action interface is well suited to joint-velocity control, although its off-policy updates and replay distribution differ fundamentally from the on-policy learner used here. Dual-arm DDPG (DADDPG) is the collaborative dual-arm formulation proposed by Liu et al. [12]. DDPG and DADDPG provide continuous-action, off-policy references that complement the discrete value-based DQN and TDQN baselines.

PPO is an on-policy policy-optimization method based on a clipped surrogate objective [23]. In this study, it is instantiated as a continuous-action actor–critic learner. The clipping operation regulates excessively large policy-ratio changes during a local update. In DA-MRL, PPO supplies the base learner for both inner task adaptation and outer meta-optimization. Unlike DDPG and DADDPG, PPO does not reuse a replay buffer of off-policy transitions: support and query trajectories are collected under the corresponding current policies. This property aligns the base learner with the support–query adaptation protocol used in the present study.

2.3 Meta-reinforcement learning for rapid adaptation

Meta-reinforcement learning considers policies that must be specialized with a prescribed amount of interaction with a new task. MAML optimizes an initialization from which a small number of gradient updates adapts a model to a sampled task [3]. For dynamic obstacle avoidance, tasks may differ in target location, obstacle trajectory, effective dynamics, observation noise, or action delay. Meta-training organizes experience across these variations so that task-specific updates begin from a common initialization.

Existing Meta-RL methods follow several adaptation mechanisms. Proximal policy search methods optimize policies for rapid adaptation through gradient updates [18], whereas PEARL infers task context from experience and conditions the policy on that context [17]. Meta-World provides a multi-task manipulation benchmark for examining transfer across related robotic tasks [26]. DA-MRL adopts gradient-based parameter adaptation because the target task is represented directly by new support trajectories collected under the current dual-arm policy.

Gradient-based Meta-RL requires the data used for task-specific adaptation to be distinguished from the post-adaptation data used by the meta-objective. This distinction is particularly relevant to an on-policy learner because an adapted policy induces a different trajectory distribution. DA-MRL adopts this structure for centralized dual-arm control, and the experiments compare first-order and full meta-gradients under a common task sampler and interaction budget.

3 DA-MRL: dual-arm meta-reinforcement learning

Figure 2 summarizes the information flow of DA-MRL. A centralized actor–critic starts from a shared initialization and receives a sampled dual-arm avoidance task. Support trajectories adapt the actor and critic to that task; fresh query trajectories determine the outer update of the shared initialization. After meta-training, adaptation to an evaluation task proceeds without further updates to the shared initialization.

3.1 Dynamic dual-arm task distribution

Following the task definition illustrated in Figure 2, dynamic dual-arm obstacle avoidance is formulated as an adaptation problem over a distribution of related tasks rather than a single fixed environment. Each task describes a specific interaction condition between the dual manipulators and the surrounding workspace, including target configurations, obstacle motion patterns, and task-dependent environmental variations.

The task distribution is defined as

$$\tau_i \sim p(\tau), \tag{1}$$

where τ_i denotes the i-th task sampled from the task distribution $p(\tau)$. A task instance can be represented as

$$\tau_i = (x_T^L, x_T^R, r_{\text{obs}}, r_{\text{orbit}}, \omega_{\text{obs}}, \eta_{\text{noise}}, d_{\text{delay}}, \xi_{\text{dyn}}), \tag{2}$$

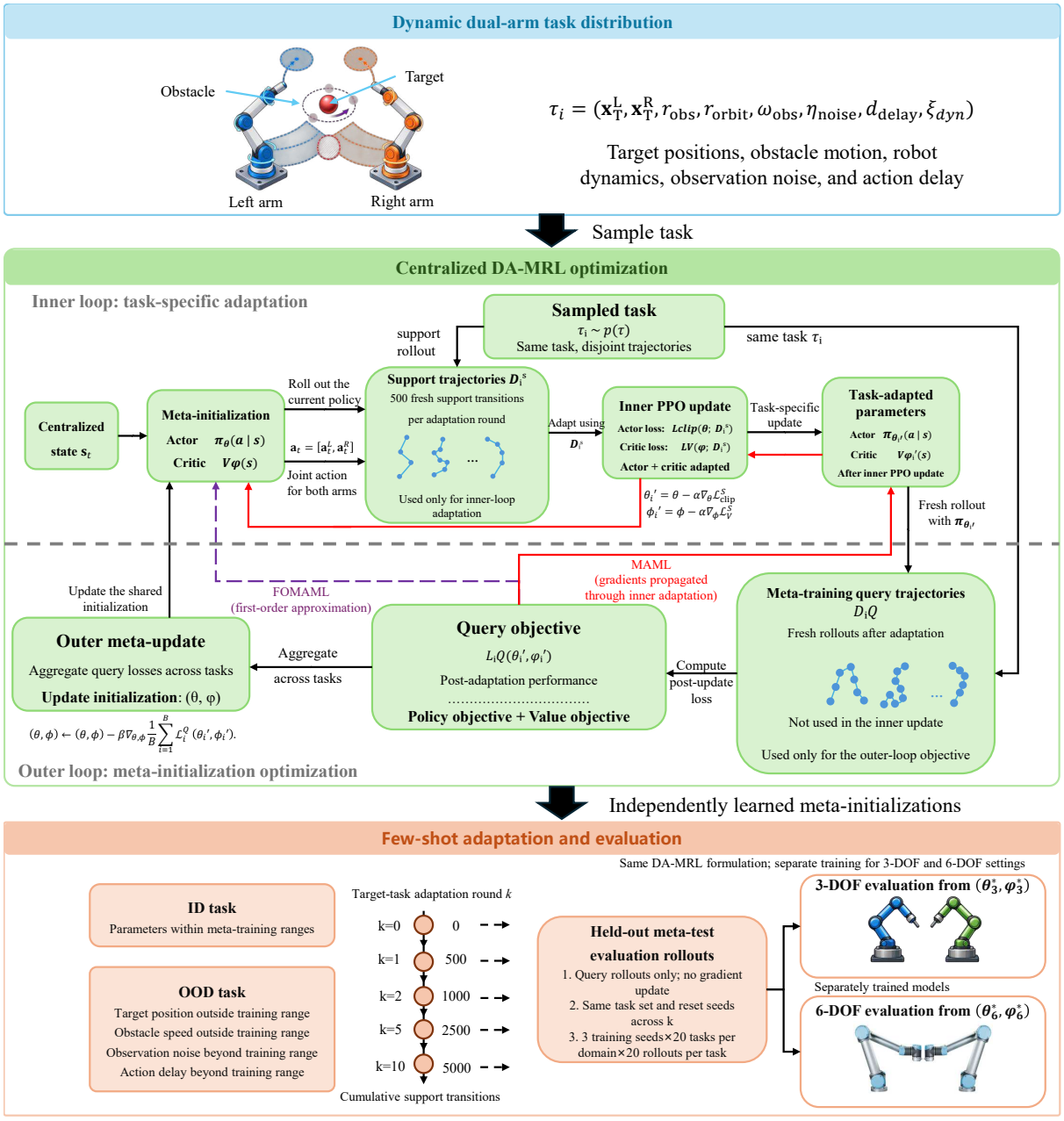


Fig. 2 Overall framework of DA-MRL. A dynamic dual-arm task specifies target positions, obstacle motion, dynamics variation, observation noise, and action delay. For each sampled task, support trajectories are used to adapt the actor and critic through inner updates, whereas fresh query trajectories from the same task define the outer meta-objective. FOMAML approximates the meta-gradient, while MAML differentiates through the inner adaptation. The learned 3-DOF and 6-DOF initializations are evaluated separately on held-out ID and OOD task manifests after 0, 1, 2, 5, and 10 adaptation updates. Each manifest contains 20 evaluation tasks, and each task is evaluated over 20 independent query rollouts per training seed.

where x_T^L and x_T^R denote the target positions of the left and right manipulators, respectively; r_{obs} is the obstacle radius; r_{orbit} and ω_{obs} describe the obstacle motion; η_{noise} represents observation noise; d_{delay} denotes action delay; and ξ_{dyn} represents dynamics variations. In the 3-DOF formulation, both arms approach the common sampled target, so $x_T^L = x_T^R = p_T$. The 6-DOF formulation

assigns an individual contour subtarget to each arm.

Each sample from $p(\tau)$ therefore defines a complete dynamic obstacle-avoidance condition rather than an isolated geometric perturbation. Variation in geometry, obstacle motion, sensing, delay, and dynamics is incorporated into the same task representation so that adaptation is driven by the trajectories observed under the current condition. DA-MRL uses these related tasks to learn an initialization that can be specialized with a prescribed amount of target-task interaction.

The meta-learning objective is formulated as

$$(\theta^*, \phi^*) = \arg\max_{\theta,\phi} \mathbb{E}_{\tau_i \sim p(\tau)} \left[J_{\tau_i}(\theta_i', \phi_i') \right], \tag{3}$$

where (θ^*, ϕ^*) denotes the learned actor–critic initialization and (θ_i', ϕ_i') denotes its task-adapted parameters. The objective evaluates the return after adaptation rather than the return of the unadapted policy alone. Consequently, meta-training rewards an initialization whose actor and critic can respond productively to new support trajectories while retaining the centralized coordination required by the two manipulators.

3.2 Centralized DA-MRL optimization

The proposed centralized DA-MRL framework performs task adaptation through centralized policy optimization over the sampled dual-arm tasks. Following gradient-based meta-learning formulations, MAML and FOMAML are considered as optimization variants for updating the shared actor–critic initialization. During training, information from both manipulators and the shared workspace is jointly considered to optimize a task-generalized initialization. The adapted policy subsequently generates coordinated velocity commands for dual-arm execution.

3.2.1 Centralized dual-arm policy

The centralized policy uses configuration-specific observations that preserve the kinematic and geometric information required by each system. For the 3-DOF system, arm i contributes

$$\begin{aligned} o_{t,i}^{\mathrm{3D}} = \big[& q_{t,i}, \dot{q}_{t,i}, p_{e,t}^{j} - p_{e,t}^{i}, \\ & p_{T,t} - p_{e,t}^{i}, p_{O,t} - p_{e,t}^{i} \big] \in \mathbb{R}^{15}, \end{aligned} \tag{4}$$

where $q_{t,i} \in \mathbb{R}^3$ and $\dot{q}_{t,i} \in \mathbb{R}^3$ are the joint positions and velocities, $p_{e,t}^{i}$ is the end-effector position of arm i, $p_{e,t}^{j}$ is the peer end-effector position with $j \neq i$, and $p_{T,t}$ and $p_{O,t}$ are the target and obstacle positions. The fixed-order concatenation $s_t^{\mathrm{3D}} = [o_{t,R}^{\mathrm{3D}}, o_{t,L}^{\mathrm{3D}}] \in \mathbb{R}^{30}$ forms the centralized state, and the corresponding action is $a_t^{\mathrm{3D}} \in [-1, 1]^6$.

For the full 6-DOF system, arm i contributes a 24-dimensional observation block:

$$\begin{aligned} o_{t,i}^{\mathrm{6D}} = \big[& q_{t,i}, \dot{q}_{t,i}, p_{T,t} - p_{e,t}^{i}, p_{O,t} - p_{e,t}^{i}, \\ & p_{TS,t}^{i} - p_{e,t}^{i}, p_{OS,t}^{i} - p_{e,t}^{i} \big] \in \mathbb{R}^{24}. \end{aligned} \tag{5}$$

Here $p_{TS,t}^{i}$ and $p_{OS,t}^{i}$ are the closest points on the target and obstacle support structures to end effector i. The fixed-order concatenation $s_t^{\mathrm{6D}} = [o_{t,R}^{\mathrm{6D}}, o_{t,L}^{\mathrm{6D}}] \in \mathbb{R}^{48}$ provides the centralized observation used by both actor branches and the critic.

Based on the centralized observation, the actor network generates the coordinated action:

$$a_t = \pi_\theta(s_t), \tag{6}$$

where $a_t = [a_t^L, a_t^R]$ represents the joint velocity command of both manipulators and θ denotes the actor parameters.

The control objective is to guide both end effectors to their assigned targets while maintaining collision-free execution with respect to moving obstacles and inter-arm interference. Successful task completion therefore requires bilateral target achievement within the episode horizon without any collision during execution.

The critic network estimates the expected return of the current state $V_\phi(s_t)$, where ϕ denotes the critic parameters. The actor and critic are jointly optimized to improve target-reaching performance while maintaining collision-free coordination.

The actor is optimized by minimizing the negative clipped PPO surrogate objective:

$$L_{\text{clip}}(\theta) = -\mathbb{E}_t\Big[\min\Big(r_t(\theta)A_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)A_t\Big)\Big], \tag{7}$$

where $r_t(\theta)$ denotes the probability ratio between the updated and previous policies, A_t represents the estimated advantage, and ϵ is the clipping coefficient.

The critic is optimized by minimizing the value regression loss:

$$L_V(\phi) = \mathbb{E}_t\left[(V_\phi(s_t) - R_t)^2\right], \tag{8}$$

where R_t represents the target return used for value estimation.

The actor and critic objectives are jointly optimized during meta-training and task-specific adaptation.

The centralized formulation couples the two arms at both decision and evaluation time. Each action is conditioned on the complete dual-arm state, while the critic assigns value to the joint configuration and shared workspace. This coupling allows an avoidance maneuver by one arm to be evaluated together with the target progress and clearance of the other arm, rather than as two independent single-arm decisions.

3.2.2 Task-specific adaptation with support trajectories

For each sampled task τ_i, a support dataset is collected to perform task-specific adaptation:

$$D_i^S = \{(s_t, a_t, r_t)\}_{t=0}^{H_S} \tag{9}$$

The support set contains the limited interaction available for specializing the shared initialization to task τ_i. Returns and generalized advantage estimates are computed from these trajectories and treated as fixed targets within each surrogate update. Both the actor and critic are adapted, so the task-specific policy update is accompanied by a value function fitted to the same target condition.

The inner-loop actor and critic updates at step j are formulated as:

$$\theta_i^{j+1} = \theta_i^j - \alpha_\theta \nabla_{\theta_i^j} L_{\text{clip}}(\theta_i^j; D_i^S), \tag{10}$$

$$\phi_i^{j+1} = \phi_i^j - \alpha_\phi \nabla_{\phi_i^j} L_V(\phi_i^j; D_i^S), \tag{11}$$

where j denotes the inner-loop update index, and $(\theta_i^0, \phi_i^0) = (\theta, \phi)$ represents the current shared actor–critic initialization for task τ_i. The task-adapted parameters are obtained as $(\theta_i', \phi_i') = (\theta_i^{N_{\text{inner}}}, \phi_i^{N_{\text{inner}}})$ after N_{inner} updates, where α_θ and α_ϕ denote the actor and critic inner-loop learning rates, respectively.

Repeated inner updates apply Eqs. (10) and (11) to the current task while preserving the common starting parameters for the other tasks in the meta-batch. This separation yields one adapted actor–critic pair per sampled task. The resulting parameters encode the task-specific response inferred from its support trajectories and are not copied between tasks.

3.2.3 Meta-optimization with query trajectories

After task-specific adaptation, independent query trajectories are collected using the adapted policy:

$$D_i^Q = \{(s_t, a_t, r_t)\}_{t=0}^{H_Q}. \tag{12}$$

The query trajectories define the outer objective evaluated at the adapted actor–critic parameters. They are sampled after the inner update and remain separate from the support data, so the meta-objective measures how the adapted policy behaves on fresh experience from the same task:

$$\mathcal{L}_i^Q = L_\pi^Q(\theta_i'; D_i^Q) + \lambda L_V^Q(\phi_i'; D_i^Q), \tag{13}$$

Minimizing this loss updates the shared initialization according to post-adaptation actor and critic performance. The support set therefore determines the task-specific update, whereas the query set determines whether that update produces a policy that generalizes within the sampled task.

The outer-loop meta-update is formulated as:

$$(\theta, \phi) \leftarrow (\theta, \phi) - \beta \nabla_{\theta,\phi} \frac{1}{B} \sum_{i=1}^{B} \mathcal{L}_i^Q(\theta_i', \phi_i'). \tag{14}$$

where β denotes the meta-update learning rate, B represents the number of sampled tasks in a meta-batch, and $\mathcal{L}_i^Q(\theta_i', \phi_i')$ represents the query loss evaluated using the adapted actor–critic parameters.

Algorithm 1 DA-MRL meta-training with support–query optimization

Require: Shared parameters (θ, ϕ); task distribution $p(\tau)$; batch size B; inner steps N_{inner}; rates $(\alpha_\pi, \alpha_V, \beta)$; value-loss weight λ; gradient mode $g \in \{\text{MAML}, \text{FOMAML}\}$

Ensure: Shared initialization (θ, ϕ) for validation

for each meta-iteration **do**
  Sample $\mathcal{B} = \{\tau_i\}_{i=1}^{B}$ from $p(\tau)$
  for each $\tau_i \in \mathcal{B}$ **do**
    Initialize $(\theta_i^0, \phi_i^0) \leftarrow (\theta, \phi)$
    Collect support trajectories $\mathcal{D}_i^S$ with $\pi_{\theta_i^0}$
    Compute fixed support returns $\hat{R}_i^S$ and GAE advantages $\hat{A}_i^S$
    for $j = 0$ to $N_{\text{inner}} - 1$ **do**
      Evaluate the support actor and critic losses
      Update $(\theta_i^{j+1}, \phi_i^{j+1})$ using Eqs. (10)–(11)
    end for
    Set $(\theta_i', \phi_i') \leftarrow (\theta_i^{N_{\text{inner}}}, \phi_i^{N_{\text{inner}}})$
    Collect fresh query trajectories $\mathcal{D}_i^Q$ with $\pi_{\theta_i'}$
    Compute query losses $\mathcal{L}_{\pi,i}^Q$ and $\mathcal{L}_{V,i}^Q$
  end for
  Form $\mathcal{L}_{\text{meta}}$ using the query objectives.
  if $g = \text{MAML}$ **then**
    Differentiate $\mathcal{L}_{\text{meta}}$ through the inner updates
  else
    Omit second-order terms in the meta-gradient
  end if
  Update $(\theta, \phi) \leftarrow (\theta, \phi) - \beta \nabla_{\theta,\phi} \mathcal{L}_{\text{meta}}$
end for

MAML differentiates the query loss through the inner actor and critic updates, retaining the dependence of (θ_i', ϕ_i') on the shared initialization. FOMAML uses the same support and query trajectories but omits the corresponding higher-order terms when constructing the outer gradient. The controlled comparisons in Sections 4.4 and 4.5 therefore isolate the meta-gradient calculation while holding the policy architecture, task manifests, adaptation budget, and evaluation metrics fixed.

3.3 Few-shot adaptation and evaluation

After meta-training, support trajectories from a target task τ_{tar} adapt the learned initialization:

$$(\theta_{\text{tar}}', \phi_{\text{tar}}') = \mathit{Adapt}((\theta^*, \phi^*), D_{\text{tar}}^S), \tag{15}$$

where (θ^*, ϕ^*) denotes the learned actor–critic initialization, D_{tar}^S represents the support trajectories collected from the target task, and $(\theta_{\text{tar}}', \phi_{\text{tar}}')$ denotes the adapted actor–critic parameters.

Independent query trajectories evaluate the adapted policy without contributing further parameter updates. Performance is measured at $k = \{0, 1, 2, 5, 10\}$, where k is the number of task-specific updates after initialization. Comparing these checkpoints exposes both the zero-update quality of the learned initialization and the rate at which task performance changes as additional support data become available.

ID tasks follow the meta-training distribution, whereas OOD tasks extend its obstacle motion, target, sensing, delay, and dynamics ranges. Both domains use held-out task manifests during evaluation. The same task definitions and query-reset seeds are retained across adaptation checkpoints, so changes from $k = 0$ to $k = 10$ reflect the adapted parameters under a paired evaluation protocol.

To investigate the applicability of DA-MRL across different robotic configurations, the adaptation procedure is evaluated on reduced-order 3-DOF dual-arm systems and full 6-DOF dual-UR5 platforms. Because their observation and action spaces differ, each system is optimized using its corresponding task formulation and policy initialization. The comparison examines whether the same support–query learning procedure produces useful adaptation in both kinematic settings; it does not transfer a common parameter vector between networks of different dimensions.

For physical deployment, the simulation-adapted 6-DOF policy remains fixed while RGB-D measurements construct its observation and the resulting velocity commands drive the dual-UR5 platform. This final stage evaluates the closed-loop behavior of the selected simulation checkpoint

Algorithm 2 Adaptation and evaluation of DA-MRL

Require: Selected (θ^*, ϕ^*); target task τ_{tar}; support budget M; maximum updates K; checkpoints $\mathcal{C}$; query count N_Q; test learning rates $(\alpha_\pi^{\text{test}}, \alpha_V^{\text{test}})$

Ensure: Adapted parameters (θ'_K, ϕ'_K) and query evaluations at checkpoints $\mathcal{C}$

Initialize $(\theta'_0, \phi'_0) \leftarrow (\theta^*, \phi^*)$
Collect N_Q query rollouts with $\pi_{\theta'_0}$ and report checkpoint $k = 0$
for $k = 1$ to K **do**
    Collect M fresh support transitions $\mathcal{D}_k^S$ on τ_{tar} with $\pi_{\theta'_{k-1}}$
    Reset the same task condition as needed until all M transitions are obtained
    Compute fixed support returns $\hat{R}_k^S$ and GAE advantages $\hat{A}_k^S$
    Evaluate the support actor loss L_π^S and critic loss L_V^S
    Update actor and critic parameters to (θ'_k, ϕ'_k) using $(\alpha_\pi^{\text{test}}, \alpha_V^{\text{test}})$
    if $k \in \mathcal{C}$ **then**
        Collect N_Q fresh query rollouts with $\pi_{\theta'_k}$ and report the prescribed metrics
    end if
end for

through the same centralized observation and joint-action interface used by the 6-DOF model.

The complete adaptation and evaluation procedure is summarized in Algorithm 2.

4 Experiments

The experiments progress from 3-DOF baseline behavior and controlled adaptation to full 6-DOF control and physical deployment. Standard RL and Meta-RL baselines first characterize whether conventional training and direct fine-tuning yield coordinated collision-free completion. Non-meta-learned initialization controls then separate prolonged target-task optimization from meta-trained adaptation. The FOMAML/MAML study evaluates the effect of the meta-gradient under matched data budgets, after which the same adaptation procedure is examined with full 6-DOF control and a frozen-policy physical deployment.

4.1 Experimental protocol and metrics

All simulation studies are conducted in MuJoCo with centralized dual-arm observations and joint actions. Sections 4.3–4.5 report controlled comparisons using three training seeds. Table 1 summarizes the 3-DOF configuration; Section 4.5 specifies the full 6-DOF system.

The 3-DOF meta-training distribution and held-out ID manifest use target coordinates $x \in [0.600, 0.800]$ m, $y \in [-0.100, 0.100]$ m, and $z \in [0.400, 0.600]$ m. The obstacle follows a circular orbit with radius $r_{\text{orbit}} \in [0.150, 0.250]$ m, angular velocity $\omega_{\text{obs}} \in [1.000, 2.000]$ rad/s, phase in $[0, 2\pi]$, and physical radius $r_{\text{obs}} = 0.100$ m. These tasks use unit dynamics scaling, zero observation noise, zero action delay, and an all-zero initial joint configuration. The held-out OOD manifest expands the target ranges to $x \in [0.500, 0.900]$ m, $y \in [-0.250, 0.250]$ m, and $z \in [0.300, 0.700]$ m. Its obstacle orbit radius spans $[0.260, 0.350]$ m, angular velocity spans $[2.100, 3.000]$ rad/s, and physical radius spans $[0.050, 0.080]$ m. OOD tasks also sample dynamics scaling from $[0.750, 1.250]$, observation-noise standard deviation from $[0.000, 0.030]$, and action delay from zero to two control steps while retaining the same all-zero initial joint configuration.

The team reward is designed to balance target-reaching progress, action regularization, obstacle avoidance, and inter-arm separation. For the common reward terms, the per-step reward is defined as

$$r_t = \frac{1}{N} \sum_{i=1}^{N} \Big[-c_1 L_\delta(d_t^i) - c_4 \|a_t^i\|_2^2 - c_3 \psi(d_O^i) - w_c \sum_{j \neq i} \psi(d_{ij}) \Big], \tag{16}$$

where $N = 2$ is the number of manipulators, d_t^i denotes the distance between arm i and its assigned target, d_O^i represents the clearance between arm i and dynamic obstacles, and d_{ij} denotes the inter-arm distance. $L_\delta(\cdot)$ is the Huber loss and $\psi(\cdot)$ is the distance-based penalty function. The coefficients c_1 (denoted as w_g in Table 1), c_4 (w_a), c_3 (w_o), and w_c weight target-reaching progress, action penalty, obstacle avoidance, and inter-arm separation, respectively. The collaboration bonus w_{collab} and global optimal

bonus R_{global} listed in Table 1 are additional sparse bonuses applied in the 3-DOF baseline reward.

An episode terminates when a collision occurs, the maximum episode horizon ($T_{\max}$ in 3-DOF or the defined control steps in 6-DOF) is reached, or the task is completed. Collision detection and safety termination are evaluated independently of the shaped reward. In the simulation evaluation, collision events include detected physical contact or geometric overlap between the two manipulators, or between either manipulator and the moving obstacle, whereas positive clearances used for safety margin monitoring are not counted as collisions.

The adaptation protocol follows Algorithm 2. Each update collects $M = 500$ target-task transitions, resetting the same task as needed to complete the support batch. Checkpoint $k = 0$ evaluates the initialization; checkpoint k follows $500k$ cumulative support transitions. The controlled 3-DOF FOMAML/MAML and 6-DOF studies use fixed ID and OOD manifests of 20 tasks per domain. For seeds 42, 43, and 44, every task receives $N_Q = 20$ query rollouts per checkpoint, giving 400 rollouts per seed and 1,200 in total. Metrics are averaged over rollouts, tasks, and then seeds, with task definitions and reset seeds fixed across checkpoints. Section 4.3 specifies the fixed-task protocol used by the non-meta-learned controls.

An episode is considered successful when both arms reach their assigned targets within the episode horizon without contact with the dynamic obstacle or the other arm. The collision indicator records inter-arm contact and contact between either arm and the moving obstacle, whereas same-arm non-adjacent contact, table contact, and ground contact are excluded from the formal collision criterion. The reported metrics include collision-free success rate (SR), collision rate (CR), and mean worst final end-effector distance (MWFD). Let $y^{\text{succ}}_{s,i,r}$ and $y^{\text{coll}}_{s,i,r}$ denote the binary success and collision indicators for rollout r of evaluation task i under training seed s. For the two final end-effector distances $d^{R,\text{final}}_{s,i,r}$ and $d^{L,\text{final}}_{s,i,r}$, the worst final distance is defined as

$$d^{\text{worst}}_{s,i,r} = \max\left(d^{R,\text{final}}_{s,i,r}, d^{L,\text{final}}_{s,i,r}\right) \tag{17}$$

Table 1 Common nominal parameters of the training environment, network, and optimization for the 3-DOF actor–critic configurations. Sampled task ranges are specified in the text.

Parameter	Symbol	Value
Environment and network		
Number of arms	N_{arms}	2
Obs. dim./arm	d_{obs}	15
Global obs. dim.	d_s	30
Action dim./arm	d_{act}	3
Nominal target center	p_T^{nom}	$[0.700, 0.000, 0.500]$
Base-position radius	r_{base}	0.500 m
Arm lengths	L	$[0.600, 0.500, 0.400]$ m
Time step	Δt	0.030 s
Maximum episode steps	$T_{\max}$	400
Nominal orbit radius	$r_{\text{orbit}}^{\text{nom}}$	0.200 m
Physical obstacle radius	r_{obs}	0.100 m
Nominal angular velocity	$\omega_{\text{obs}}^{\text{nom}}$	1.500 rad/s
Success threshold	d_{success}	0.100 m
Collision threshold	$d_{\text{collision}}$	0.050 m
Hidden-layer size	d_h	128
Reward coefficients		
Target reward weight	w_g (c_1)	1000
Obstacle reward weight	w_o (c_3)	60
Action penalty weight	w_a (c_4)	10
Inter-arm term	$w_c\psi(d_{ij})$	piecewise
Obstacle potential exponent	p	8
Collision potential exponent	q	8
Collaboration bonus	w_{collab}	100
Global optimal bonus	R_{global}	500
Training hyperparameters		
Meta-training episodes	N_{meta}	5000
Tasks per meta step	B_{meta}	8
Support-set steps	K_{steps}	500
Discount factor	γ	0.99
Meta learning rate	α_{meta}	1×10^{-4}
Inner learning rate	α_{inner}	1×10^{-3}
Inner updates (actor/critic)	N_{inner}	3/3
Test adaptation rate	α_{test}	1×10^{-3}
Fine-tuning rate	α_{ft}	1×10^{-3}

$$\text{SR}_s = \frac{1}{N_{\text{task}}R}\sum_{i=1}^{N_{\text{task}}}\sum_{r=1}^{R} y^{\text{succ}}_{s,i,r} \tag{18}$$

$$\text{CR}_s = \frac{1}{N_{\text{task}}R}\sum_{i=1}^{N_{\text{task}}}\sum_{r=1}^{R} y^{\text{coll}}_{s,i,r} \tag{19}$$

$$\text{MWFD}_s = \frac{1}{N_{\text{task}}R}\sum_{i=1}^{N_{\text{task}}}\sum_{r=1}^{R} d^{\text{worst}}_{s,i,r} \tag{20}$$

where $N_{\text{task}} = 20$ and $R = 20$. For each seed-level metric $m_s \in \{\text{SR}_s, \text{CR}_s, \text{MWFD}_s\}$, the

reported result is obtained by averaging over the three training seeds:

$$\bar{m} = \frac{1}{S}\sum_{s=1}^{S} m_s, \qquad S = 3. \tag{21}$$

Thus, MWFD first selects the larger final distance between the two arms in each rollout and then follows the same rollout–task–seed aggregation hierarchy as the success and collision metrics. A lower MWFD indicates that both manipulators terminate closer to their respective target regions.

4.2 3-DOF algorithmic results

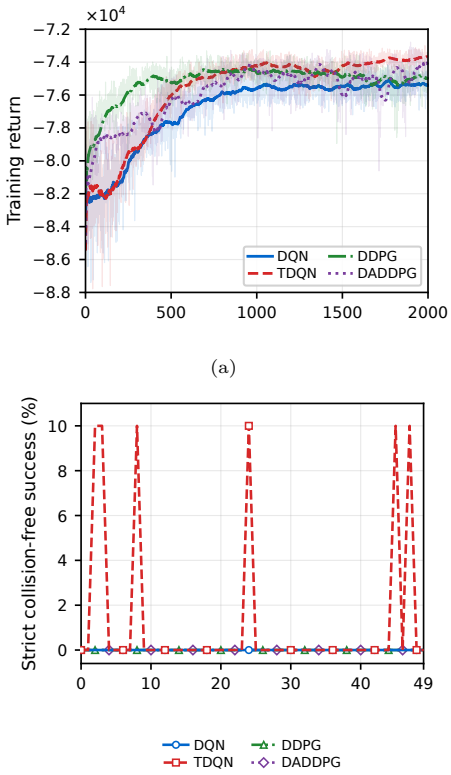
4.2.1 Standard RL baseline behavior

The four reinforcement-learning baselines use their original 30-dimensional observations, comprising joint states, absolute end-effector positions, and relative target and obstacle positions; DA-MRL adds the relative inter-arm representation in Eq. (4). Figures 3 and 4 report their training behavior and representative trajectories in the 3-DOF environment.

Training returns increase, but strict bilateral success remains close to zero for most methods during evaluation. The return traces indicate that each learner improves its shaped objective, whereas the success curves show that this improvement does not consistently satisfy the joint requirement of collision avoidance and bilateral target completion. The trajectory plots clarify the resulting behaviors: some policies approach the target region but intersect the obstacle path, while others maintain separation at the cost of incomplete target progress. These results establish the difficulty of converting reward improvement into coordinated task completion under the stated protocol.

4.2.2 Meta-RL baselines

Meta-DQN, Meta-TDQN, Meta-DDPG, and Meta-DADDPG start from the corresponding standard RL weights. Figure 5 reports the seed-42 runs over approximately 8×10^5 environment steps; each marker averages a non-overlapping 50,000-step interval. Returns improve during optimization, whereas collision-free success remains intermittent. The separation between these two signals again distinguishes progress under the shaped training objective from strict episode-level completion. The endpoint comparison with the formal three-seed 6-DOF results is reported in Panel B of Table 4, where success, collision rate, and final target error are evaluated together.



(a)

(b)

Fig. 3 Standard RL behavior in the 3-DOF environment. (a) Training returns for individual episodes (light traces) and their 50-episode moving averages (main curves). (b) Strict bilateral collision-free success during adaptation evaluation. Each point represents the average success rate over ten evaluation runs at the corresponding adaptation episode.

4.3 Non-meta-learned initialization controls on fixed 3-DOF tasks

All three controls use the same Proximal Policy Optimization (PPO) actor–critic learning rule [23]. The evaluated conditions are random initialization (PPO-scratch) on one fixed ID task, pretraining followed by adaptation on the same fixed ID task, and the pretrained initialization adapted on one fixed OOD task. The ID task places the target at $(0.750, -0.050, 0.550)$ m and

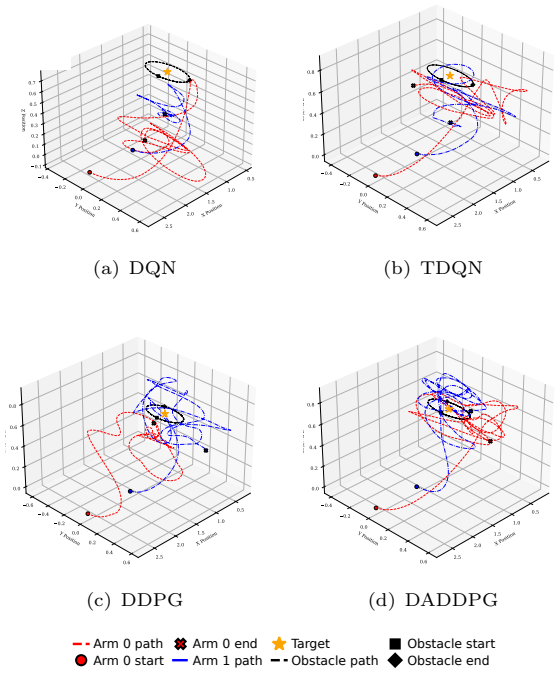


Fig. 4 Representative 3-DOF dual-arm trajectories generated by DQN, TDQN, DDPG, and DADDPG. The shared legend indicates the trajectories of both manipulators, their initial and target positions, and the moving obstacle path.

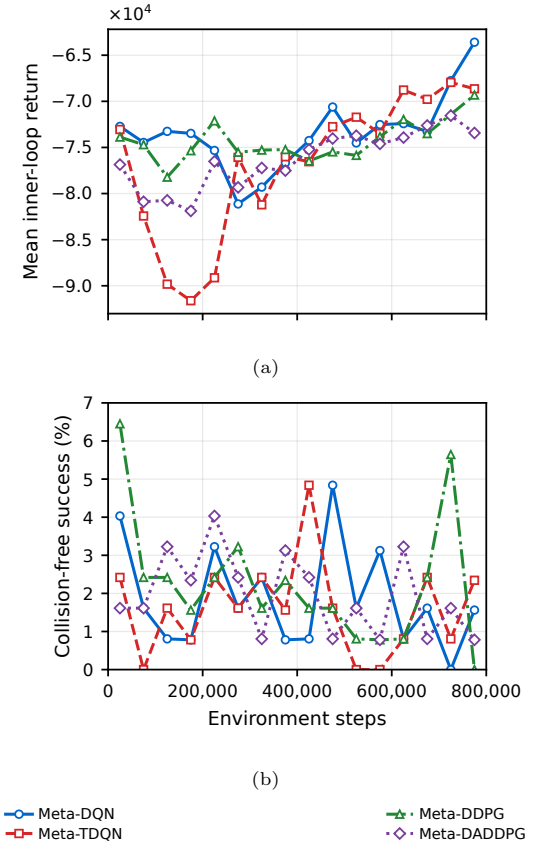


Fig. 5 Training behavior of Meta-DQN, Meta-TDQN, Meta-DDPG, and Meta-DADDPG after initialization from their corresponding RL weights: (a) mean inner-loop return and (b) collision-free success rate. Each marker is the mean of the updates within a non-overlapping 50,000-step interval.

uses an obstacle orbit radius of 0.220 m, physical obstacle radius of 0.100 m, angular velocity of 1.200 rad/s, unit dynamics scaling, zero observation noise, and zero action delay. The OOD task places the target at $(0.773, 0.162, 0.586)$ m and uses an orbit radius of 0.330 m, physical obstacle radius of 0.068 m, angular velocity of 2.590 rad/s, dynamics scaling of 0.773, observation-noise standard deviation of 0.024, and an action delay of one control step. Both tasks start from the all-zero joint configuration.

The checkpoints at 0, 1, 5, 10, 25, and 50 updates correspond to cumulative support budgets of 0, 500, 2,500, 5,000, 12,500, and 25,000 transitions per seed. At every checkpoint, each of the policies trained with seeds 42, 43, and 44 is evaluated over the same 20 reset seeds on its assigned fixed task. Each reported point therefore summarizes 20 episodes per training seed and 60 episodes across the three seeds. Metrics are calculated within each seed before the three seed-level values are averaged.

Table 2 reports 0% collision-free success and 100% collision for all three controls. From zero to fifty updates, pretraining reduces final distance from 0.496 to 0.474 m in ID and from 0.561 to 0.537 m in OOD, whereas PPO-scratch remains near 0.57 m. Thus, pretraining changes the terminal target error but does not produce a rollout that simultaneously satisfies bilateral arrival and the collision criterion. Extending adaptation from 5,000 to 25,000 target-task transitions does not change the success or collision outcome. The subsequent DA-MRL comparison tests whether an initialization optimized explicitly for post-adaptation query performance produces a different adaptation profile under the same task objective.

Table 2 Non-meta-learned PPO initialization controls on fixed 3-DOF tasks. Values are three-seed means from 20 episodes per seed and checkpoint; each update consumes 500 support transitions. Arrows indicate the preferred direction.

Initialization	Domain	Updates	Success rate (%) ↑	Collision rate (%) ↓	Final distance (m) ↓
PPO-scratch	ID	0	0.0	100.0	0.574
		1	0.0	100.0	0.575
		5	0.0	100.0	0.571
		10	0.0	100.0	0.570
		25	0.0	100.0	0.567
		50	0.0	100.0	0.569
PPO-pretrain	ID	0	0.0	100.0	0.496
		1	0.0	100.0	0.496
		5	0.0	100.0	0.493
		10	0.0	100.0	0.481
		25	0.0	100.0	0.470
		50	0.0	100.0	0.474
	OOD	0	0.0	100.0	0.561
		1	0.0	100.0	0.557
		5	0.0	100.0	0.549
		10	0.0	100.0	0.542
		25	0.0	100.0	0.538
		50	0.0	100.0	0.537

4.4 DA-MRL adaptation results under ID and OOD task shifts

The 3-DOF comparison isolates the meta-gradient calculation within the protocol of Section 4.1. FOMAML approximates the outer gradient, whereas MAML differentiates through the actor and critic updates; the remaining training and evaluation conditions are matched.

4.4.1 Adaptation dynamics across update budgets

Figure 6 traces collision-free success, collision rate, and MWFD over 0, 1, 2, 5, and 10 updates. In ID, MAML develops a stronger success increase and larger collision and distance reductions than FOMAML. In OOD, MAML overtakes the initially higher FOMAML success after the early non-monotonic updates and improves through five and ten updates.

4.4.2 Quantitative comparison under ID and OOD shifts

Table 3 shows that MAML reaches 89.9% ID success after ten updates, 10.0 points above FOMAML, with lower collision rate and MWFD. In OOD, MAML moves from a 4.0-point initial deficit to a 14.6-point advantage, gaining 32.4 success points, reducing collision by 31.5 points, and lowering MWFD from 0.342 to 0.136 m.

4.5 DA-MRL in full 6-DOF dual-UR5 control

Full dual-UR5 control retains the reduced-order task formulation, centralized policy optimization, and actor–critic adaptation procedure. The policy dimensions change with the kinematics, so the 6-DOF system uses an independently trained initialization. Its evaluation examines adaptation within the full 6-DOF system, comparison with alternative Meta-RL methods, and the corresponding changes relative to the 3-DOF results.

4.5.1 6-DOF implementation and evaluation protocol

The dual-UR5 policy commands all six joints of each manipulator, rather than the three joints per arm used in the reduced-order system. The 48-dimensional observation in Eq. (5) is separated into two 24-dimensional arm blocks and also encoded as the global context supplied to both actor branches. Their outputs form the 12-dimensional joint action, while the critic evaluates the complete observation. Meta-training and adaptation to an evaluation task follow Algorithms 1 and 2, respectively. At the control interface, $a_t \in [-1, 1]^{12}$ represents normalized joint-velocity commands. The MuJoCo controller runs at 50 Hz, applies a first-order filter with coefficient 0.250, limits desired joint velocity to 0.800 rad/s and acceleration to 2.000 rad/s^2, and enforces the UR5 joint limits. These constraints define the

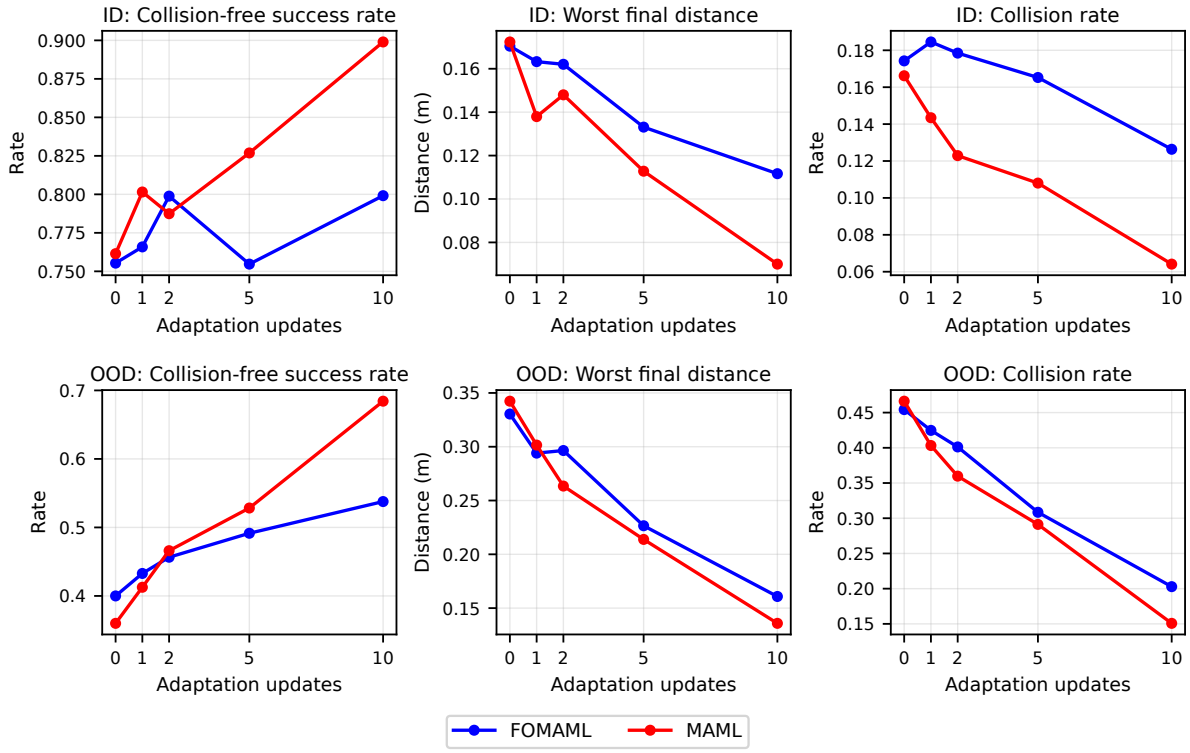


Fig. 6 3-DOF FOMAML and MAML adaptation. Rows denote ID and OOD; columns show strict collision-free success, MWFD, and collision rate. Each point averages three seeds, 20 tasks per domain, and 20 query rollouts per task.

Table 3 3-DOF FOMAML and MAML adaptation under ID/OOD shifts. Values average three seeds, 20 tasks per domain, and 20 query rollouts per task; each update adds 500 support transitions. MWFD is defined in Eq. (20), and arrows indicate the preferred direction.

Method	Domain	Metric	0	1	2	5	10
FOMAML	ID	Success rate (%) ↑	75.5	76.6	79.9	75.5	79.9
		Collision rate (%) ↓	17.4	18.5	17.8	16.5	12.6
		MWFD (m) ↓	0.170	0.163	0.162	0.133	0.112
FOMAML	OOD	Success rate (%) ↑	40.0	43.3	45.7	49.1	53.8
		Collision rate (%) ↓	45.4	42.5	40.1	30.8	20.3
		MWFD (m) ↓	0.330	0.294	0.296	0.227	0.161
MAML	ID	Success rate (%) ↑	76.1	80.2	78.7	82.7	89.9
		Collision rate (%) ↓	16.6	14.3	12.3	10.8	6.4
		MWFD (m) ↓	0.172	0.138	0.148	0.113	0.070
MAML	OOD	Success rate (%) ↑	36.0	41.3	46.6	52.8	68.4
		Collision rate (%) ↓	46.6	40.3	36.0	29.1	15.1
		MWFD (m) ↓	0.342	0.302	0.263	0.214	0.136

command actually executed during training and evaluation.

The 6-DOF meta-training distribution contains 100 fixed task definitions. Target centers are sampled within $x \in [0.420, 0.480]$ m, $y \in [-0.040, 0.040]$ m, and $z \in [1.060, 1.140]$ m. Obstacle motion varies through an orbital radius of 0.160–0.320 m, an angular velocity of 0.000–0.480 rad/s, and a phase of 0–2π, with the obstacle radius fixed at 0.025 m. The orbit is centered on the sampled target center, and each task defines a subtarget for each arm. The executed task is further varied through perturbations of up to ± 0.035 rad around the initial home configuration,

dynamics scaling of 0.850–1.150, a standard deviation of 0.000–0.010 for the observation noise, and an action delay of zero or one control step.

Meta-training runs for 400 iterations with eight tasks sampled without replacement per iteration. Each task contributes 500 support and 1,000 query transitions, giving 4.8×10^6 environment transitions per method and seed. Support data adapt the task-specific parameters, whereas the query data contribute to the outer update, as defined in Section 3. The resulting checkpoint is selected on validation data before adaptation is evaluated.

Evaluation uses fixed 6-DOF ID and OOD manifests of 20 tasks per domain. ID tasks retain the training ranges. OOD target coordinates expand to $x \in [0.361, 0.534]$ m, $y \in [-0.106, 0.106]$ m, and $z \in [0.992, 1.217]$ m; obstacle radii span 0.037–0.049 m and angular velocities 0.519–0.793 rad/s. Dynamics scaling spans 0.745–1.284, observation-noise standard deviation 0.020–0.039, action delay two to four steps, and initial-joint perturbations up to ± 0.080 rad. The obstacle orbit remains target-centered.

The 6-DOF study follows the three-seed aggregation protocol in Section 4.1. Strict success requires both end effectors to remain within 0.100 m of their subtargets for ten consecutive control steps without collision. Collision termination follows MuJoCo contact or geometric overlap between the arms or between an arm and the obstacle; inter-arm and obstacle clearances of 0.030 and 0.003 m are monitored as margins. Validation checkpoints are selected lexicographically by higher collision-free success, lower collision rate, lower MWFD, and then higher return.

4.5.2 Adaptation on 6-DOF ID and OOD tasks

Panel A of Table 4 follows the independently trained 6-DOF initializations through 0, 1, 2, 5, and 10 adaptation updates. In ID, MAML raises collision-free success from 40.2% to 67.3%, a gain of 27.1 percentage points, while FOMAML rises from 41.7% to 64.2%, a gain of 22.5 points. Their collision rates also fall, from 38.2% to 20.5% for MAML and from 36.2% to 22.2% for FOMAML. The reduction in MWFD for both variants indicates that the improvement in bilateral completion is accompanied by a closer final approach to the two subtargets.

Under the OOD shift, MAML success increases from 28.6% to 55.1%, compared with 30.1% to 52.4% for FOMAML. MAML also lowers collision rate from 49.5% to 27.7% and MWFD from 0.459 to 0.273 m. FOMAML has higher success before adaptation in both domains, while MAML has higher success and lower collision rate after ten updates. The rows for one and two updates contain local reversals, followed by improvement at five and ten updates. In ID, FOMAML retains the lower MWFD after ten updates (0.218 versus 0.232 m), while MAML has higher success and lower collision rate. The endpoint mean differences between MAML and FOMAML are smaller in the 6-DOF setting than in the 3-DOF study. Accordingly, the 6-DOF results support successful adaptation with both meta-gradient formulations, while the reported endpoint means favor MAML in success rate and collision rate.

4.5.3 Comparison with alternative Meta-RL methods

Panel B compares DA-MRL with Meta-DQN, Meta-TDQN, Meta-DDPG, and Meta-DADDPG under the same manifests, support budget, collision definition, and three-seed protocol. After ten updates, DA-MRL reaches 67.3% ID and 55.1% OOD success; Meta-TDQN reaches 5.0% in ID, and the other baseline endpoints remain at zero success. Meta-DADDPG illustrates the distinction between approach and completion: its OOD MWFD falls from 0.815 to 0.291 m while collision rises from 25.0% to 90.0%.

4.5.4 Adaptation across 3-DOF and 6-DOF configurations

Table 5 aligns the results before adaptation and after ten updates for the separately trained 3-DOF and 6-DOF systems. Both use the same DA-MRL learning and adaptation procedure, but they differ in kinematics, task geometry, control interface, and success criterion. In particular, 6-DOF success requires both arms to hold their subtargets for ten consecutive control steps. The comparison centers on the change produced by ten updates within each configuration.

For MAML, collision-free success increases by 13.8 and 32.4 percentage points in the 3-DOF ID

Table 4 6-DOF adaptation results. Panel A compares DA-MRL meta-gradients; Panel B compares alternative Meta-RL baselines at 0 and 10 updates. Values average three seeds, 20 tasks per domain, and 20 query rollouts per task. Best values are bolded within each Panel-A domain and each Panel-B domain–update setting.

Method	Domain	Updates	Success (%) ↑	Collision (%) ↓	MWFD (m) ↓
Panel A: DA-MRL meta-gradient ablation					
FOMAML	ID	0	41.7	36.2	0.362
		1	45.3	33.5	0.345
		2	43.8	35.1	0.354
		5	54.9	28.6	0.277
		10	64.2	22.2	**0.218**
FOMAML	OOD	0	30.1	46.9	0.440
		1	33.3	43.5	0.413
		2	32.0	45.2	0.424
		5	42.2	36.2	0.352
		10	52.4	29.8	0.286
MAML	ID	0	40.2	38.2	0.383
		1	47.1	31.5	0.351
		2	45.7	32.9	0.363
		5	58.6	25.2	0.294
		10	**67.3**	**20.5**	0.232
MAML	OOD	0	28.6	49.5	0.459
		1	34.9	42.2	0.422
		2	33.2	43.9	0.435
		5	44.7	34.3	0.342
		10	**55.1**	**27.7**	**0.273**
Panel B: Alternative Meta-RL baseline comparison at the endpoint budgets					
DA-MRL(Ours)	ID	0	**40.2**	**38.2**	0.383
		10	**67.3**	**20.5**	**0.232**
Meta-DDPG	ID	0	0.0	65.0	0.389
		10	0.0	90.0	0.334
Meta-DADDPG	ID	0	0.0	40.0	0.705
		10	0.0	95.0	0.330
Meta-DQN	ID	0	0.0	95.0	**0.329**
		10	0.0	80.0	0.457
Meta-TDQN	ID	0	0.0	75.0	0.483
		10	5.0	80.0	0.301
DA-MRL(Ours)	OOD	0	**28.6**	49.5	0.459
		10	**55.1**	**27.7**	**0.273**
Meta-DDPG	OOD	0	0.0	65.0	0.441
		10	0.0	95.0	0.417
Meta-DADDPG	OOD	0	0.0	**25.0**	0.815
		10	0.0	90.0	0.291
Meta-DQN	OOD	0	0.0	100.0	**0.373**
		10	0.0	65.0	0.503
Meta-TDQN	OOD	0	0.0	65.0	0.566
		10	0.0	80.0	0.386

and OOD settings, respectively, and by 27.1 and 26.5 points in the corresponding 6-DOF settings. FOMAML gains 4.4 and 13.8 points in 3-DOF and 22.5 and 22.3 points in 6-DOF. Collision rate and MWFD decline from the initial evaluation to the tenth update for every combination of method and domain shown. The full dual-UR5 system exhibits the adaptation pattern observed in the 3-DOF study. The 6-DOF MAML policy obtained after ten updates is subsequently evaluated on the physical platform.

Table 5 3-DOF and 6-DOF endpoint results by domain, meta-gradient, and adaptation budget. Each value averages three seeds, 20 tasks per domain, and 20 query rollouts per task.

Method	Domain	Updates	3-DOF			6-DOF		
			Success (%) ↑	Collision (%) ↓	MWFD (m) ↓	Success (%) ↑	Collision (%) ↓	MWFD (m) ↓
FOMAML	ID	0	75.5	17.4	0.170	41.7	36.2	0.362
		10	79.9	12.6	0.112	64.2	22.2	0.218
	OOD	0	40.0	45.4	0.330	30.1	46.9	0.440
		10	53.8	20.3	0.161	52.4	29.8	0.286
MAML	ID	0	76.1	16.6	0.172	40.2	38.2	0.383
		10	89.9	6.4	0.070	67.3	20.5	0.232
	OOD	0	36.0	46.6	0.342	28.6	49.5	0.459
		10	68.4	15.1	0.136	55.1	27.7	0.273

4.6 Sim-to-real deployment

4.6.1 Physical deployment protocol

The same 6-DOF MAML policy checkpoint obtained after ten adaptation updates in simulation is used in all six physical records. Its network parameters remain fixed during hardware execution; the centralized 48-dimensional observation continues to change with the measured robot and workspace state.

Visual observations come from a fixed Angstrong HP60C RGB-D camera in an eye-to-hand configuration overlooking two UR5 manipulators, a target, and a motor-driven rotating obstacle. Synchronized RGB and depth observations are acquired at 640×480 pixels and 20 Hz. The target and obstacle are detected in each synchronized RGB observation, and registered depth values expressed in millimetres provide their three-dimensional positions in the camera frame.

The camera-frame detections are transformed into the world frame using the RGB intrinsics ($f_x = 584.583$, $f_y = 584.618$, $c_x = 336.039$, and $c_y = 236.530$) and a fixed eye-to-hand extrinsic transform. Robot joint and end-effector states are combined with the transformed target and obstacle measurements to populate the centralized observation expected by the simulated policy. The camera and robot calibration parameters remain fixed throughout the six records.

Each synchronized camera observation triggers one policy inference and one high-level command update at 20 Hz. The resulting command is held between policy updates and tracked by the low-level controller at 500 Hz. Robot states and deployment measures are logged at 100 Hz, while the sinusoidal obstacle command is updated at 50 Hz. The deployment layer constrains and smooths the joint command while monitoring obstacle and inter-arm clearance.

Hardware safety is provided by the controller cabinet, teach-pendant functions, and the physical emergency stop. Planned motions are checked against the planning scene, while policy-generated joint commands remain subject to the configured joint and velocity limits. The software emergency stop cancels active commands and blocks new commands until manual release. Force/torque measurements support execution monitoring, and visual detections provide the target and obstacle states used by the policy.

The obstacle follows constant- or sinusoidal-velocity motion at maximum angular speeds of 0.28, 0.48, and 0.68 rad/s. Constant-velocity motion rotates in the positive direction at $\omega_{\max}$. Under sinusoidal-velocity motion, the obstacle also rotates in the positive direction with an 8-s period, and its speed varies from $0.35\omega_{\max}$ to $\omega_{\max}$. The respective ranges are 0.098–0.28, 0.168–0.48, and 0.238–0.68 rad/s. Each of the six combinations of motion profile and speed produces one continuous 200-s record containing 20,000 samples logged at 100 Hz. The policy and deployment interface are unchanged across these records.

4.6.2 Physical evaluation measures and outcomes

Figure 7 shows preparation, left- and right-arm avoidance, and coordinated target approach in a rollout without collision-induced termination at 0.48 rad/s. As the obstacle traverses the workspace, the arm making the larger adjustment changes before both arms resume target approach; the lower row shows the corresponding perception output.

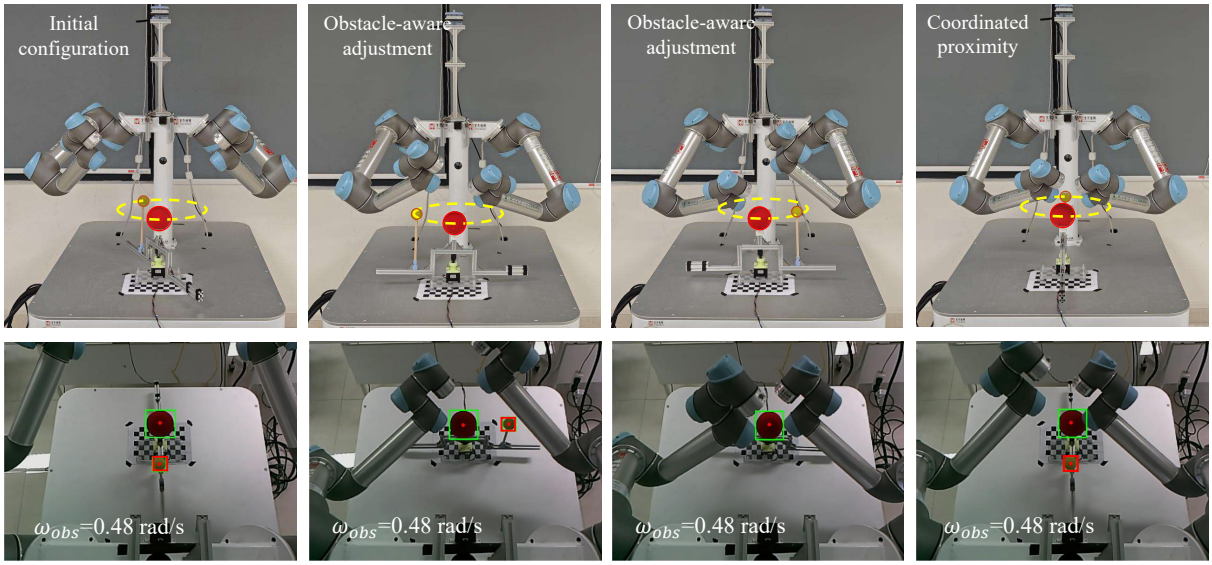


Fig. 7 Representative keyframes of sim-to-real deployment at a maximum obstacle angular velocity of 0.48 rad/s. From left to right: preparation, left-arm obstacle avoidance, right-arm obstacle avoidance, and coordinated approach toward the target regions. Red circles indicate the target-region boundaries, yellow curves indicate the obstacle trajectory, and the green and red bounding boxes in the camera views represent the detected targets and obstacle, respectively.

Four deployment metrics are reported. Simultaneous target-region occupancy(STO) is the fraction of logged steps at which both end effectors lie within their corresponding target regions:

$$\mathrm{STO} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{I}\big(d^{L}_{\mathrm{tar},t} \le \rho \ \wedge \ d^{R}_{\mathrm{tar},t} \le \rho\big)\,. \quad (22)$$

Here T is the number of logged steps, $d^{L}_{\mathrm{tar},t}$ and $d^{R}_{\mathrm{tar},t}$ are the target distances of the left and right arms, and $\rho = 0.10$ m is the radius of the target region. Requiring both inequalities at the same logged step captures simultaneous occupation of the two target regions. This measure is computed over logged samples and differs from the collision-free success rate computed over complete episodes in simulation. Collision-induced termination is recorded as a binary outcome for each complete 200-s deployment record. The time averages below include preparation, obstacle avoidance, and continued execution after arrival within each record. The time-averaged two-arm target distance and the time-averaged smaller of the two arm-to-obstacle surface clearances are defined, respectively, as

$$\bar{d}_{\mathrm{tar}} = \frac{1}{T}\sum_{t=1}^{T} \frac{d^{L}_{\mathrm{tar},t} + d^{R}_{\mathrm{tar},t}}{2}, \quad (23)$$

$$\bar{d}^{*}_{\mathrm{obs}} = \frac{1}{T}\sum_{t=1}^{T} \min\big(d^{L}_{\mathrm{obs},t}, d^{R}_{\mathrm{obs},t}\big)\,. \quad (24)$$

Here, $d^{L}_{\mathrm{obs},t}$ and $d^{R}_{\mathrm{obs},t}$ denote the instantaneous obstacle surface clearances of the two arms. The quantity $\bar{d}^{*}_{\mathrm{obs}}$ averages the nearer-arm clearance at each logged step rather than taking the minimum clearance over the entire record.

None of the six deployment records in Table 6 ends because of collision-induced termination. Simultaneous target-region occupancy ranges from 0.406 to 0.491 under constant-velocity motion and from 0.461 to 0.511 under sinusoidal motion. The time-averaged target distance remains within 0.114–0.117 m, and the time-averaged nearer-arm obstacle clearance remains within 0.257–0.262 m. Occupancy varies with motion profile and speed, while the two distance measures occupy narrower ranges. The keyframes and logged measures together describe execution of the frozen policy adapted in simulation under both obstacle motion profiles.

Table 6 Physical deployment results after 10 adaptation updates. Each combination of motion profile and speed contains one 200-s record with 20,000 samples logged at 100 Hz. Target-region occupancy is evaluated over logged samples, and collision-induced termination is reported for each complete record.

Metric	$\omega_{\max}$ (rad/s)		
	0.28	0.48	0.68
Constant-velocity motion			
STO ↑	0.406	0.461	0.491
Collision-induced termination	No	No	No
$\bar{d}_{\text{tar}}(m)$ ↓	0.117	0.114	0.115
$\bar{d}^*_{\text{obs}}(m)$ ↑	0.262	0.258	0.257
Sinusoidal-velocity motion			
STO ↑	0.511	0.461	0.482
Collision-induced termination	No	No	No
$\bar{d}_{\text{tar}}(m)$ ↓	0.117	0.115	0.115
$\bar{d}^*_{\text{obs}}(m)$ ↑	0.259	0.258	0.257

5 Conclusion

This paper presented DA-MRL, a centralized meta-reinforcement learning framework that learns a task-generalized actor–critic initialization for rapid adaptation in dynamic dual-arm obstacle avoidance. The framework coordinates both manipulators through joint state and action modeling and adapts to unseen target configurations, obstacle motions, sensing conditions, delays, and dynamics using limited support trajectories. In MuJoCo, ten adaptation updates increased collision-free success from 76.1% to 89.9% on 3-DOF ID tasks and from 36.0% to 68.4% on 3-DOF OOD tasks, while reducing the corresponding collision rates from 16.6% to 6.4% and from 46.6% to 15.1%. This improvement was retained in the more challenging 6-DOF system, where success increased from 40.2% to 67.3% for ID tasks and from 28.6% to 55.1% for OOD tasks. The simultaneous reductions in collision rate and MWFD indicate that DA-MRL improves safe bilateral task completion rather than target approach alone. The simulation-adapted policy was further transferred to a physical dual-UR5 platform without hardware-side parameter updates. None of the six 200-s deployment records, each containing 20,000 logged samples, encountered the defined collision-induced termination condition. Simultaneous target-region occupancy ranged from 0.406 to 0.511, while the average target distance and nearer-arm obstacle clearance remained within 0.114–0.117 m and 0.257–0.262 m, respectively. These results demonstrate coordinated obstacle avoidance and target approach under the evaluated sim-to-real conditions.

Nevertheless, the physical evaluation is limited to one robot setup, one obstacle geometry, and six deployment records. Future work will investigate safe real-world adaptation, perception uncertainty, more diverse and human-driven obstacles, and broader validation across robots, workspaces, and manipulation tasks.

References

[1] Aljalbout E, Chen J, Ritt K, et al (2021) Learning vision-based reactive policies for obstacle avoidance. In: Proceedings of the 2020 Conference on Robot Learning, pp 2040–2054, URL https://proceedings.mlr.press/v155/aljalbout21a.html

[2] Cao G, Bai J (2025) Multi-agent deep reinforcement learning-based robotic arm assembly research. PLOS ONE 20(2):e0311550. https://doi.org/10.1371/journal.pone.0311550

[3] Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 70. PMLR, pp 1126–1135, URL https://proceedings.mlr.press/v70/finn17a.html

[4] Flacco F, Krüger T, De Luca A, et al (2012) A depth space approach to human-robot collision avoidance. In: IEEE International Conference on Robotics and Automation, pp 338–345, https://doi.org/10.1109/ICRA.2012.6225245

[5] Forlini M, Neri F, Ciccarelli M, et al (2024) Experimental implementation of skeleton tracking for collision avoidance in

collaborative robotics. The International Journal of Advanced Manufacturing Technology 134:57–73. https://doi.org/10.1007/s00170-024-14104-7

[6] Haddadin S, Albu-Schäffer A, De Luca A, et al (2008) Collision detection and reaction: A contribution to safe physical human-robot interaction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 3356–3363, https://doi.org/10.1109/IROS.2008.4650764

[7] Ji G, Gao Q, Sun M, et al (2023) Surgical continuum manipulator control using multiagent team deep q learning. In: 2023 45th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp 1–5, https://doi.org/10.1109/EMBC40787.2023.10340943

[8] Kalakrishnan M, Chitta S, Theodorou E, et al (2011) Stomp: Stochastic trajectory optimization for motion planning. In: 2011 IEEE International Conference on Robotics and Automation, pp 4569–4574, https://doi.org/10.1109/ICRA.2011.5980280

[9] Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research 5(1):90–98. https://doi.org/10.1177/027836498600500106

[10] Lillicrap TP, Hunt JJ, Pritzel A, et al (2016) Continuous control with deep reinforcement learning. In: International Conference on Learning Representations, URL https://openreview.net/forum?id=BxYsP-7ggf

[11] Lin Y, Church A, Yang M, et al (2023) Bi-touch: Bimanual tactile manipulation with sim-to-real deep reinforcement learning. IEEE Robotics and Automation Letters 8(9):5472–5479. https://doi.org/10.1109/LRA.2023.3295991

[12] Liu L, Liu Q, Song Y, et al (2021) A collaborative control method of dual-arm robots based on deep reinforcement learning. Applied Sciences 11(4):1816. https://doi.org/10.3390/app11041816

[13] Maciejewski AA, Klein CA (1985) Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. The International Journal of Robotics Research 4(3):109–117. https://doi.org/10.1177/027836498500400308

[14] Mnih V, Kavukcuoglu K, Silver D, et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533. https://doi.org/10.1038/nature14236

[15] Moe S, Pettersen KY, Gravdahl JT (2020) Set-based collision avoidance applications to robotic systems. Mechatronics 69:102399. https://doi.org/10.1016/j.mechatronics.2020.102399

[16] Perrusquía A, Yu W, Li X (2021) Multi-agent reinforcement learning for redundant robot control in task-space. International Journal of Machine Learning and Cybernetics 12(1):231–241. https://doi.org/10.1007/s13042-020-01167-7

[17] Rakelly K, Zhou A, Finn C, et al (2019) Efficient off-policy meta-reinforcement learning via probabilistic context variables. In: Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 97. PMLR, pp 5331–5340, URL https://proceedings.mlr.press/v97/rakelly19a.html

[18] Rothfuss J, Lee D, Clavera I, et al (2019) Promp: Proximal meta-policy search. In: International Conference on Learning Representations, URL https://openreview.net/forum?id=SkxXCi0qFX

[19] Rusu AA, Večerík M, Rothörl T, et al (2017) Sim-to-real robot learning from pixels with progressive nets. In: Proceedings of the 1st Annual Conference on Robot Learning, pp 262–270, URL https://proceedings.mlr.press/v78/rusu17a.html

[20] Sangiovanni B, Rendiniello A, Incremona GP, et al (2018) Deep reinforcement learning for collision avoidance of robotic manipulators. In: 2018 European Control Conference, pp 2063–2068, https://doi.org/10.23919/ECC.2018.8550363

[21] Scalera L, Giusti A, Vidoni R, et al (2022) Enhancing fluency and productivity in human-robot collaboration through online scaling of dynamic safety zones. The International Journal of Advanced Manufacturing Technology 121:6783–6798. https://doi.org/10.1007/s00170-022-09781-1

[22] Schulman J, Duan Y, Ho J, et al (2014) Motion planning with sequential convex optimization and convex collision checking. The International Journal of Robotics Research 33(9):1251–1270. https://doi.org/10.1177/0278364914528132

[23] Schulman J, Wolski F, Dhariwal P, et al (2017) Proximal policy optimization algorithms. arXiv preprint arXiv:170706347 URL https://arxiv.org/abs/1707.06347

[24] Shahid AA, Sesin JSV, Pecioski D, et al (2021) Decentralized multi-agent control of a manipulator in continuous task learning. Applied Sciences 11(21):10227. https://doi.org/10.3390/app112110227

[25] Yamada J, Lee Y, Salhotra G, et al (2021) Motion planner augmented reinforcement learning for robot manipulation in obstructed environments. In: Proceedings of the 2020 Conference on Robot Learning, pp 589–603, URL https://proceedings.mlr.press/v155/yamada21a.html

[26] Yu T, Quillen D, He Z, et al (2020) Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In: Proceedings of the Conference on Robot Learning, Proceedings of Machine Learning Research, vol 100. PMLR, pp 1094–1100, URL https://proceedings.mlr.press/v100/yu20a.html

[27] Zhu T, Mao J, Han L, et al (2024) Real-time dynamic obstacle avoidance for robot manipulators based on cascaded nonlinear mpc with artificial potential field. IEEE Transactions on Industrial Electronics 71(7):7424–7434. https://doi.org/10.1109/TIE.2023.3306405

[28] Zucker M, Ratliff N, Dragan AD, et al (2013) Chomp: Covariant hamiltonian optimization for motion planning. The International Journal of Robotics Research 32(9–10):1164–1193. https://doi.org/10.1177/0278364913488805

Statements and Declarations

Funding. This work was supported by the Natural Science Basic Research Program of Shaanxi (Grant No. 2026JC-YXQN-155) and the Fundamental Research Funds for the Central Universities, CHD (Grant No. 300102255301).

Competing Interests. The authors declare that they have no competing interests.

Author Contributions. Kai Ding supervised the research and reviewed and edited the manuscript. Lingxiao Hu developed the methodology, implemented the algorithms, conducted the experiments, analyzed the results, and prepared the original manuscript. Yang Zhang contributed to research conceptualization, methodological guidance, and manuscript revision. Hong Wang contributed to methodological guidance and manuscript review and editing. All authors read and approved the final manuscript.

Data Availability. The datasets generated and/or analyzed during the current study are not publicly available but are available from the corresponding author upon reasonable request.

Code Availability. The code developed during the current study is not publicly available but is available from the corresponding author upon reasonable request.

Ethics Approval and Consent to Participate. Not applicable.

Consent for Publication. Not applicable.