Proceedings of the 8th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/2026.37819

A Training Framework for Strategic Badminton Rally Robots Based on Progressive Curriculum Reinforcement Learning

Xingzhi Sun

Beijing No.2 Middle School, Beijing, China
sb_18601035190@qq.com

Abstract. This study develops and evaluates a four-stage (V1–V4) progressive curriculum reinforcement learning framework for strategic badminton rallies between two simulated mobile manipulators. Each agent combines a Clearpath Ridgeback omnidirectional base, a Franka Panda arm, and a racket end effector in NVIDIA Isaac Lab. The task is formulated as a finite-horizon Markov decision process and optimized with Proximal Policy Optimization. The curriculum progresses from fixed-base single-shot interception (V1), through dual-robot mobile rallies with staged serve noise (V2), asymmetric tactical placement (V3), and symmetric full-court placement with predicted-intercept rewards and a product-form difficulty schedule (V4). The reported learning curves indicate that each stage acquired the behavior emphasized by its reward and curriculum design: stable interception in V1, improved rally continuity in V2, lateral placement and pursuit in V3, and more difficult tactical targets in V4. These results support progressive curriculum design as a practical way to mitigate sparse rewards in this simulated task. However, the evidence is limited to one simulation configuration and descriptive training curves; physical-robot validation, repeated-seed analysis, ablation studies, and non-curricular baselines remain necessary. The framework should therefore be interpreted as a simulation proof of concept and a basis for future sim-to-real evaluation, richer shot-type control, and opponent-diverse self-play.

Keywords: Badminton Robot, Curriculum Learning, Embodied Artificial Intelligence, Mobile Manipulation, Reinforcement Learning

1. Introduction

Embodied artificial intelligence (embodied AI) extends artificial intelligence from disembodied perception and computation to agents that sense, act, and adapt in physical environments. This interaction-centered perspective is increasingly important in robotics because task performance depends on the coupling among perception, control, morphology, and environmental dynamics [1].

Robotic skill acquisition may use supervised, imitation, or reinforcement learning, and these paradigms are complementary rather than mutually exclusive. Supervised and imitation learning can exploit curated demonstrations, whereas reinforcement learning (RL) directly optimizes behavior through interaction and task feedback. RL is therefore attractive for dynamic control tasks in which exhaustive action labels are difficult to provide, although its performance depends strongly on simulation fidelity, reward design, exploration, and sample efficiency.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.37819

Badminton is a demanding benchmark for embodied control. Shuttlecock flight is dominated by high aerodynamic drag and rapid deceleration, producing trajectories that differ markedly from those of approximately spherical balls [2]. Experimental trajectory studies further show that launch conditions and aerodynamic parameters materially affect the flight path [3]. A badminton robot must therefore coordinate interception timing, racket pose, whole-body motion, and return placement under substantial state uncertainty.

Recent sports-robotics studies demonstrate coordinated badminton play by legged manipulators [4] and force-aware dynamic manipulation by legged systems [5]. Nevertheless, a simulated rally framework still faces several coupled challenges: shuttlecock dynamics, fast contact timing, mobile manipulation, alternating-agent coordination, sparse rewards, and tactical target selection.

This study addresses these challenges with a four-stage training pipeline based on Proximal Policy Optimization (PPO) [6] and curriculum learning, which organizes training examples or task conditions from easier to more difficult configurations [7]. The system uses two-layer multilayer perceptrons with 128 hidden units per layer, a composite phase-dependent reward, a rally state machine, and GPU-parallel simulation. The contributions are: (1) a progression from fixed-base interception to dual-mobile rallies; (2) an asymmetric-to-symmetric tactical curriculum; and (3) predicted-intercept and target-expansion mechanisms for full-court placement. All evidence reported here is simulation-based; the work does not claim human-level play or validated real-world transfer.

2. Literature review

This section positions the proposed framework relative to recent work on robotic badminton, curriculum-based mobile manipulation, competitive sports robots, and simulation-aware actor–critic methods.

Ma et al. trained a quadrupedal mobile manipulator (ANYmal-D with a DynaArm) for badminton and validated the policy on real hardware [4]. Their system incorporated perceptual uncertainty, actuator constraints, and online shuttlecock-trajectory estimation. It provides an important demonstration of coordinated whole-body control and simulation-to-reality transfer for robotic badminton.

Liu et al. trained a humanoid badminton system using a three-stage RL curriculum that separated footwork, racket-swing skill, and whole-body coordination [8]. Their staged design illustrates how decomposing a fast, contact-rich task can reduce the difficulty of learning all behaviors simultaneously.

For wheel-legged loco-manipulation, Wang et al. incorporated arm-motion constraints into the learning objective and used a four-stage curriculum that gradually combined locomotion and manipulation [9]. This work is relevant to the present system because it treats mobility, arm safety, and task execution as coupled rather than independent control problems.

Zhalehmehrabi investigated manually designed and teacher–student curricula for robotic air-hockey control, reporting stronger performance for the manual curriculum in the studied configuration [10]. Orsula likewise showed that a dynamic air-hockey agent can learn from sparse rewards with model-based RL and self-play, while also observing overfitting to a single opponent style [11]. Together, these studies motivate both the hand-designed progression used here and the need for more diverse opponents in future work.

Pinto et al. introduced an asymmetric actor–critic method in which the actor uses observations available at deployment while the critic can exploit privileged simulator state during training [12]. The present study instead uses a symmetric actor–critic architecture with the same 81-dimensional

DOI: 10.54254/2755-2721/2026.37819

state vector for both networks. This choice simplifies implementation but does not exploit privileged information and should not be described as asymmetric actor–critic learning.

3. Methodology

3.1. System architecture

The proposed framework integrates a physics-based simulation environment, PPO-based strategy learning, and a progressive curriculum for dual-robot badminton rallies. Figure 1 summarizes the three modules.

• Simulation environment: NVIDIA Isaac Sim and Isaac Lab provide the court, rigid-body dynamics, shuttlecock flight model, robot assets, and parallel environment instances.

• Reinforcement learning: PPO, a symmetric actor–critic network, receiver-gated actions, and a phase-dependent composite reward are used to learn interception, return, and placement behaviors.

• Progressive curriculum: training advances from fixed-base single-shot interception to mobile rallies, asymmetric tactical placement, and symmetric full-court target selection with increasing variability.

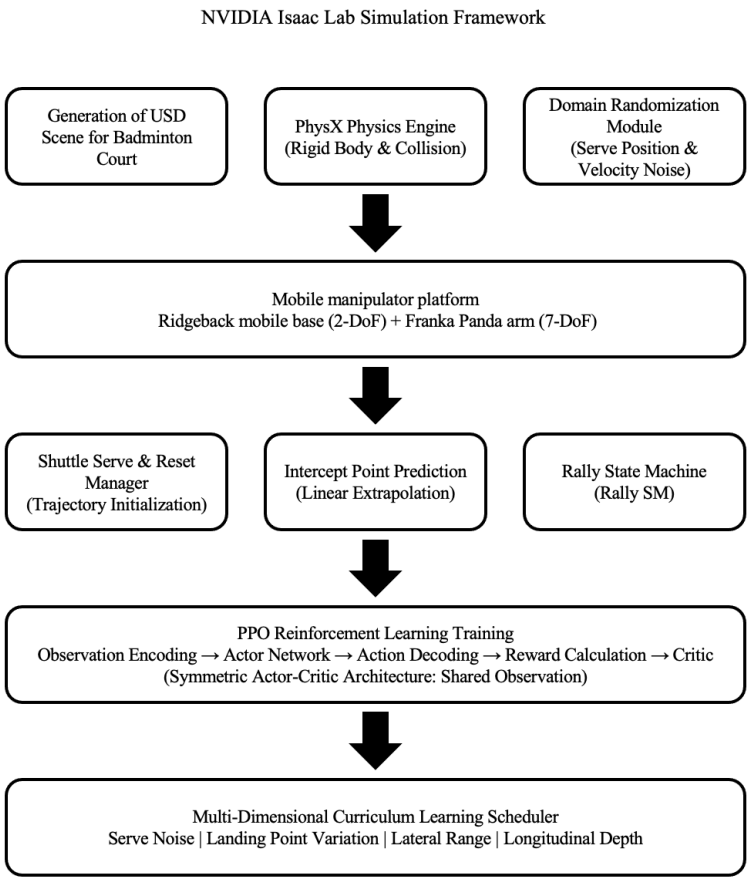

Figure 1. Overall architecture of the training system

DOI: 10.54254/2755-2721/2026.37819

3.2. Simulation environment

3.2.1. Simulation platform

NVIDIA Isaac Lab is used as the core robot-learning framework. Isaac Lab evolved from the Orbit framework and is built on NVIDIA Isaac Sim, providing modular environments and GPU-accelerated physics for large-scale robot learning [13]. This implementation instantiates multiple environments on one GPU and uses PhysX rigid-body simulation for court, robot, racket, and shuttlecock interactions.

3.2.2. Badminton court

The court is constructed in OpenUSD using the Pixar USD Python API. Its geometry follows the Badminton World Federation (BWF) Laws of Badminton [14].

The doubles court measures 13.40 m × 6.10 m, while the singles width is 5.18 m. A planar court surface is marked with 0.04 m boundary lines for the baselines, sidelines, and service areas [14].

The net is located at x = 0, with posts 1.55 m high and a court-spanning length of 6.10 m. The net is represented by an aggregate mesh with a thickness of 0.075 m, and collision is enabled through the UsdPhysics collision API.

A dome light and environment lighting provide scene illumination. Physically based rendering materials are used for visual inspection; visual realism is not used as an input to the state-based policy.

3.2.3. Domain randomization

To improve robustness and prepare for future simulation-to-reality transfer, the training environment randomizes serve position and velocity. Domain randomization exposes a policy to a distribution of simulated conditions and has been used to support transfer from simulation to physical systems [15]. In this study, the noise magnitude is increased through the curriculum schedule shown in Table 1.

Table 1. Domain-randomization parameters

Category	Parameter	Range	Distribution
Serve	Longitudinal Position Noise (m)	0 → ±0.25 (progressive)	Uniform
Serve	Lateral Position Noise (m)	0 → ±0.38 (progressive)	Uniform
Serve	Longitudinal Velocity Noise (m/s)	0 → ±0.35 (progressive)	Uniform
Serve	Lateral Velocity Noise (m/s)	0 → ±0.16 (progressive)	Uniform

DOI: 10.54254/2755-2721/2026.37819

3.3. Robot platform modeling

3.3.1. Franka Panda arm

The Franka Emika Panda seven-degree-of-freedom arm is used as the striking actuator. It is a widely used research platform for manipulation and physical interaction [16]. The model includes seven revolute joints, an approximate reach of 0.855 m, a payload capacity of 3 kg, and the joint limits listed in Table 2.

Table 2. Franka Panda joint limits

Joint	Joint Limits (rad)	Max Velocity (rad/s)	Max Torque (Nm)
Joint 1	[-2.897, 2.897]	2.175	87
Joint 2	[-1.762, 1.762]	2.175	87
Joint 3	[-2.897, 2.897]	2.175	87
Joint 4	[-3.071, -0.069]	2.175	87
Joint 5	[-2.897, 2.897]	2.610	12
Joint 6	[-0.017, 3.752]	2.610	12
Joint 7	[-2.897, 2.897]	2.610	12

3.3.2. Mobile base and full-court coverage

For mobile court coverage, the arm is mounted on a Clearpath Ridgeback omnidirectional base. Ridgeback uses four Mecanum wheels to support planar omnidirectional motion [17]. In simulation, the base is abstracted as two prismatic degrees of freedom corresponding to longitudinal (x-axis) and lateral (y-axis) movement.

The base workspace is limited to x in [-1.20, 1.20] m and y in [-1.50, 1.50] m. A proportional-derivative (PD) controller tracks the target position produced by the policy, as defined in Equation (1):

$$\tau_{base} = K_p\left(q_{target} - q_{current}\right) + K_d(\dot{q}_{target} - \dot{q}_{current}) \quad (1)$$

where Kp = 100.0 and Kd = 10.0 are the position and velocity gains, respectively; qtarget and qcurrent denote the target and current base coordinates.

3.3.3. Racket model

The badminton racket is attached as the arm end effector. Its string bed is modeled as an elliptical plane whose normal defines the racket-face orientation. Hit detection uses the central 75% of the modeled string-bed area to exclude edge contacts and enforce a consistent contact region.

DOI: 10.54254/2755-2721/2026.37819

3.4. Reinforcement learning framework

3.4.1. Markov decision process formulation

The rally task is formulated as a finite-horizon Markov decision process defined by the six-tuple (S, A, P, R, γ, H), where S is the state space, A is the action space, P is the transition model, R is the reward function, $\gamma = 0.98$ is the discount factor, and H is the episode horizon. The state contains robot proprioception, shuttlecock motion, and task context; the continuous action combines arm-joint targets and mobile-base targets; and PhysX determines state transitions under the implemented dynamics.

3.4.2. Observation space

The observation design prioritizes task-relevant information, compact dimensionality, and consistency across curriculum stages. In V2–V4, the full observation vector has 81 dimensions, as summarized in Table 3.

Table 3. Observation-space composition

Category	Observation Term	Dims	Description
Right Robot	Joint Position	7	Relative joint positions for panda_joint1-7
Right Robot	Joint Velocity	7	Current velocity of 7 revolute joints
Right Robot	Base X Position/Velocity	2	dummy_base_prismatic_x position and velocity
Right Robot	Base Y Position/Velocity	2	dummy_base_prismatic_y position and velocity
Right Robot	Racket Sweet Spot Position	3	Racket string-bed center position in world frame
Right Robot	Racket-to-Shuttle Relative Pos.	3	Shuttlecock position relative to racket sweet spot
Right Robot	String-Bed Offset	3	Normalized shuttlecock position in racket string-bed frame
Left Robot	Mirror of Right Robot	27	Observation set symmetric to Right Robot
Shared Observation	Shuttle Position	3	Shuttlecock position in world frame (x, y, z)
Shared Observation	Shuttle Velocity	3	Shuttlecock linear velocity (vx, vy, vz)
Shared Observation	Receiver Indicator	1	+1=Right, -1=Left
Shared Observation	Rally Hit Count	1	Completed rally hits (normalized ÷ 4)
Shared Observation	Time to Hit	1	Remaining time until nominal interception
Shared Observation	Previous Action	18	Previous 18-dim action vector (autoregressive policy input)

DOI: 10.54254/2755-2721/2026.37819

The dual-robot observation contains the complete modeled state of both robots and shared task variables, including shuttlecock position and velocity, receiver identity, rally hit count, time to interception, and the previous action. Both actor and critic receive this same vector.

3.4.3. Action space

The action space must coordinate rapid racket positioning with planar base motion. A hybrid target-position representation is therefore used for each robot.

Robotic-arm control: Equation (2) defines the seven joint-position increments produced by the policy relative to the default pose at each control cycle:

$$\boldsymbol{a}_{arm} = [\Delta q_1, \Delta q_2, \ldots, \Delta q_7] \tag{2}$$

Each increment is clipped to [-0.5, 0.5] rad and converted to joint torque by the PD controller in Equation (3):

$$\tau_j = K_p \cdot \left(\Delta q_j + q_j^{current} - q_j^{measured}\right) - K_d \cdot \dot{q}_j^{measured} \tag{3}$$

Mobile-base control: Equation (4) defines the two-dimensional target-position increment produced by the policy:

$$\boldsymbol{a}_{base} = [\Delta x_{base}, \Delta y_{base}] \tag{4}$$

The base increment is limited to x $\in$ [−0.75, 0.75] m and y $\in$ [−1.0, 1.0] m in the earlier stage, then expanded to x $\in$ [−1.20, 1.20] m and y $\in$ [−1.50, 1.50] m. Independent PD controllers track the targets.

Receiver-gated action: only the active receiver is permitted to execute a hitting motion. When the receiver indicator does not match a robot, that robot's action is set to zero and it returns toward a standby pose. The joint action vector contains 18 components (seven arm and two base commands per robot), although only nine components are active at a given step.

3.4.4. Reward function

Reward design strongly influences reinforcement-learning behavior. The proposed reward is a weighted sum of phase-dependent objectives for approach, contact, return quality, safety, and motion regularization. Because reward shaping can unintentionally alter the intended solution if it is poorly specified, the individual terms are tied to observable task events and safety constraints (Equation (5)):

$$R_{total} = \sum_k w_k \cdot r_k \tag{5}$$

The terms are grouped into pre-hit, hit-moment, post-hit, and global components. Table 4 summarizes the terms and indicates the curriculum versions in which each one is active.

DOI: 10.54254/2755-2721/2026.37819

Table 4. Core reward-function terms

Phase	Reward Term	Weight	Description	Versions
All Time	r_alive	0.2	Per-step survival reward, providing baseline positive feedback	v1, v2, v3, v4
Pre-Hit	r_timed_intercept	4.0 / 6.0	Gaussian-kernel time-gated interception precision reward within the hit time window (~0.55 s)	v1, v2, v3, v4
Pre-Hit	r_intercept	0.7 / 1.0	Full-time racket string-bed ellipse proximity to shuttlecock	v1, v2, v3, v4
Pre-Hit	r_approach	1.0 / 1.4	Tanh reward for racket sweet-spot distance to shuttlecock	v1, v2, v3, v4
Pre-Hit	r_base_planar_alignment	1.5 / 2.2	Mobile base planar alignment to predicted interception point	v2, v3, v4
Hit-Moment	r_forward_swing	0.25/0.35	Racket sweet-spot velocity component along the return direction	v1, v2, v3, v4
Hit-Moment	r_face_alignment	0.5 / 0.8	Racket face normal alignment with target return direction (dot product)	v2, v3, v4
Post-Hit	r_return_progress	1.2 / 1.2	Normalized progress of return toward target half-court (0–1)	v1, v2, v3, v4
Post-Hit	r_net_clearance	2.5	Height margin when the return crosses the net	v1, v2, v3, v4

DOI: 10.54254/2755-2721/2026.37819

Table 4. (continued)

Post-Hit	r_opponent_height	0.8	Airborne height of return on the opponent's side	v1, v2, v3, v4
Post-Hit	r_return_target	0.6 / 0.6	Precision of return approaching the opponent's interception point	v1, v2, v3, v4
Post-Hit	r_out_of_bounds	−6.0	Negative penalty for shuttlecock going out of bounds	v2, v3, v4
Global	r_action_rate	−0.002	smoothing penalty on action rate	v1, v2, v3, v4
Global	r_joint_vel	−0.0005	Regularization penalty on joint velocity	v1, v2, v3, v4
Global	r_base_planar_position	−0.05	Penalty on base displacement from the nominal position	v2, v3, v4
Global	r_post_hit_arm_home	−0.8	Penalty on deviation from the default arm posture after a hit	v2, v3, v4
Global	r_racket_base_clearance	−8.0	Height penalty for racket entering base safety zone (strongest)	v2, v3, v4
Global	r_base_planar_velocity	−0.004 / −0.004	penalty on mobile base planar velocity	v2, v3, v4
Global	r_base_planar_acceleration	−0.0008 / −0.0008	penalty on mobile base planar acceleration (smoothing)	v2, v3, v4
Global	r_post_hit_arm_action	−0.08 / −0.08	penalty on arm action commands after the hit phase	v2, v3, v4

DOI: 10.54254/2755-2721/2026.37819

Note: paired weights are reported as right/left values. During asymmetric training, selected positive weights are larger for the left robot to provide additional learning signal; the weights are later equalized for symmetric training.

A hit is registered only when the shuttlecock satisfies the string-bed geometry and all task-state checks at the same simulation step:

(1) receiver_mask: the robot is the active receiver;

(2) is_on_string_bed : the shuttlecock's racket-frame projection lies within the elliptical string-bed region ($x^2/l^2 + y^2/w^2 \leq 1$) and its z -offset is within the half-thickness;

(3) face_is_aligned : dot product between racket face normal and target return direction ≥ 0.75 (angular deviation $< 41.4^\circ$);

(4) incoming: the horizontal shuttlecock velocity points toward the active receiver's half-court;

(5) in_hit_window : elapsed phase time $\in [0.701.25]$ s. When all five conditions hold in the same frame, the hit event fires, writes the return velocity, and instantly swaps the receiver.

After a valid event, the simulator applies a scripted impulse by setting the shuttlecock velocity to a target return vector. The return magnitude and direction are selected by the current curriculum stage. Consequently, this work learns positioning and event timing around an abstracted contact model rather than learning the full racket–shuttle impact dynamics.

3.4.5. PPO and symmetric actor–critic architecture

The policy is optimized with PPO [6], an on-policy policy-gradient algorithm that limits each update through the clipped surrogate objective in Equation (6):

$$L^{CLIP}(\theta) = \widehat{\mathbb{E}}_t[min(r_t(\theta)\widehat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\widehat{A}_t)] \tag{6}$$

where rt(θ) is the importance-sampling ratio, Ât is the generalized advantage estimate (GAE) [18], and ε = 0.2 is the clipping threshold. Table 5 lists the PPO hyperparameters.

Table 5. PPO training hyperparameters

Hyperparameter	Value	Description
Learning Rate	1.0×10^{-3}	Adam initial learning rate, adaptive schedule
Discount Factor γ	0.98	Discount rate for long-term returns
GAE Parameter λ	0.95	Bias-variance trade-off for advantage estimation
Clipping Parameter ε	0.2	Clipping range for PPO surrogate objective
Value Loss Coefficient	1.0	Weight for value function loss in PPO objective
Use Clipped Value Loss	True	Apply clipping to value function loss for stability
Entropy Coefficient	0.002	Exploration regularization strength, preventing premature convergence
Target KL Divergence	0.01	KL threshold for adaptive learning rate adjustment
Steps per Environment	32	Experience collection steps per policy update

DOI: 10.54254/2755-2721/2026.37819

Table 5. (continued)

Learning Epochs	5	Gradient update epochs per PPO update
Mini-Batches	4	Four equal mini-batch divisions; 8,192 samples each for a 32,768-sample rollout
Initial Action Noise Std	1.0	Initial exploration noise standard deviation for action sampling
Gradient Clipping	1.0	Maximum gradient norm
Max Iterations	8000	Total training iterations
Parallel Environments	1024	GPU-parallel simulation instances
Hidden Layers	[128, 128]	Shared hidden layer dimensions for Actor and Critic
Activation Function	ELU	Nonlinear activation for hidden layers
Network Output	Actor: 18-dim Gaussian / Critic: scalar value	Actor outputs mean + std; Critic outputs state value

This study uses a symmetric actor–critic architecture. Both networks receive the complete 81-dimensional observation vector and no privileged simulator-only input. The actor passes the observation through two 128-unit ELU hidden layers and parameterizes an 18-dimensional Gaussian action distribution. The critic uses the same input and hidden-layer dimensions and outputs one scalar state-value estimate.

Both networks use the same input dimensionality and architecture. Observation normalization is disabled, weights use orthogonal initialization, and the actor's initial action standard deviation is 1.0 to encourage early exploration. Only the actor is required for policy inference.

3.4.6. Network architecture

The actor and critic are multilayer perceptrons (MLPs) with two hidden layers of 128 units. An MLP is appropriate because the observation is a fixed-length state vector rather than an image or a sequence. The compact architecture also reduces per-environment computation during parallel training.

Each hidden layer uses an exponential linear unit (ELU) activation [19], defined in Equation (7):

$$ELU(x) = x \text{ for } x \geq 0; \alpha(e^x - 1) \text{ for } x \leq 0 \tag{7}$$

For negative inputs, ELU provides nonzero outputs and gradients, while positive inputs remain linear. The negative branch can move the mean activation closer to zero and avoids the hard zero-gradient region of ReLU for negative values [19]. No claim is made that ELU alone guarantees better control performance; it is one architectural choice in the reported implementation.

3.4.7. Reward engineering

The reward is engineered around three principles. First, phase-wise decomposition separates pre-hit, hit-moment, and post-hit objectives. Second, asymmetric V3 weights provide a stronger positive

DOI: 10.54254/2755-2721/2026.37819

signal to the left robot while it learns difficult returns; V4 later equalizes the weights. Third, the r_racket_base_clearance penalty has weight −8.0 to discourage self-collision. These choices are heuristic reward shaping decisions [20], and their individual effects require ablation.

3.4.8. Massively parallel training

Training uses Isaac Lab's GPU-parallel simulation. Related GPU-native simulators place physics and policy learning on the same accelerator to reduce data-transfer overhead [21], and massively parallel RL has produced large training-speed improvements in robotic control [22]. Here, 1,024 environments run with a physics time step of 1/120 s; the policy acts every two physics steps (approximately 60 Hz). The maximum 720-step episode corresponds to 12.0 s of simulated time.

PPO performs one policy update after each 32-step rollout from 1,024 environments, yielding 32,768 samples per update. Each update uses five learning epochs and four equal mini-batches of 8,192 samples. The maximum training lengths are 2,000 iterations for V1, 3,000 for V2, 4,000 for V3, and 8,000 for V4.

3.5. Dynamic rally strategy

A rally state machine coordinates alternating receivers, successful contacts, and episode termination (Figure 2). After a valid hit, receiver identity is switched and the new receiver becomes active.

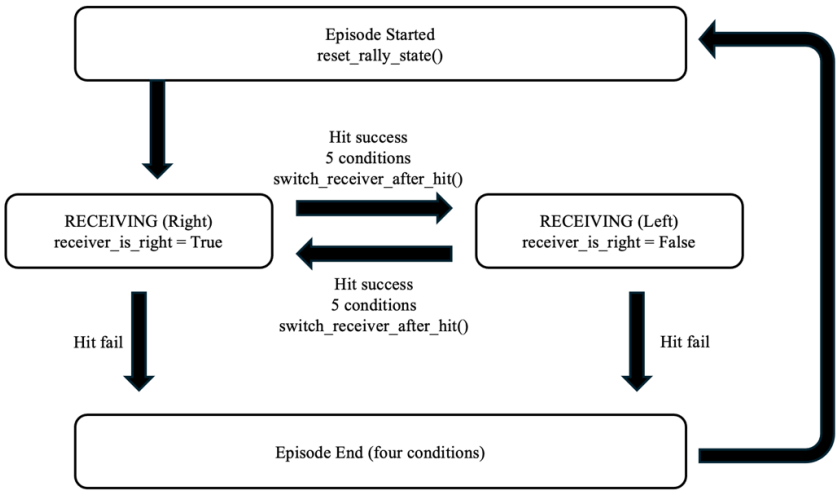

Figure 2. Rally state-machine transitions

The five-condition contact detector described in Section 3.4.4 is evaluated at 120 Hz. If a valid contact is detected, the return is applied and receiver identity is swapped. Otherwise, the episode terminates when one of the following conditions is met:

Four termination conditions are implemented:

(1) rally_miss_timeout: the active receiver does not make a valid hit within 1.75 s;

(2) shuttle_ground: shuttlecock height $z < 0.08$ m;

(3) shuttle_bounds: the shuttlecock leaves $x \in [-4.6, 4.6]$ m, $y \in [-2.6, 2.6]$ m, or $z \in [-0.20, 3.0]$ m;

(4) time_out: total episode duration exceeds 12 s.

DOI: 10.54254/2755-2721/2026.37819

3.6. Progressive curriculum

Starting directly from fully randomized competitive play produces sparse rewards and unstable early learning. The framework therefore follows curriculum learning's easy-to-difficult principle [7] and increases task difficulty with training progress. Four consecutive versions are used: V1 establishes fixed-base contact, V2 adds mobile dual-robot rallies, V3 introduces asymmetric placement, and V4 enables symmetric full-court targets.

3.6.1. V1: fixed-base interception

V1 trains a fixed-base seven-degree-of-freedom Franka arm to intercept one incoming shuttlecock at approximately 7 m/s within a 0.55 s window and return it to the opposite half-court. It establishes four reusable elements for later stages: string-bed contact detection, direction-aware swing guidance, a three-level return-quality reward chain, and common task and physics parameters. The objective is to learn a repeatable single-shot interception behavior before mobility and rally coordination are introduced.

3.6.2. V2: mobile dual-robot rallies

V2 adds two Ridgeback–Franka agents and requires the active receiver to strike while repositioning. Three mechanisms structure the task: a rally state machine that alternates receivers; receiver-gated actions that return the inactive robot toward its standby pose; and phase-specific rewards assigned to the active receiver, with post-hit recovery and direction-specific return-quality terms.

The two agents reuse one Ridgeback–Franka asset. The right-side instance is rotated 180° at its root pose, which avoids maintaining a separate mirrored robot model.

Serve variability follows three phases: no serve noise from steps 0–6,000; a linear ramp from zero to full noise during steps 6,000–36,000; and full noise thereafter. This schedule allows the agents to establish contact timing before confronting the complete serve distribution.

V2 therefore tests whether the V1 interception behavior can be extended to alternating, mobile rallies under progressively randomized serves.

3.6.3. V3: asymmetric tactical training

V3 is a transitional tactical stage. It resumes from the converged V2 checkpoint with compatible observation and action dimensions, preserving the previously learned rally behavior while introducing asymmetric target placement.

V3 introduces asymmetric lateral-target logic. At contact, the left robot samples a sideline target $y \in \pm[0.95, 1.05]$ m, so the opponent must pursue the realized post-hit trajectory; the right robot returns toward the center. This right-defends/left-attacks structure trains lateral placement on the left side and wide-range pursuit on the right side, providing complementary behaviors for V4.

The serve-noise curriculum retains the V2 schedule: fixed serves for steps 0–6,000, a linear noise ramp for steps 6,000–36,000, and full randomization thereafter.

3.6.4. V4: symmetric full-court training

V4 extends the V3 checkpoint to symmetric target placement and full planar base tracking. Both agents can select lateral and longitudinal targets, and curriculum difficulty depends on global training progress as well as the number of hits already completed in the current rally.

DOI: 10.54254/2755-2721/2026.37819

V4 enables both robots to sample lateral and longitudinal targets, replacing V3's asymmetric right-defends/left-attacks design. The mobile base tracks the predicted interception position on both the x- and y-axes. The HitWindowJointPositionAction component computes the base target from the difference between predicted shuttlecock position and the nominal hit position, with separate gains and limits for each axis. During the non-receiving phase, the base returns toward its nominal reset position.

V4 adds predicted_racket_string_bed_tanh and predicted_racket_to_shuttle_tanh rewards. They estimate the shuttlecock position at the nominal hit time with a ballistic predictor and reward advance positioning near that point. These terms complement the existing timed_racket_string_bed_tanh reward, which uses the current shuttlecock position near contact. The predictor is an approximation and does not replace the simulator's aerodynamic model.

The episode limit is increased from 6.0 s in V3 to 12.0 s in V4, allowing longer rallies. The 0.55 s hit window is unchanged. Under the configured simulator, this horizon can accommodate approximately 8–12 contacts rather than the 3–5 contacts typical of the shorter V3 episodes.

V4 uses a two-phase curriculum with four difficulty dimensions: serve noise, lateral target range, longitudinal target range, and within-rally progression.

Phase structure:

• Play phase (steps 0–20,000): difficulty is fixed at 65%, serve noise is fully active, and lateral and longitudinal targets remain at moderate ranges to establish bilateral rally stability.

• Competitive phase (steps 20,000–200,000): all four curriculum components progressively expand toward the full target distribution over 180,000 steps.

Serve-noise curriculum: full longitudinal position and velocity noise is applied because interception has already been trained in V3 and the Play phase. In this configuration, position_noise = (0.25, 0.00, 0) and velocity_noise = (0.35, 0.00, 0).

Lateral-target curriculum: global training progress is multiplied by an in-rally hit-count scale. The global curriculum scale is defined in Equation (8):

$$C_{global}(t) = clamp((t - 20{,}000) / 180{,}000, 0, 1) \quad (8)$$

The in-rally hit-count scale is defined in Equation (9):

$$R_{rally}(h) = clamp((h - h_{min} + 1) / h_{ramp}, 0, 1) \quad (9)$$

where hmin = 2 means that tactical targets are not applied to the first two hits, and hramp = 4 means that four additional hits are required to reach the full within-rally scale. Equation (10) defines the effective difficulty:

$$D_{effective} = C_{global}(t) \times R_{rally}(h) \quad (10)$$

The product schedule keeps extreme targets rare early in training, even in longer rallies. Later, short rallies remain easier while longer rallies can reach the full target range, producing a smooth rather than abrupt increase in difficulty.

The lateral target range is interpolated from a narrow start range to the final range using Equation (11):

$$R_{scaled} = R_{start} + (R_{final} - R_{start}) \times D_{effective} \quad (11)$$

where Rstart = 0.45 m and Rfinal = 2.00 m, with a minimum absolute offset of 1.50 m for sideline targets. Tactical-target probability is 1.0 × Cglobal. A uniform probability gate adds stochastic variation so that not every eligible hit uses a tactical target.

Longitudinal-depth curriculum: V4 adds front/back-court target variation. The right robot samples x in [-4.20, -2.20] m and the left robot samples x in [2.20, 4.20] m. C_global widens the range from a narrow band around the nominal contact point according to Equation (12), and Equation (13) defines the corresponding width increment:

DOI: 10.54254/2755-2721/2026.37819

$x_target \sim Uniform(x_start, x_start + \Delta x)$(12)

$\Delta x = start_half_width + C_global(full_width - start_half_width)$(13)

where $start_half_width = 0.20$ m.

Early competitive training therefore concentrates returns near the nominal contact position; the configured range expands toward the full front-to-back interval as Cglobal approaches 1.0.

Within the configured simulator, V4 is designed to combine the following behaviors:

- Sustained alternating rallies under the 12 s episode horizon;
- Lateral and longitudinal return placement on either half-court;
- A shared policy and symmetric reward structure for both agents;
- Advance base positioning toward a predicted interception point;
- A smooth increase in configured difficulty through the two-phase, product-form curriculum.

Across V1–V4, task complexity increases from fixed-base interception to mobile rallies and tactical target placement. Because contact is applied through a scripted impulse and evaluation is limited to simulation, these behaviors should be understood as components of a simulated training pipeline rather than a complete autonomous competition system.

4. Results

Results are reported for the four sequential curriculum versions. The analysis uses the available training curves and task metrics; it does not include repeated random seeds, confidence intervals, ablations, or a non-curricular baseline. Accordingly, curve stabilization is treated as evidence of learning progress within each configuration, not as proof of statistical superiority.

4.1. V1: fixed-base single-robot training

V1 evaluates whether the fixed-base Franka arm can learn to intercept a shuttlecock following the configured trajectory and return it across the net.

Figure 3 shows that mean reward rises rapidly during approximately the first 100 iterations, fluctuates between roughly iterations 150 and 250, and stabilizes after about 400 iterations near 6.3. Within the V1 reward definition, this pattern is consistent with acquisition of the rewarded interception and return behaviors.

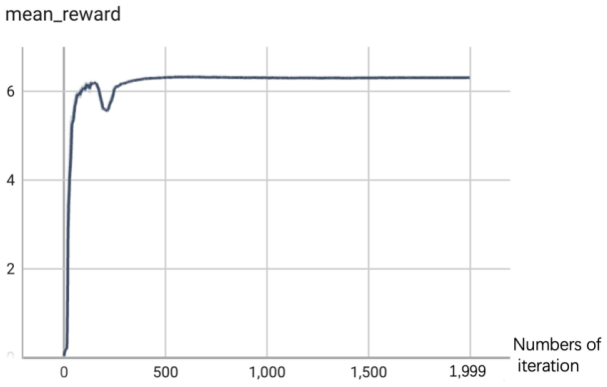

Figure 3. Mean reward during V1 training

DOI: 10.54254/2755-2721/2026.37819

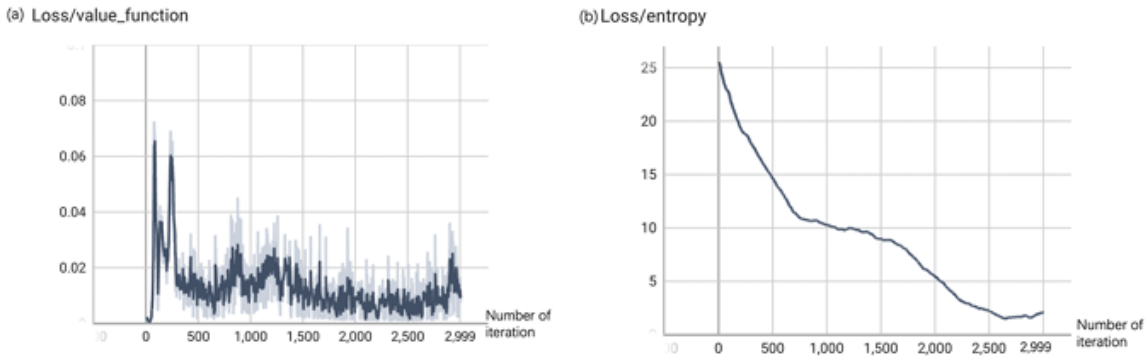

Figure 4. (a) Value-function loss and (b) entropy loss during V1 training

Figure 4 presents entropy loss and value-function loss. Entropy loss decreases over training, indicating reduced policy stochasticity. Value loss falls rapidly during the first 200 iterations and remains near zero on the sampled training rollouts. A small training value loss indicates fit to those targets, but does not by itself establish perfect value prediction or generalization.

4.2. V2: dual mobile-robot rally training

V2 adds a Ridgeback base and a second symmetrically configured robot. Each agent contributes seven arm and two base commands, giving an 18-dimensional joint action vector, of which nine components are receiver-gated at each step. The V2–V4 observation vector contains 81 dimensions, including both robot states, shuttlecock motion, receiver identity, rally count, timing, and the previous action.

V2 retains PPO, increases the maximum training length to 3,000 iterations, and uses 1,024 parallel environments. The stage evaluates mobile interception and alternating rallies under randomized serves.

In Figure 5, mean reward increases during the first approximately 450 iterations, fluctuates until about iteration 1,500, and then stabilizes near 35.5. This is consistent with convergence under the configured reward, although reward alone does not identify which component behaviors improved.

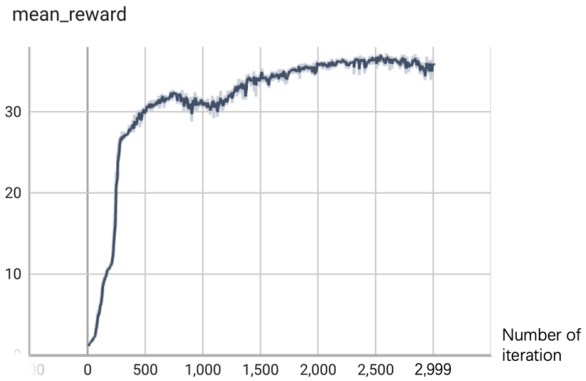

Figure 5. Mean reward during V2 training

Figure 6(a) shows that value-function loss fluctuates during the first approximately 300 iterations and later remains near 0.01. This indicates a small error on sampled training targets; it should not be

DOI: 10.54254/2755-2721/2026.37819

interpreted as a direct measure of real-world accuracy.

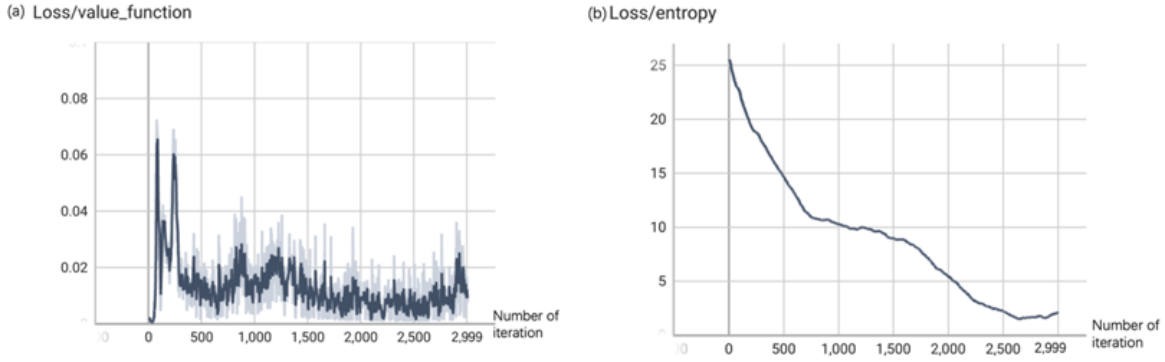

Figure 6. (a) Value-function loss and (b) entropy loss during V2 training

Figure 6(b) shows a substantial early decrease in entropy loss, a plateau from approximately iterations 790–2,200, and later stabilization near 1.6. Relative to V1, the positive terminal value suggests that the learned policy retained more stochasticity in the dual-agent task.

Figure 7(a) shows that successful contacts are absent in the earliest iterations, increase from roughly iterations 100–300, and then stabilize near seven hits per episode. The configured episode horizon limits the maximum observable rally length, so the plateau should be interpreted together with that cap.

Figure 7(b) shows a high and variable shuttle-ground termination rate early in training, followed by a value near zero after approximately 330 iterations. Under the V2 configuration, this is consistent with improved rally continuation.

Figure 7(c) shows a similarly high early miss rate that approaches zero after approximately 300 iterations, suggesting improved contact timing on the training distribution.

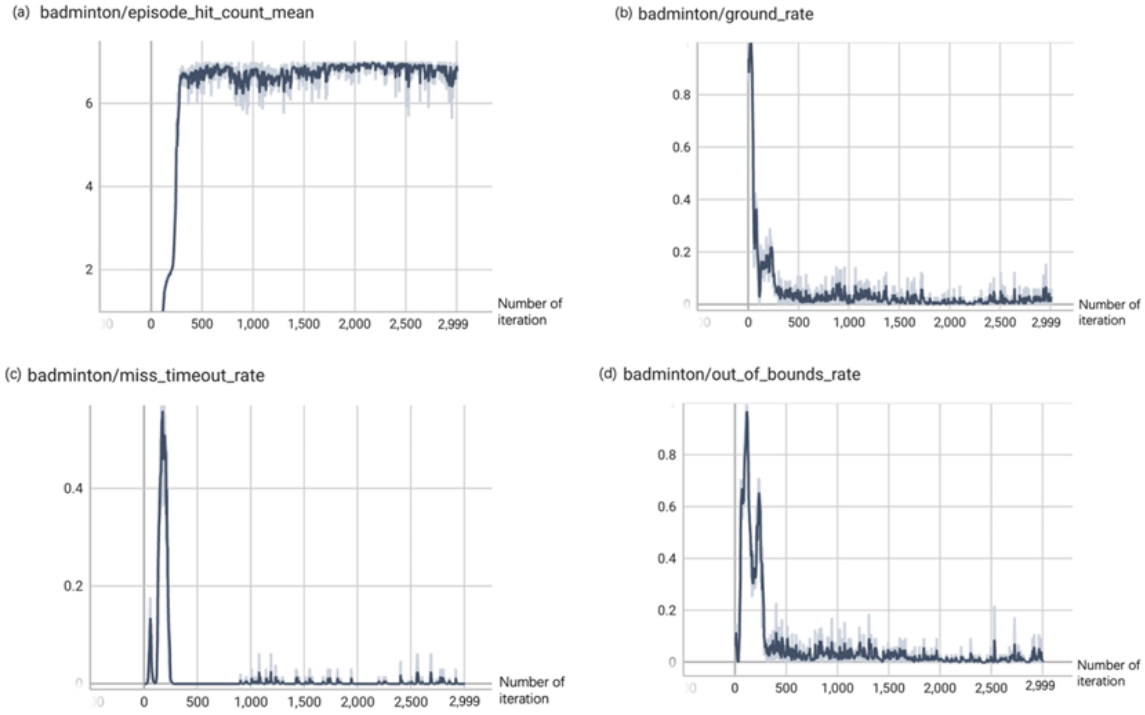

Figure 7. (a) Hits per episode, (b) landing rate, (c) miss rate, and (d) out-of-bounds rate in V2

DOI: 10.54254/2755-2721/2026.37819

Figure 7(d) shows that the out-of-bounds rate also approaches zero. Taken together, the four metrics indicate that V2 learned the rewarded mobile-rally behavior in simulation. They do not separately quantify robustness to unseen serves or modeling error.

4.3. V3: asymmetric tactical training

V3 initializes from V2 and introduces an asymmetric task: the left robot is rewarded for lateral target placement, while the right robot is rewarded for pursuit and return. The stage is intended to prepare separate placement and interception behaviors before the symmetric V4 task.

Figure 8 shows that mean reward increases during the first approximately 1,000 V3 iterations and then stabilizes near 80. The horizontal axis continues from the V2 checkpoint, so V3 does not begin at the origin. The curve is consistent with optimization of the revised asymmetric reward.

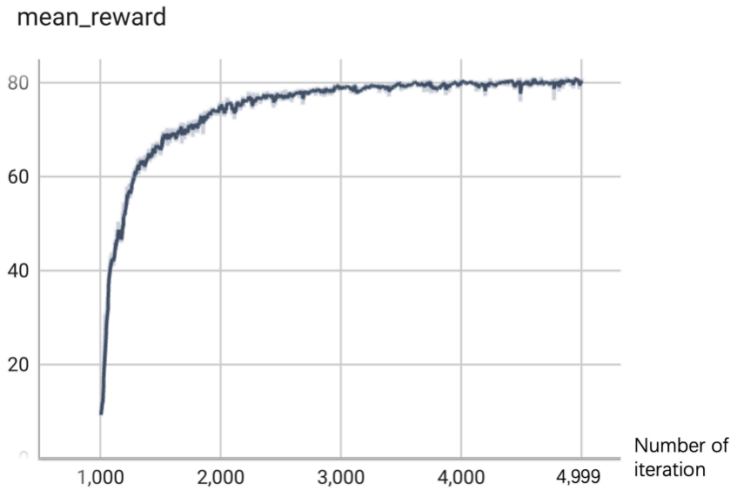

Figure 8. Mean reward during V3 training

In Figure 9(a), entropy loss decreases during the first approximately 3,000 V3 iterations and then stabilizes near 13.6. The curve reflects changing policy stochasticity under the V3 objective, but its absolute value is not, by itself, a measure of tactical diversity.

Figure 9(b) shows that value-function loss is initially variable and later stabilizes near 0.04. This supports stable fitting of sampled V3 returns but does not establish out-of-distribution accuracy.

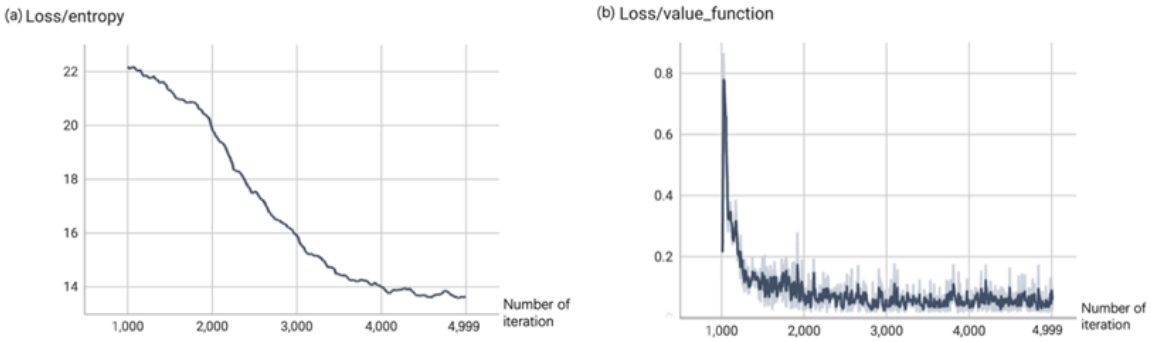

Figure 9. (a) Entropy loss and (b) value-function loss during V3 training

DOI: 10.54254/2755-2721/2026.37819

Figure 10(a) shows an average of approximately seven hits per episode after about 1,000 V3 iterations, consistent with sustained rallies under the configured horizon.

Figure 10(b) shows the landing-related termination rate approaching zero after approximately 1,000 iterations, indicating fewer missed returns on the training distribution.

Figure 10(c) shows that the left robot's lateral-target activation rate stabilizes near 0.44. This confirms frequent use of the programmed tactical target mechanism; it does not independently measure whether each target would defeat a stronger or unseen opponent.

Figures 10(d) and 10(e) report the right robot's lateral pursuit distance and contact success rate. Both rise early and later stabilize, with pursuit distance near 1.2 m and the reported success metric near 1.0. Together with the target-activation curve, these results are consistent with the intended V3 division between left-side placement and right-side interception.

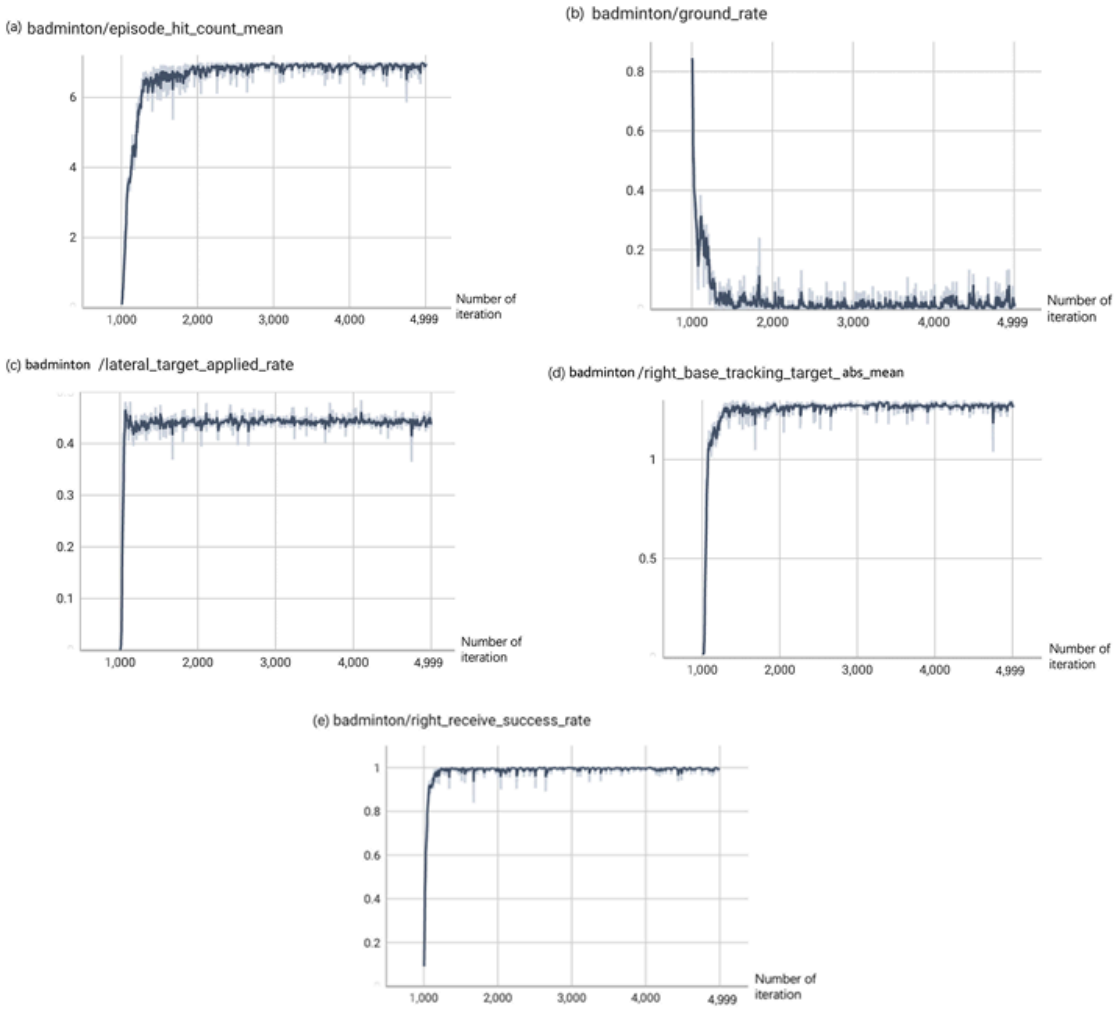

Figure 10. (a) Hits per episode, (b) landing rate, (c) left-side lateral-target probability, (d) right-side lateral pursuit distance, and (e) right-side hit success rate in V3

4.4. V4: symmetric full-court training

V4 initializes from V3, enables tactical targets for both robots, expands base motion to both planar axes, and applies the two-phase composite curriculum. The aim is to examine symmetric target placement and interception rather than to demonstrate complete strategic reasoning.

DOI: 10.54254/2755-2721/2026.37819

Figure 11 shows an early rise in mean reward, followed by a decline until approximately iteration 5,000 and subsequent stabilization near 87. Because curriculum difficulty and reward composition change during V4, the reward is non-stationary and cannot be compared directly with V1–V3. The decline is consistent with a harder task, but an ablation with fixed difficulty would be required to attribute the change specifically to tactical learning.

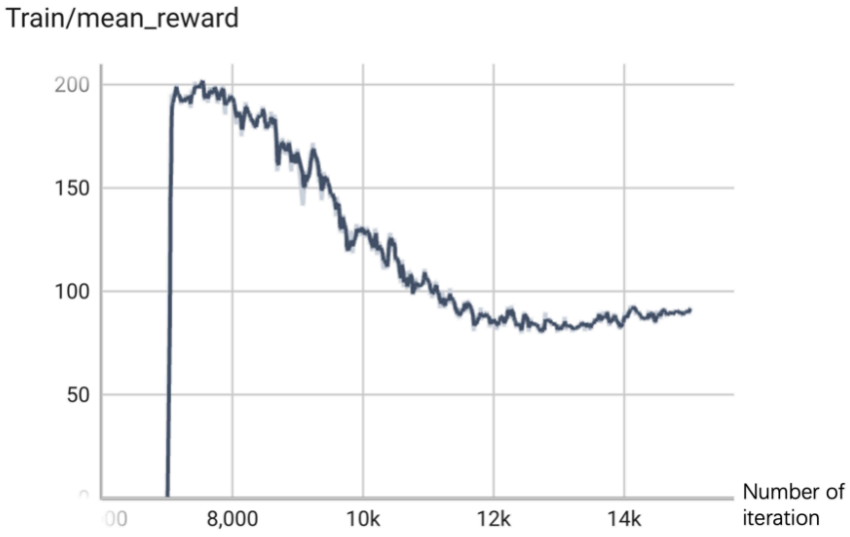

Figure 11. Mean reward during V4 training

Figure 12(a) shows a decrease, intermediate rise, and later decline in entropy loss. The intermediate change coincides with the period in which curriculum difficulty increases and may indicate renewed exploration, but the curve alone cannot identify newly discovered tactics or rule out other causes.

Figure 12(b) shows a related transient increase in value-function loss near iteration 3,000, followed by stabilization around 0.2 after approximately iteration 7,000. The pattern is compatible with a changing return distribution under curriculum and symmetric play.

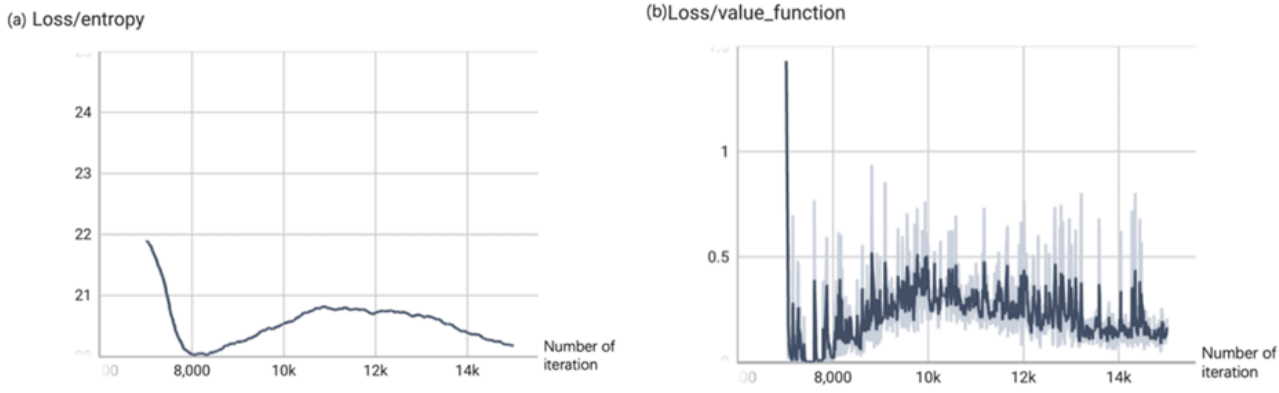

Figure 12. (a) Entropy loss and (b) value-function loss during V4 training

Figure 13(a) shows that the proportion of episodes terminating when the shuttlecock lands increases toward approximately 0.93. This metric indicates the dominant termination mode, but it

DOI: 10.54254/2755-2721/2026.37819

does not specify which robot won, whether the landing was forced, or whether placement quality improved.

Figure 13(b) shows that hits per episode stabilize near five after declining during the curriculum. Shorter rallies may be consistent with more difficult target placement, but they are not sufficient evidence of tactical superiority without winner labels or a controlled opponent.

Figure 13(c) shows that tactical-target activation begins after the initial curriculum period and stabilizes near 0.64. This verifies that tactical targets are increasingly sampled and executed under the V4 schedule. A baseline without tactical rewards and evaluation against unseen opponents are needed to quantify the resulting competitive advantage.

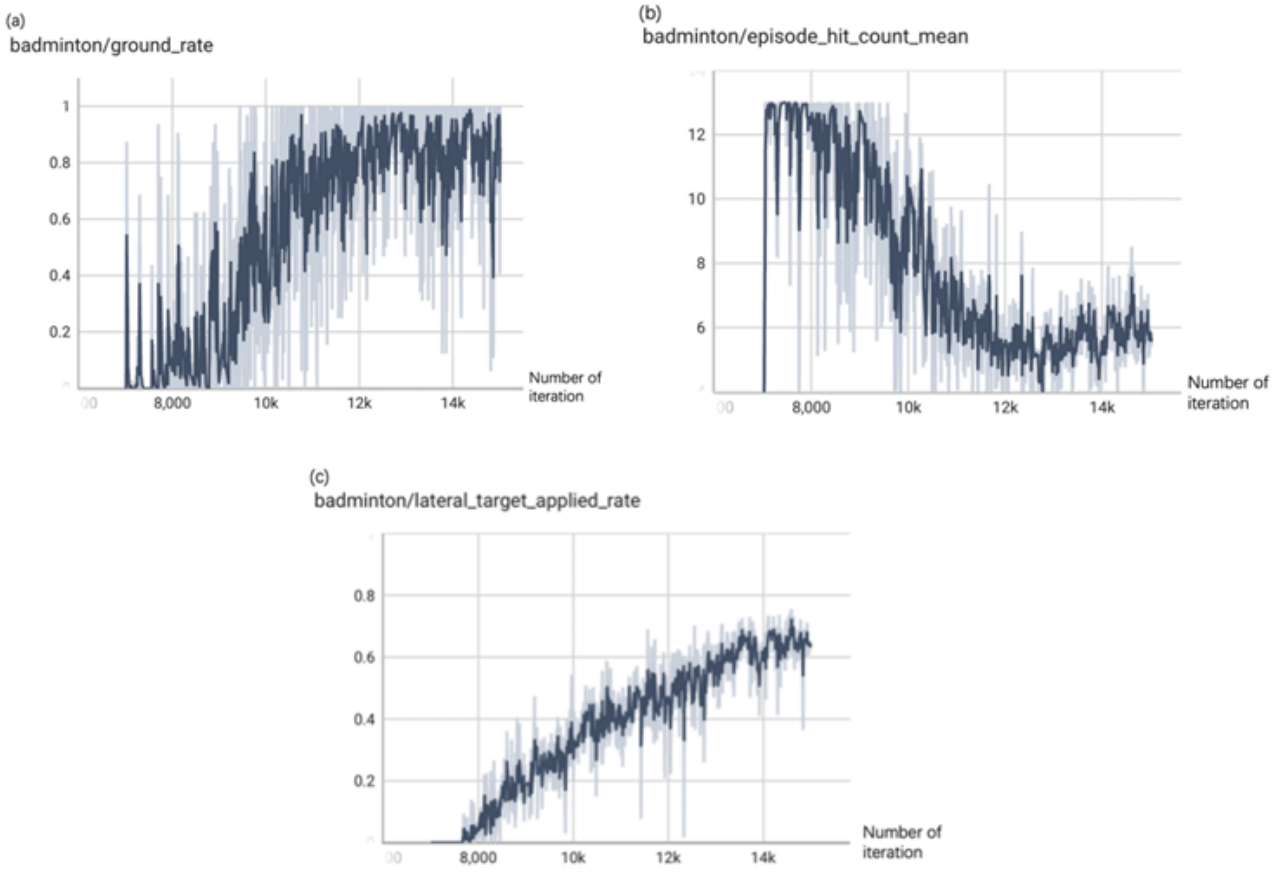

Figure 13. (a) Landing-related termination rate, (b) hits per episode, and (c) tactical-target activation rate during V4 training

5. Discussion

The four-stage curriculum produced distinct learning progressions aligned with the intended task decomposition. V1 established fixed-base interception; V2 added mobile, alternating rallies; V3 separated placement and pursuit through asymmetric rewards; and V4 combined these behaviors in a symmetric target-placement task. This progression is consistent with the motivation for curriculum learning [7], but the current evidence does not isolate the curriculum's causal effect because no random-task or single-stage baseline was trained.

Mirroring the converged right-side policy to initialize the left side reduced the cold-start difficulty of dual-agent training. Receiver gating and the rally state machine further simplified

DOI: 10.54254/2755-2721/2026.37819

coordination by activating one hitting policy at a time. This engineering choice improves tractability, but it also reduces the interaction to an alternating control process rather than unconstrained simultaneous multi-agent play. In V3 and V4, landing-point rewards produced target-placement behavior, not evidence of general tactical awareness. Competitive sports agents can overfit to a single opponent style [11]; therefore, evaluation against independently trained and behaviorally diverse opponents is important.

Several limitations constrain interpretation. First, all experiments were conducted in simulation and used a scripted post-contact velocity update, so impact dynamics and sim-to-real transfer remain unvalidated. Second, the figures report single training traces without repeated seeds, uncertainty intervals, or held-out evaluation conditions. Third, no ablation isolates the contributions of curriculum scheduling, predicted-intercept rewards, receiver gating, or domain randomization. Fourth, the shared V4 policy and symmetric opponent may encourage a narrow equilibrium. Finally, the policy selects target locations but does not distinguish high clears, drops, drives, or smashes.

Future work should first establish reproducible evaluation with multiple seeds, fixed test serves, curriculum ablations, and independently trained opponents. The contact abstraction should then be replaced or calibrated with a racket–shuttle impact model, followed by system identification and hardware experiments on a physical Ridgeback–Franka platform. Dynamics, latency, sensing, and aerodynamic parameters can be randomized more broadly for transfer [15]. A hierarchical policy could separate high-level shot and target selection from low-level interception, while opponent pools or self-play against diverse checkpoints could reduce specialization to one play style [11].

6. Conclusion

This study presents a four-stage curriculum RL framework for simulated badminton rallies with mobile manipulators. The pipeline progresses from fixed-base interception to mobile dual-robot rallies, asymmetric placement, and symmetric full-court target selection. Under the reported configurations, training curves and task metrics show that each stage acquired behaviors associated with its rewards and curriculum schedule. The main contribution is therefore a structured simulation and training pipeline that decomposes a sparse, dynamic contact task into increasingly difficult stages.

The results do not yet demonstrate statistically robust superiority or real-world competitive play. Evaluation is limited to descriptive curves from simulation, without repeated seeds, baselines, ablations, unseen opponents, or physical deployment. In addition, the scripted contact impulse omits part of the racket–shuttle interaction, and tactical control is limited to landing-point selection. The next steps are controlled multi-seed evaluation, ablation of the curriculum and predicted-intercept terms, opponent-diverse self-play, richer shot-type control, calibrated shuttlecock and impact dynamics, and staged transfer to physical hardware. These experiments are necessary to determine whether the observed simulation behaviors generalize beyond the present training distribution.

References

[1] Kwon, W., Baek, S., Baek, J., Shin, W., Gwak, M., Park, P., et al. (2025). Reinforced intelligence through active interaction in real world: A survey on embodied AI. *International Journal of Control, Automation and Systems, 23*(6), 1597–1612. https://doi.org/10.1007/s12555-025-0127-1

[2] Cohen, C., Darbois Texier, B., Quéré, D., & Clanet, C. (2015). The physics of badminton. *New Journal of Physics, 17*(6), 063001. https://doi.org/10.1088/1367-2630/17/6/063001

[3] Chen, L. M., Pan, Y. H., & Chen, Y. J. (2009). A study of shuttlecock's trajectory in badminton. *Journal of Sports Science and Medicine, 8*(4), 657–662.

DOI: 10.54254/2755-2721/2026.37819

[4] Ma, Y., Cramariuc, A., Farshidian, F., & Hutter, M. (2025). Learning coordinated badminton skills for legged manipulators. *Science Robotics, 10*(102), eadu3922. https://doi.org/10.1126/scirobotics.adu3922
[5] Portela, T., Margolis, G. B., Ji, Y., & Agrawal, P. (2024). Learning force control for legged manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 15366–15372). IEEE. https://doi.org/10.1109/ICRA57147.2024.10611066
[6] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv*. https://doi.org/10.48550/arXiv.1707.06347
[7] Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 41–48). ACM. https://doi.org/10.1145/1553374.1553380
[8] Liu, C., Jiang, L., Wang, Y., Yao, K., Fu, J., & Ren, X. (2025). Humanoid whole-body badminton via multi-stage reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.2511.11218
[9] Wang, Z., Jia, Y., Shi, L., Wang, H., Zhao, H., Li, X., et al. (2024). Arm-constrained curriculum learning for loco-manipulation of a wheel-legged robot. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 10770–10776). IEEE. https://doi.org/10.1109/IROS58592.2024.10802062
[10] Zhalehmehrabi, A. (2024). *Robotic arm control via curriculum deep reinforcement learning* [master's thesis]. Politecnico di Milano.
[11] Orsula, A. (2024). Learning to play air hockey with model-based deep reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.2406.00518
[12] Pinto, L., Andrychowicz, M., Welinder, P., Zaremba, W., & Abbeel, P. (2018). Asymmetric actor critic for image-based robot learning. In *Proceedings of Robotics: Science and Systems XIV*. https://doi.org/10.15607/RSS.2018.XIV.008
[13] Mittal, M., Liu, J., Yu, C., Yu, Q., Rudin, N., Hoeller, D., et al. (2023). Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters, 8*(6), 3740–3747. https://doi.org/10.1109/LRA.2023.3270034
[14] Badminton World Federation. (2025). Statutes, Section 4.1: Laws of badminton. *BWF*. https://corporate.bwfbadminton.com/statutes/
[15] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 23–30). IEEE. https://doi.org/10.1109/IROS.2017.8202133
[16] Haddadin, S. (2024). The Franka Emika robot: A standard platform in robotics research [survey]. *IEEE Robotics and Automation Magazine, 31*(4), 136–148. https://doi.org/10.1109/MRA.2024.3451788
[17] Clearpath Robotics. (n.d.). Ridgeback indoor omnidirectional platform. *Clearpath Robotics*. https://clearpathrobotics.com/ridgeback-indoor-robot-platform/
[18] Schulman, J., Moritz, P., Levine, S., Jordan, M. I., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv*. https://doi.org/10.48550/arXiv.1506.02438
[19] Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (ELUs). *arXiv*. arXiv:1511.07289. doi:10.48550/arXiv.1511.07289.
[20] Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning* (pp. 278–287). Morgan Kaufmann.
[21] Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., et al. (2021). Isaac Gym: High performance GPU-based physics simulation for robot learning. *arXiv*. arXiv:2108.10470. doi:10.48550/arXiv.2108.10470.
[22] Rudin, N., Hoeller, D., Reist, P., & Hutter, M. (2022). Learning to walk in minutes using massively parallel deep reinforcement learning. *Proceedings of Machine Learning Research, 164*, 91–100.