MOTION PLANNING AND CONTROL

Learning coordinated badminton skills for legged manipulators

Yuntao Ma*, Andrei Cramariuc, Farbod Farshidian†, Marco Hutter

Coordinating the motion between lower and upper limbs and aligning limb control with perception are substantial challenges in robotics, particularly in dynamic environments. To this end, we introduce an approach for enabling legged mobile manipulators to play badminton, a task that requires precise coordination of perception, locomotion, and arm swinging. We propose a unified reinforcement learning–based control policy for whole-body visuomotor skills involving all degrees of freedom to achieve effective shuttlecock tracking and striking. This policy is informed by a perception noise model that uses real-world camera data, allowing for consistent perception error levels between simulation and deployment and encouraging learned active perception behaviors. Our method includes a shuttlecock prediction model and constrained reinforcement learning for robust motion control to enhance deployment readiness. Extensive experimental results in a variety of environments validate the robot's capability to predict shuttlecock trajectories, navigate the service area effectively, and execute precise strikes against human players, demonstrating the feasibility of using legged mobile manipulators in complex and dynamic sports scenarios.

INTRODUCTION

Human sport competitions like badminton pose substantial challenges for players because of the complex interplay required between footwork and upper limb movements. Competent players must develop advanced loco-manipulation skills to effectively cover the extensive court area, complemented by precise hand-eye coordination to anticipate and correctly hit the shuttlecock. Players must predict the shuttlecock's most likely incoming trajectory to synchronize the timing, location, angle, and velocity for strikes that achieve a successful return.

Complex interplay among perception, locomotion, and manipulation makes such sports applications a formidable challenge to developing advanced unified skills for legged mobile manipulation systems given deficiencies in current paradigms for controllers and hardware. The main challenge for such robots involves balancing rapid and responsive locomotion with accurate arm movements. Although a robot's high number of degrees of freedom (DoFs) in principle allows for agile movements, realizing such potential in practice depends heavily on the control algorithms.

This balancing act is further complicated by the limitations of commercial onboard camera systems, which often must compromise among frame rate, angular resolution, field of view (FOV), and transmission delay, in contrast with human eyes, which have much better motion stabilization, adjustable focus, and more sophisticated information processing. To attain similar performance, digital cameras on robots require a perception-aware controller that moves the camera using smooth motions while keeping the target in the FOV.

Various methods have been used for athletic robot control in prior works. Model-based control methods have been used for executing complex maneuvers such as front/backflips (*1*–*3*) and throwing objects (*4*, *5*). These techniques either require model simplifications or only allow local feedback control around preoptimized trajectories. Recently, noteworthy progress in reinforcement learning (RL)–based robot running (*6*–*9*) and parkour (*10*–*13*) has demonstrated the potential of learning-based methods to advance robot agility. However, these developments are primarily restricted to locomotion in static scenes. Moreover, the exploration of pedipulation—namely, using robot feet as manipulators—through RL has uncovered promising approaches to enhance robots' applicability in interactive sports (*14*, *15*). Despite its potential, this method is generally characterized by a limited operational reach, potentially curtailing its utility in activities where extensive spatial interaction is essential.

In terms of application, table tennis has been extensively researched for both accuracy (*16*–*18*) and strategy (*16*, *19*), primarily using fixed-base or gantry manipulators with external vision systems. In contrast, our work emphasizes whole-body visuomotor skills and relies solely on onboard perception, integrating both legged locomotion and arm swinging—an approach that more closely mirrors human gameplay.

To date, the integration of locomotion and manipulation within the realm of legged manipulation controllers has often seen these tasks treated as distinct entities (*20*–*26*). This decoupled approach simplifies the optimization process; however, it imposes substantial constraints on the robot's range of motion and its agility, which are critical factors in dynamic environments. Several recent studies have deviated from this traditional decoupling paradigm, albeit with modifications that either maintain a slow-moving state (*27*) or focus solely on self-manipulation tasks (*28*), which does not fully exploit the manipulation capabilities of legged robots in more dynamic settings. Conversely, there has been noteworthy progress in achieving dynamic motions through imitation learning. Techniques developed for simulated agents playing tennis (*29*) or humanoid robots engaging in generic imitative behaviors (*30*, *31*) showcased the potential of using demonstration data to inform robotic control systems while requiring demonstrations from agents of similar morphology. A recent approach involving task-space imitation learning on quadrupedal mobile manipulator platforms demonstrated dynamic manipulation capabilities with the arms and grippers while not fully leveraging the locomotive potential of the legs in hardware deployments (*32*).

Another major challenge is the trade-off between active perception behavior and agile motion control. Privileged learning has been

Robotic Systems Lab, ETH Zurich, 8092 Zurich, Switzerland.
*Corresponding author. Email: mayuntao94@gmail.com
†Present address: Robotics and AI Institute, 145 Broadway, Cambridge, MA, USA.

used in scenarios where a student policy reconstructs privileged teacher observations on the basis of past interactions and perception history (*33*–*35*). In this framework, the teacher policy observes privileged information and is not incentivized to learn active perception behaviors; namely, visiting trajectories are informative for decoding the privileged observations. This results in an information gap between the teacher and student policies. Some recent works incorporating perception directly into the RL training loop represent an advancement. For instance, neural rendering techniques use a learned perception model where the latent representations can be efficiently integrated within the training process (*10*, *36*). This approach has shown efficacy in structured and static scenes, enabling the potential learning of active exploration behaviors. Furthermore, the emergence of active perception behaviors has been documented in recent research (*37*, *38*). However, the emergent active perception behavior was not quantitatively evaluated in these works.

Additional practical deployment challenges include the constraints on electrical current supply to robot actuators and the need to mitigate perception and communication delays. These elements collectively pose barriers to developing robots that can play badminton at a level comparable to that of human players.

To address the complex control challenges encountered in real-scale badminton games, we developed a unified RL-based controller trained in simulation that controls both the base locomotion and the arm manipulative actions. Extending a previous work (*39*), this integrated approach leverages all DoFs of the robot to track target end-effector (EE) states at specified points in time, enabling effective responses to the shuttlecock's incoming trajectory. To prepare the control policy for consecutive shuttlecock hits and learn postswing follow-through behaviors, we implemented multiple swing targets that were 2 s apart per learning episode. To ensure that the Markov decision process (MDP) was well posed for the critic to estimate the value function, we used an asymmetric actor-critic formulation (*40*). We incorporated a parameterized shuttlecock perception model on the basis of real-world camera data in the training loop. This model captured the effect of robot motion on perception quality by accounting for both single-frame object-tracking errors and final interception predictions, which reduced the perception sim-to-real gap and allowed the robot to learn perception-driven behaviors that were also effective on the hardware. This perception model enabled us to reward the policy on the basis of the final perception error and avoided the need for hard-coded orientation strategies, preserving motion efficiency. As an example, the robot may pitch up to keep the shuttlecock in the camera FOV until it needs to pitch down again to swing the racket. The RL algorithm balances the trade-off between agile control and accurate shuttlecock perception by optimizing the policy's overall ability to hit the shuttlecock in simulation. Furthermore, the system integrates shuttlecock prediction (*41*), constrained RL (*42*), state estimation (*43*, *44*), and system identification on the manipulator dynamics (*45*, *46*) to facilitate the hardware deployment of the trained control policy.

Here, we demonstrate a quadrupedal mobile manipulator that autonomously plays badminton with human opponents using only onboard perception. Through an integrated RL approach that coordinates whole-body motion with perception, the robot adapts its gait patterns on the basis of time and distance constraints to track and intercept shuttlecocks with swing velocities up to 12.06 m s^{-1}. Extensive hardware and simulation experiments validate the system's ability to maintain long rallies in collaborative matches with humans, showcasing emergent active perception behaviors and consistent shuttlecock interception within the court, all while balancing stability with agile arm swings.

RESULTS

Movie 1 summarizes the results of the presented work. Our experiments demonstrated the robot's ability to autonomously track, intercept, and return shuttlecocks during gameplay with human opponents. The system successfully coordinated whole-body movements, adapting its posture and gait patterns on the basis of the shuttlecock's trajectory while maintaining effective visual tracking. We evaluated the robot's performance in aspects such as interception success rates, swing velocity tracking, active perception capabilities, and adaptive locomotion strategies.

System overview

The quadrupedal mobile manipulator robot used in this work consisted of the ANYmal-D base (*47*) and the DynaArm. The robot was equipped with a ZED X stereo camera with global shutters for shuttlecock perception (Fig. 1A). The badminton racket was oriented at a 45° angle with respect to the wrist joint, which proved to be the most effective configuration on the basis of early simulation tests of various orientations.

For deployment, the robot's state estimation operated at a frequency of 400 Hz, and the robot control policy updated observations and sent joint position commands at a rate of 100 Hz, as illustrated in Fig. 1. The system's perception included shuttle position measurement, state estimation, and trajectory prediction. It ran asynchronously at 60 Hz on a Jetson AGX Orin module. Further details are available in the Materials and Methods section.

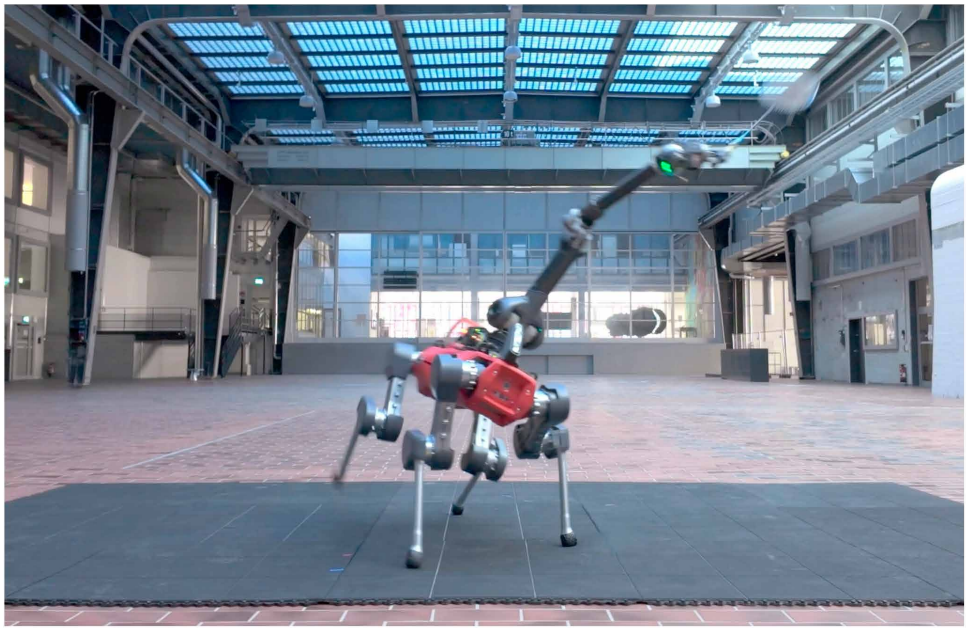
Collaborative game with human players

Collaborative games were held between the robot and amateur players to validate the system's capability to play real-scale badminton and maintain long rallies, as shown in Movie 1. Throughout the games, ANYmal was able to respond appropriately to the incoming shuttle with various velocity and landing positions, albeit with some failures to return them. The perception module took on average 0.357 s after the opponent had hit the shuttlecock to register trajectories for interception. This left on average 0.654 s until the shuttle trajectory crossed the target interception height of 1.25 m above the robot base height. The fastest hit from the policy was 0.367 s after the interception position was computed.

The robot was capable of consecutive hits. Multiple rallies are documented in Movie 1, with a streak of 10 shots in a single rally. The policy also demonstrated the emergent behavior of moving back near the center of the court after each hit, similar to how human players prepare for the next hit.

Badminton motor skills

To evaluate the effectiveness of our control policy, we measured the overall success rate of the robot in hitting shuttlecocks that would land at various distances. Figure 2A indicates the simulated and hardware success rates in different parts of the badminton court, with the robot always starting at the center. The shuttlecock trajectories in this evaluation followed the same initial velocity, and the initial positions were shifted such that the landing locations were distributed as in the figure. We assessed in simulation the policy's ability to



Movie 1. Summary of the results and the method. The video demonstrates our approach for enabling a legged mobile manipulator robot to play badminton through coordinated control. It showcases how our training pipeline produced policies that balance mobility requirements with rapid arm movements for successful shuttlecock returns. The video also highlights our active perception framework, which incorporated real-world camera noise models into reinforcement learning to develop perception-aware behaviors. This allowed the robot to track fast-moving shuttlecocks while traversing the court for interception and return shots. Various experiments with human players illustrated the system's capabilities across different gameplay scenarios.

intercept the shuttlecock with increasing levels of difficulty, each level corresponding to a qualitative improvement of the badminton skill.

Level I—position tracking

Given ground-truth perception, we evaluated the percentage of hits that reached the interception position within 0.1 m—the approximate distance between the racket's center and its edge—at the commanded swing time. In the service area, the simulated results indicated that the robot could intercept the incoming shuttlecock with a negligible failure rate.

Level II—perception error

To assess the robot's ability to predict the shuttle trajectory and intercept it successfully, we added the perception error to the success criteria for this level of evaluation. The perception error measures the position difference between the ground-truth shuttlecock position in the simulation and the extended Kalman filter (EKF)–estimated position at the time of the swing. To achieve a small error, the control policy had to command the robot to maintain sight on the shuttlecock for a substantial duration based on the measurement noise model while accurately tracking the racket state at the commanded point in time. This level represents the robot's ability to predict and intercept the shuttlecock successfully. This task became particularly challenging at the borders of the service court and when the shuttlecock landed directly behind the robot, as indicated by the lower success rate in Fig. 2A in these regions. We attributed this difficulty to the robot's rectangular FOV, which synergized well with base tilting maneuvers (shown in Fig. 3) to extend visual tracking of the shuttlecock. However, when the shuttlecock approached from directly overhead or behind, the robot had to pitch directly upward, making it substantially more challenging to maintain continuous visual contact.

In addition, we reported the EE velocity tracking error at interception when the robot attempted to hit shuttlecocks landing farther from the court center along lateral, longitudinal (front-back), and diagonal directions. The velocity tracking accuracy degraded when shuttlecock landings occurred beyond 2.5 m from the court center in the lateral and diagonal directions and beyond 2.0 m in the longitudinal direction.

We validated the success rate evaluation on the hardware by examining how effectively the robot could hit the shuttlecock back over the net. The hardware evaluation was conducted with the robot facing the net while intercepting shuttlecocks approaching from both lateral and frontal directions. This validation confirmed the effectiveness of our deployed control method. The robot maintained stability and avoided the arm current consumption constraint throughout the hardware experiments, demonstrating its robustness. Video documentation of this evaluation is available in movie S1.

Fast and accurate racket swing

We further assessed the system's ability to track swing velocity and position commands on hardware with the setup shown in movie S2. The robot was commanded to swing at varying target velocities to reach position targets at the center of the court at a height of 1.3 m above the starting base height. In Fig. 2B, the executed EE velocity and maximum base angular velocity were plotted against the commanded velocities. The executed swing velocity generally tracked the commanded velocity below 10 m s^{-1}, with diminishing accuracy at higher velocities. The robot achieved a peak executed velocity of 12.06 m s^{-1} when commanded to swing at 19 m s^{-1}. By comparison, amateur badminton players can reach swing velocities between 20 and 30 m s^{-1}, and a recent study on robot table tennis (*16*) reported an average swing velocity of 6.83 m s^{-1} for the fastest low-level skill. As detailed in the Materials and Methods section, the system operated near its current and joint velocity limits to achieve these commands. In addition, higher commanded velocities led to increased base angular velocities, indicating a coupling between base attitude control and manipulator swing.

Figure 2C shows the distance between the racket and the target position around the impact time, with the racket reaching its closest point precisely at the commanded impact moment. At the commanded swing of 12 m s^{-1}, the robot executed swings of mean 10.8 m s^{-1}, with a mean position error of 0.117 m, which, in terms of timing, was equivalent to a mere offset of 0.0108 s as the racket moved at the target velocity.

Active perception behavior

The coordinated interplay between perception and movement was essential for the robot to successfully track and respond to the shuttlecock during gameplay. This section discusses the policy's emergent

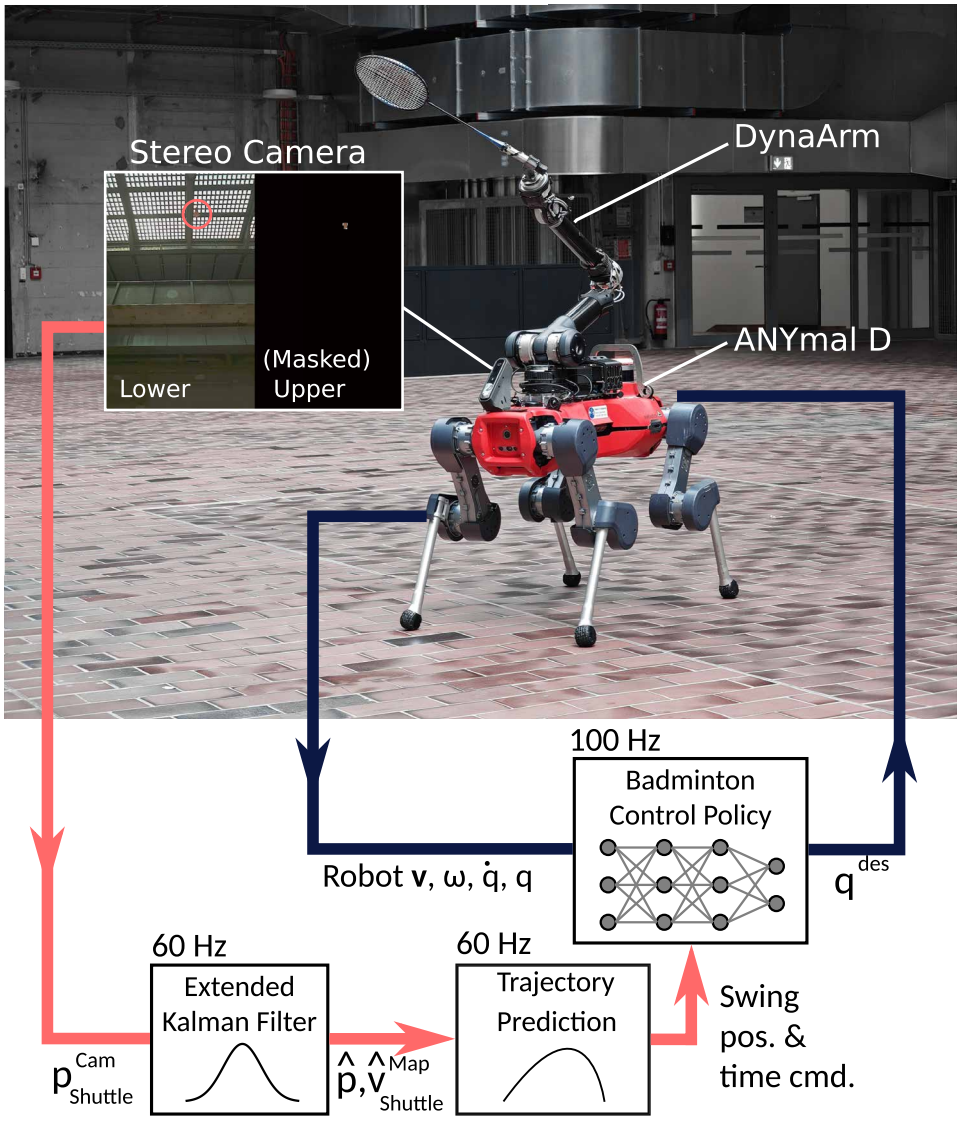


Fig. 1. System overview. The legged mobile manipulator consists of a quadrupedal base and a dynamic arm. We additionally mounted a stereo camera with global shutters. The robot controller system receives the shuttle position computed in the camera frame, predicts the interception position, and feeds it to the RL policy along with the robot proprioception observations. The policy controls all 18 robot drives by producing joint commands.

active perception behavior, evaluates its influences, and compares it with baseline methods. We evaluated the effectiveness of the proposed active perception training and compared our method with two baselines with diverse incoming shuttle trajectories in simulation with the perception noise model. The first baseline, which we call the FOV-reward policy, was trained with an explicit reward for keeping the shuttlecock within the FOV, with this reward term tuned to a scale similar to our proposed perception error reward for meaningful comparisons. The second baseline, the no-perception-behavior policy, observed the ground-truth shuttlecock states and hence the ground-truth interception position and therefore was not incentivized to learn perception-driven behaviors.

The perception error in this section refers to the mean $L2$ error between the true shuttlecock position and the estimated position from the EKF at the time of the desired racket swing based on the measurement noise model. Overall, the perception error for all three methods depended on the landing location of the shuttlecock (Fig. 4A), with larger errors occurring near the edge of the badminton service area. The baseline policy trained with ground-truth perception demonstrated no active perception behavior, resulting in substantially larger errors than the other methods.

We measured the mean and SD of perception errors by regions (a, b, and c) of the badminton court (Fig. 4B). Both the proposed method and the policy trained with the FOV reward notably outperformed the no-perception-behavior policy in region c, where the shuttlecock was more likely to exit the robot's FOV. The similarity in perception errors between our proposed method and the FOV-reward baseline was due to the scaling of the FOV reward, which

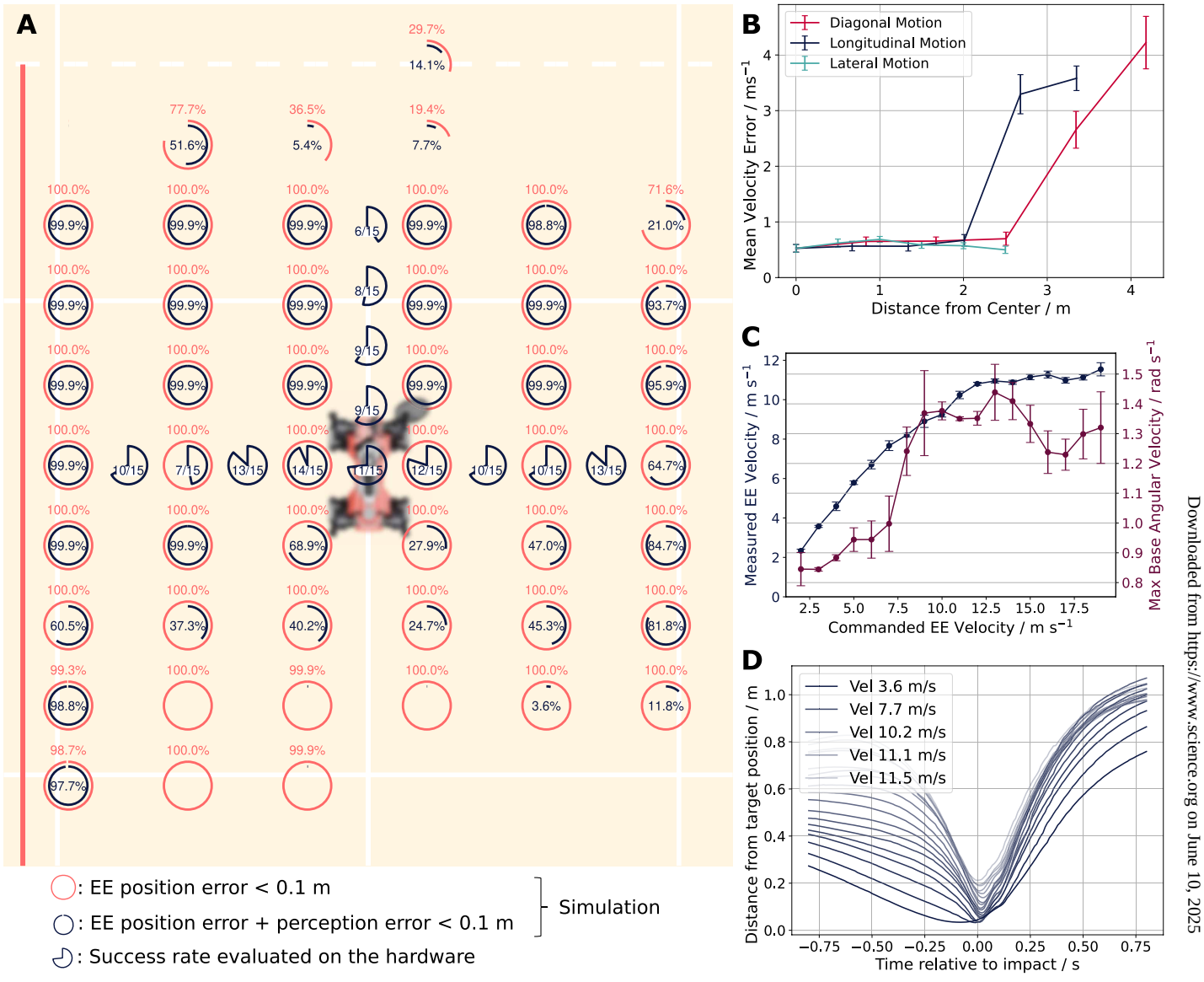


Fig. 2. Assessing the controlled racket swings. (**A**) We quantitatively evaluated the control system's success rate with progressively more difficult conditions in simulation and partly on the hardware. The EE position error condition evaluates the robot's ability to reach EE targets across the court; the additional perception error condition assesses the robot's ability to perform coordinated perception and control. On the hardware, we evaluated the robot's ability to hit the shuttlecock back across the net at intervals of 0.5 m. The target and robot's initial positions are drawn to their corresponding positions on the badminton court. (**B**) The EE velocity tracking error at interception increases as the robot attempts to hit shuttlecocks landing farther from the court center. The error bars represent the SD of the error across 6600 trials. (**C**) The EE swing velocity and maximum base angular velocity are plotted against the commanded EE velocity, with the error bar spanning from the minimum to maximum values. The robot is able to reach a maximum executed swing velocity of 12.06 ms^{-1}, with generally increased base angular velocities at higher EE velocity commands. (**D**) Distance between the racket sweet spot and the target impact position over relative time. The EE minimizes this distance precisely at the commanded impact time. The evaluations in (C) and (D) were conducted on the hardware.

was adjusted to ensure that the policy still performed reasonable locomotion and racket swings. Although we could have increased the FOV reward scaling arbitrarily to further reduce its perception error, doing so would have led to an unfair comparison because it would have overemphasized FOV tracking at the cost of other important behaviors. Nonetheless, the similar perception error between the two methods suggested that our approach could achieve active perception without relying on explicit FOV rewards.

Another comparison we made was concerning normalized mechanical power, which we define as

$$\varphi = \sum_{\text{alljoints}} \left[\tau \dot{q}\right]^{+} / d \tag{1}$$

where τ denotes the joint torque, $\dot{q}$ is the joint velocity, and d is the target distance. This metric provided an indication of the policies' energy

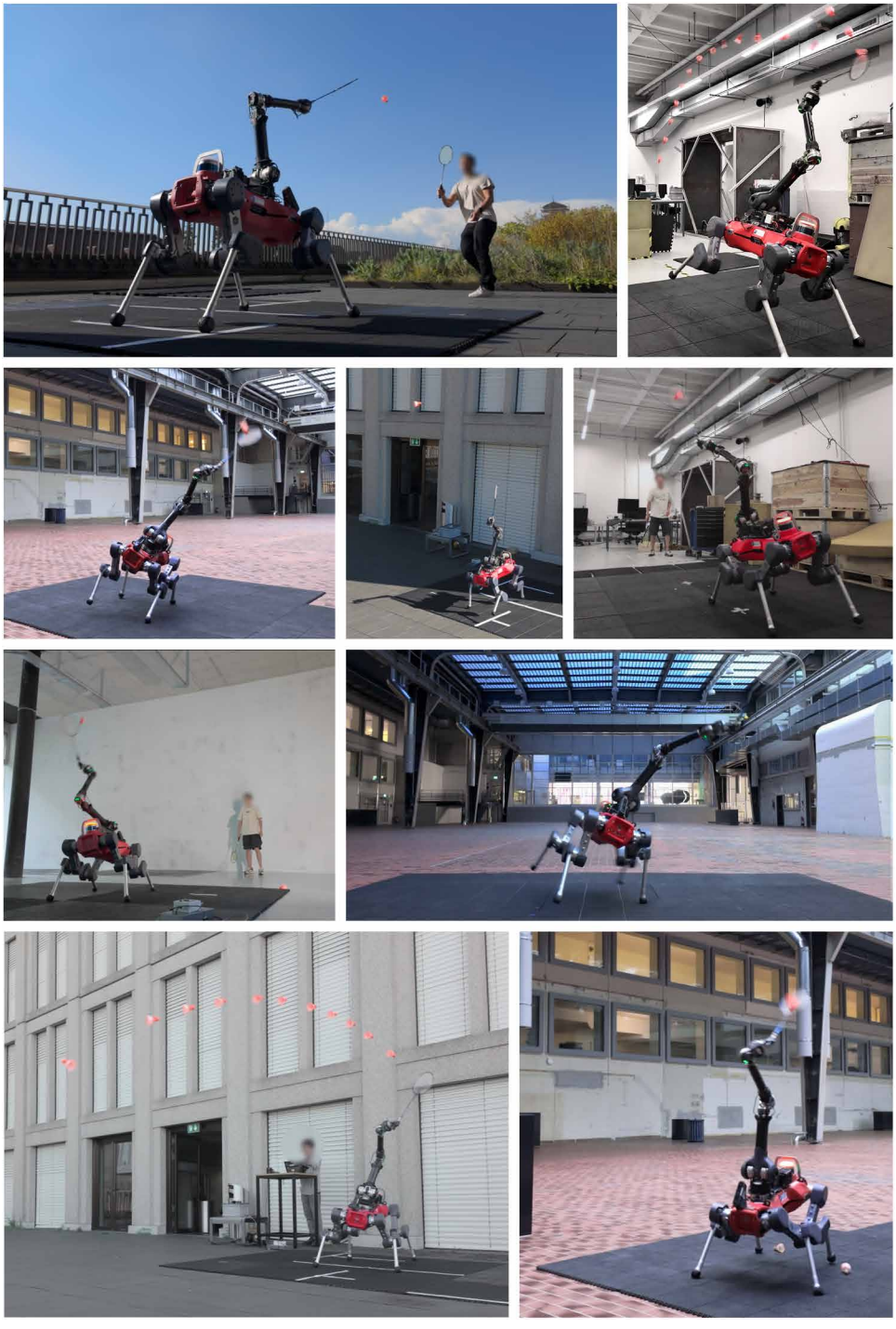


Fig. 3. Deployment of the badminton control policy on our legged manipulator. Our system operates entirely on onboard perception and computation for shuttlecock detection, trajectory prediction, and limb control. It has been tested in various environments, including the lab, a historic machine hall, and outdoor settings.

efficiency. On this scale, our method performed comparably to the no-perception-behavior baseline, with both outperforming the FOV-reward baseline (Fig. 4C). Important to note is that the no-perception-behavior baseline represented an upper bound of the mechanical power performance, given that it solved a subset of the tasks of our proposed method in this comparison. The previously presented metrics indicated that our method balanced between active perception and efficient movement, optimizing both mechanical power and EE tracking.

Figure 4 (D and E) illustrates an instance of the learned active perception behavior observed in our system. The robot started in a stationary position (Fig. 4D, i). Once the interception target was registered (Fig. 4D, ii), the robot first pitched down while keeping the shuttlecock in the upper part of the FOV. Then, it pitched up (Fig. 4D, iii) to reduce the shuttle angular velocity with respect to the camera frame, thus reducing motion blur and keeping the shuttlecock in sight for longer. As soon as the shuttlecock exited the FOV, the robot pitched down again (Fig. 4D, iv) to adjust the robot posture for the racket swing. In this instance, the active perception behavior led to 0.10 s of additional sight of the shuttle flight.

Gait adaptation

Gait adaptation played a critical role in the robot's ability to intercept and return the shuttlecock effectively under varying distances and time constraints. This section discusses the robot's gait patterns in response to different task conditions, as illustrated in Fig. 5. The figure showcases some of the emergent adaptive behavior, with additional comparisons available in the Supplementary Materials and the full video documentation in movie S3.

We first examined the relationship between gait and the distance the robot needed to cover to intercept the shuttlecock. For this, we conducted hardware experiments where we swept across increasing distances with a fixed time to reach the target (1.6 s) while keeping track of the foot contacts.

At short distances of 0.5 m, no locomotion was necessary. The robot slightly lifted its left front (LF), right front (RF), and right hind (RH) legs to reorient the base while keeping the left hind (LH) in contact with the ground, focusing on precise positioning of the EE for the swing. By the time of the swing, all feet were in contact with the ground (Fig. 5B).

At medium distances of 1.5 m, the robot moved with irregular gait patterns, engaging all four legs in swing phases. The right-side legs, being farther from the target, had notably longer air time than the left. At the time of the swing, three feet remained in contact with the ground (Fig. 5C).

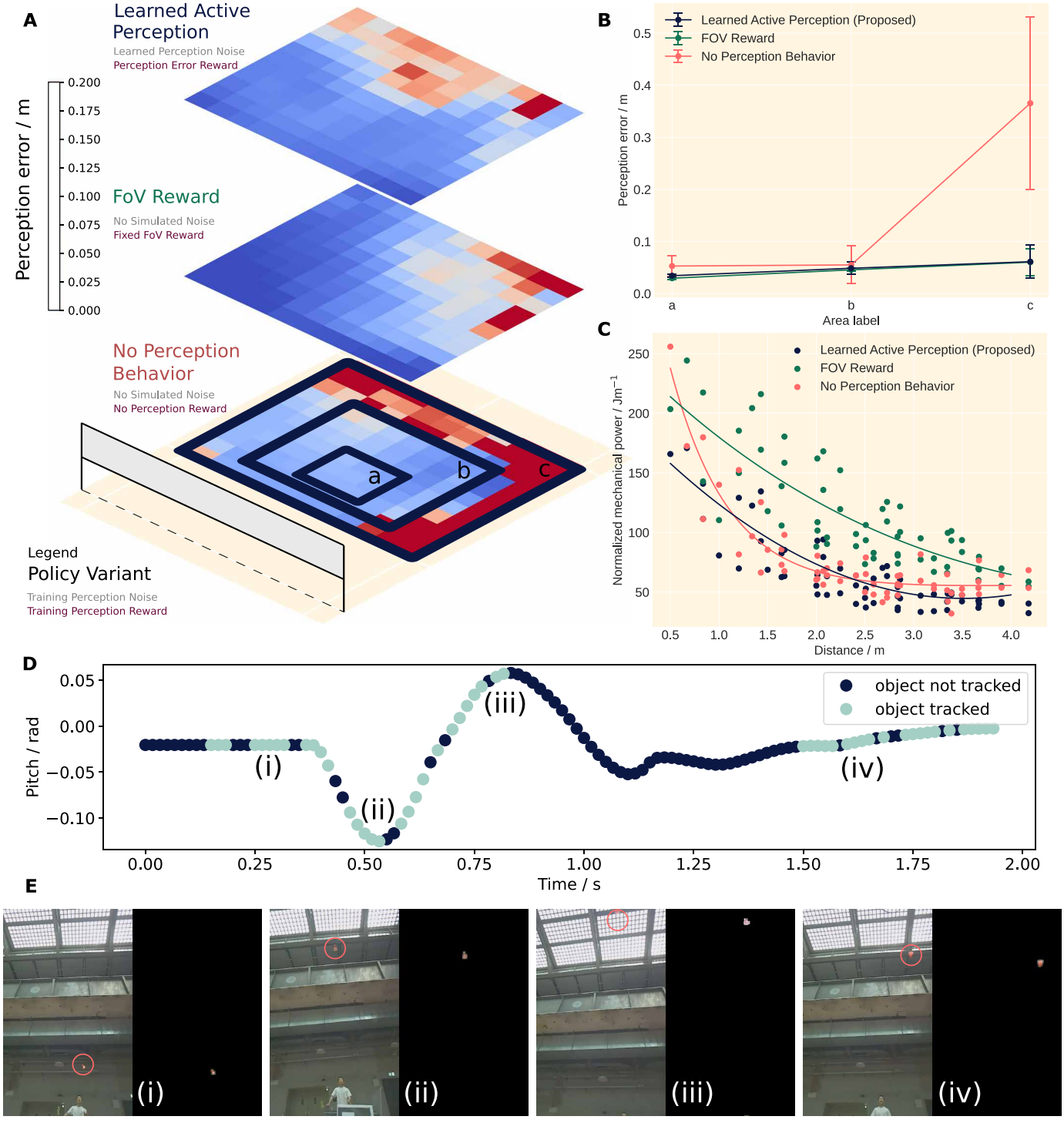


Fig. 4. Learned active perception. (**A**) Comparison of the perception error heatmap at the time of desired shuttlecock contacts. The policy trained with ground-truth shuttlecock states is not incentivized to actively track the shuttlecock, leading to a larger perception error than the other two methods. The heatmaps are laid over the badminton court. (**B**) Perception error statistics collected on the basis of different areas of the badminton court. The line graph shows the mean error, and the error bars represent the SD, computed from 1650 trajectories per cell in each region. (**C**) Mechanical power required to execute racket swings, normalized by EE travel distance. Our proposed method achieved similar mechanical power efficiency to the policy trained without a perception reward, with both outperforming the FOV reward policy. The FOV reward resulted in inefficient base attitude and locomotion control because it prioritized keeping the shuttlecock within the FOV at the expense of energy efficiency. (**D** and **E**) Example robot pitch trajectory during a hit. (i) ANYmal observed the shuttlecock circled in red. (ii) ANYmal pitched down while keeping the shuttlecock in the FOV. (iii) ANYmal pitched up to observe the shuttlecock for longer. (iv) ANYmal successfully hit back the shuttlecock, making it reappear in the FOV. Meanwhile, ANYmal returned to the stance posture.

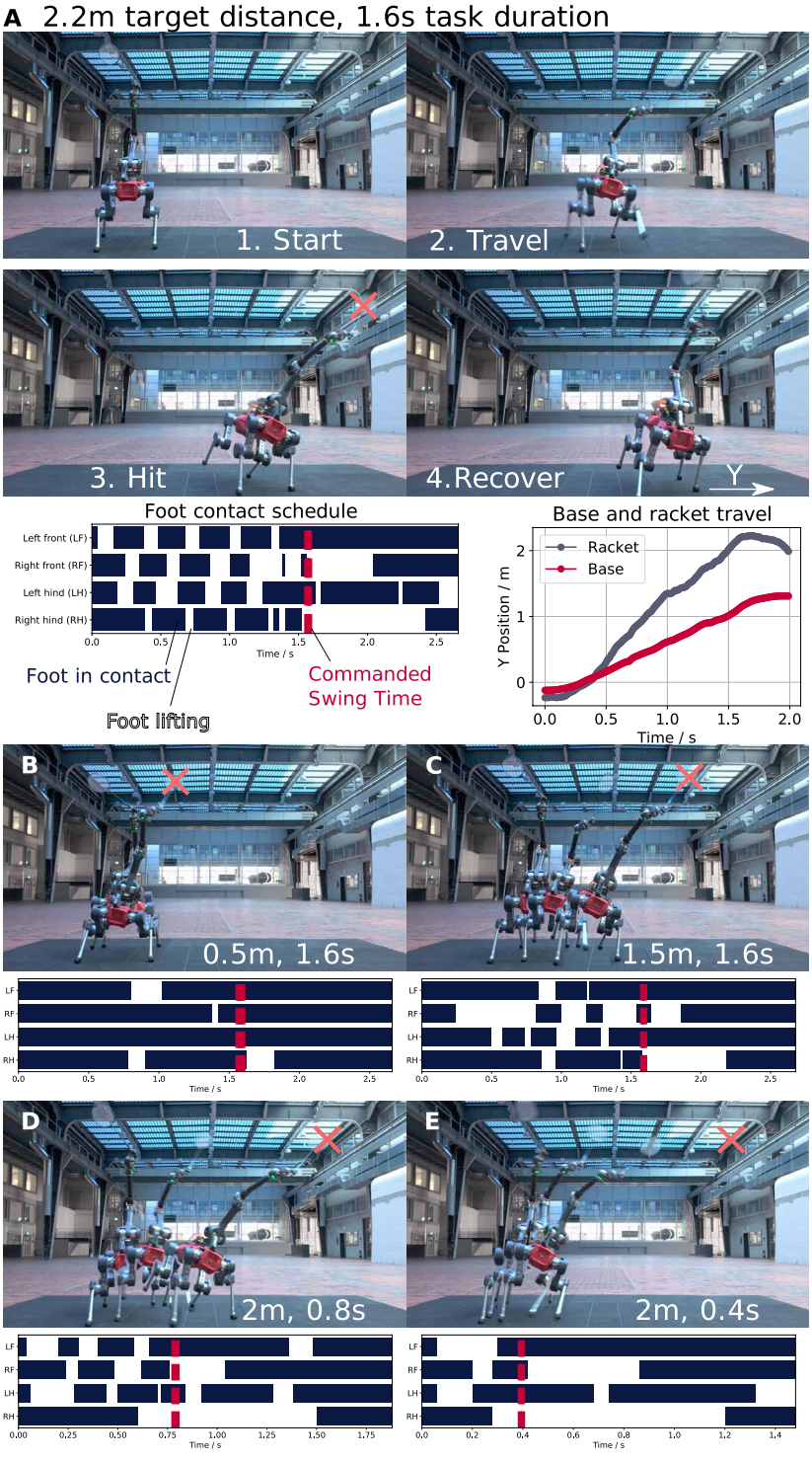


Fig. 5. Gait adaptation based on distance and time urgency. (**A**) The robot reached targets 2.2 m away from the starting base position in 1.6 s using a galloping-like gait. Near the swing, the longer RF lift phase and increased H-coordinate difference between the base and racket indicate a gait adjustment. (**B** and **C**) When targeting nearby positions, the robot barely lifted its feet. It stepped to locomote when the target was out of reach. (**D** and **E**) Under tight time constraints, the control policy balanced between maintaining safety and tracking the target accurately.

At longer distances of 2.2 m, the robot used a high-frequency gait resembling galloping between 1.6 and 0.6 s before the commanded swing. As the swing time approached, the robot adjusted its gait and prepared to lift the right legs (Fig. 5A). The extended flight phase of the right legs enabled an arm extension of 1.0 m in the direction of the target at the time of the swing. One second after swinging, the robot recovered from the dynamic pose and had all four feet in contact.

We also analyzed the gait pattern's dependency on the time the robot had to execute a maneuver. When faced with increased urgency from imminent swing targets, the robot demonstrated adaptive gaits to reach the target while prioritizing safety. The targets in this comparison were located 2 m from the robot's initial base position in the *y* direction.

Furthermore, we observed that the emergent coordination between leg and arm motion emerged under the influence of motion regularization penalties during training. In our training framework, we applied uniform joint torque and acceleration penalty scales across all joints, resulting in the robot prioritizing base tilting and arm usage for hitting nearby targets. By reducing regularization weights on the legs, we could encourage more dynamic leg movements, as demonstrated in the Supplementary Materials and Fig. 6.

When given 0.8 s to reach the target, the robot stepped with high frequency with the LF, RF, and LH feet while only making one long step with the RH leg. By extending the arm, it managed to successfully reach the swing target in time. Under a harder time constraint of 0.4 s, it was physically impossible for the robot to reach the target. Despite its attempt, the robot failed to reach the required position, resulting in a missed hit. However, it managed to avoid excessive base limb motion, showcasing the policy's robustness even when faced with unreachable commands and demonstrating awareness of its current physical limitations.

DISCUSSION

We present a legged manipulator system capable of playing badminton using only onboard perception. This system showcases advancements in coordinating legged locomotion with manipulation and balancing limb agility with perception accuracy, highlighting its potential in dynamic and competitive human sports. Through the use of multitarget training and asymmetric actor-critic RL, coupled with a perception model, the robot was able to develop sophisticated human-like badminton behaviors. These include follow-through after hitting the shuttlecock and active perception to enhance shuttle state estimation, all achieved without explicit training heuristics.

The robot's performance was extensively evaluated through various hardware experiments, including success rate assessments, tasks involving targets at different distances and under varying time constraints,

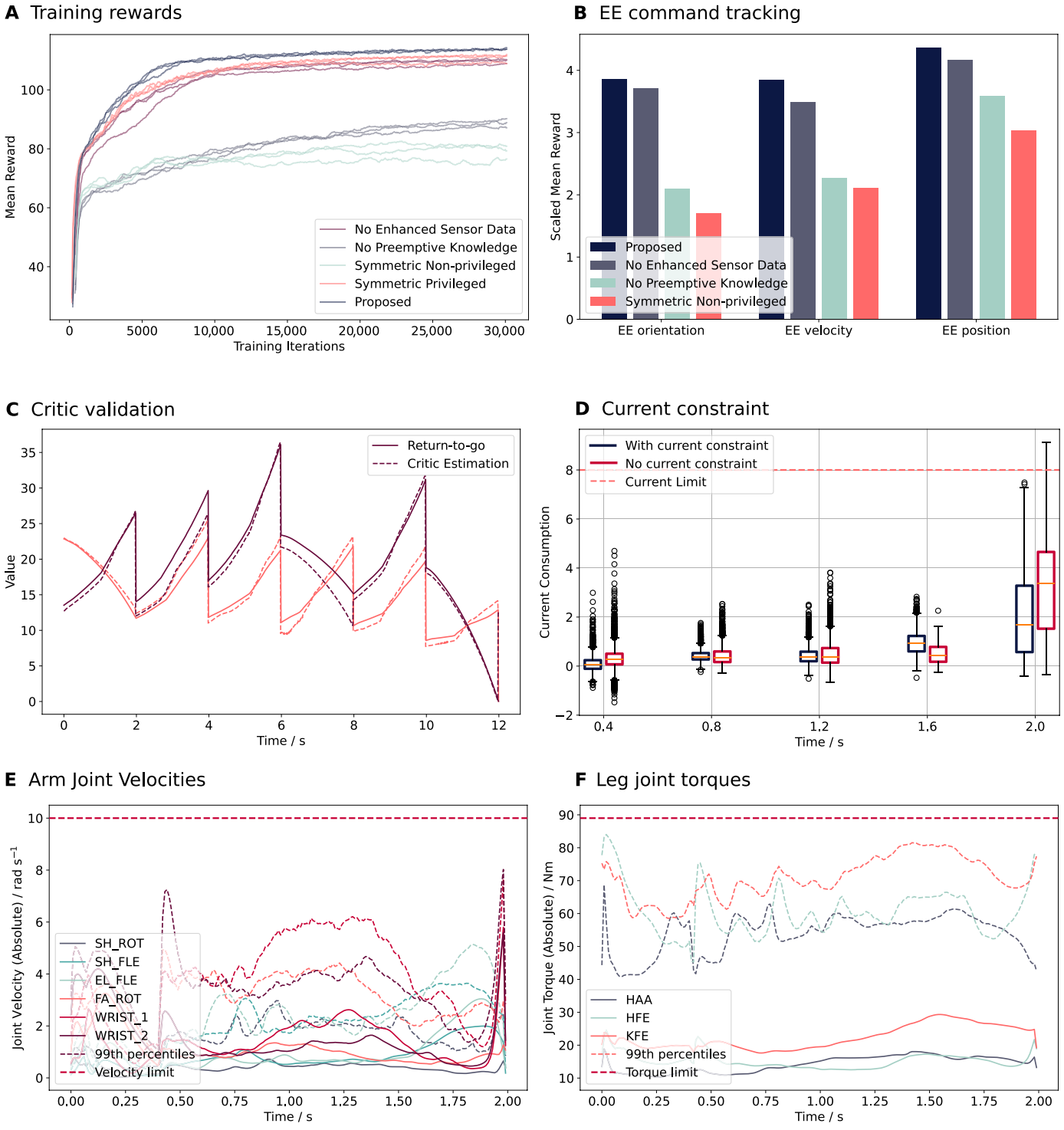


Fig. 6. Validation of the training method. (**A** and **B**) Ablation studies comparing different observation configurations show that our method consistently outperforms baseline training setups in terms of both convergence time and final EE tracking performance. (**C**) Multiple sample trajectories (represented by different colors) demonstrate that our value prediction closely aligns with the computed trajectory return. (**D**) The current consumption constraint was respected by policies trained using the N-P3O formulation. The error bars extend from the box to the farthest data points within 1.5 times the interquartile range. (**E**) Arm joint velocities over the sample swings. (**F**) Leg joint torques over the sample swings. [(D) to (F)] The box and line plots used data from all target positions on the court, each totaling 199,650 trajectories.

and verification of active perception behavior. During multiple collaborative games with humans in different environments, the system demonstrated its ability to respond to shuttle shots with varying angles, speeds, and landing locations, achieving 10 consecutive shots in a single rally under mildly windy outdoor conditions.

Incorporating the noisy perception model and using the same EKF for both training and deployment establishes a consistent mapping between motion history and expected perception outcomes across simulation and hardware. This provides a means to address a known limitation of privileged learning (teacher-student training): the information gap between the teacher policy trained with perfect perception and the student policy for deployment. In such a framework, the teacher policy has no incentive to learn active perception behaviors because it already has access to perfect observations. The student policy, trained through behavior cloning, only mimics these actions on the basis of partial observations and a latent vector reconstructed from proprioception and perception histories. As a result, neither policy develops active perception behaviors, and a discrepancy arises in the information used for control between the two policies. Our method bridges this gap by encouraging active behaviors through the aforementioned motion-perception mapping of incorporating the EKF also in training. This approach could be further extended by replacing the regressed model with learned perception models to enhance generalizability. Although this may introduce additional training complexity and computational overhead, it presents an exciting direction for future research in improving active perception learning within RL frameworks.

We identify several other promising extensions to further enhance the robot's athletic capabilities. Currently, a set of configurable rules determines the swing height, velocity, and orientation. Although the robot's high DoFs offer substantial potential for more nuanced racket control, these configurable rules underuse this capacity. A high-level badminton command policy that adapts swing commands on the basis of the opponent's body movements could improve the robot's ability to maintain rallies and increase its chances of winning. Furthermore, the current control policy was trained to hit interception targets between 0.9 and 1.4 m above the robot's base using the same side of the racket. Diversifying the swing motion by extending the training scheme could further enhance performance. Furthermore, although the policy performed well across shot directions, success rates were lower when returning shuttlecocks that landed behind the robot. This limitation stems primarily from perception constraints because giving the robot ground-truth perception, as shown in Fig. 2A, makes the performance almost symmetric. Having to maintain the shuttlecock within the FOV becomes notably harder when walking backward. A wider FOV camera or an actuated camera pitch joint could mitigate this issue.

In addition, the current system relies heavily on an EKF applied to a single off-the-shelf stereo camera for shuttlecock state estimation. This approach could be refined by integrating additional sensing modalities, such as torque and sound for impact detection, or incorporating extra RGB (red, green, and blue), depth, or event-based cameras to enhance the robot's response to physical interactions during more intense gameplay—such as when trying to hit back smash shots. Given that human players often predict shuttlecock trajectories by observing their opponents' movements, human pose estimation could also be a valuable modality for improving policy performance.

In conclusion, our research demonstrates that a legged mobile manipulator can autonomously play with human players in a full-scale sport by tightly coupling whole-body maneuvers with perception inside a single RL framework. Embedding the parameterized perception model and the same EKF used on hardware in training allows the robot learning to reduce observation error while executing agile strikes. Further simulated experiments hinted at a potential extension of the proposed framework to other legged manipulator morphologies, such as humanoids (movie S5). Beyond badminton, the method offers a template for deploying legged manipulators in other dynamic tasks where accurate sensing and rapid, whole-body responses are both critical.

MATERIALS AND METHODS

The primary goal of our system was to perceive the shuttlecock, compute the swing target, and execute the swing motion. An overview of our method is presented in Fig. 7.

RL-based dynamic whole-body visuomotor skills

We trained the robot's whole-body maneuvering policy using RL in a high-fidelity simulated environment. This environment included detailed robot dynamics, such as manipulator transmission modeling and joint actuator modeling. In addition, constrained RL was used to enforce hardware constraints specific to the robot. To further improve transferability to the physical robot, we applied domain randomization techniques, such as varying friction coefficients, adding base masses, and introducing occasional random pushes. More detailed explanations of the training environment implementation are provided later in this section and in the Supplementary Materials.

To achieve EE swing tracking and to allow the policy to learn postswing follow-through behaviors, we simulated six swing targets per episode. However, this implementation made the state value function dependent on the number of hits remaining, although this information should not be made available to the deployed policy actor. To address this, we used an asymmetric actor-critic approach (*40*) with time-based rewards (*39*) for training, as shown in Fig. 7A. In this setup, the critic network was provided with additional information, such as the number of remaining hits in the episode, to better learn the value function. The actor network only received the robot states and swing target data, which included simulated noise. Table S2 presents a detailed list of observations.

We categorized the additional critic observations into two types: enhanced sensor data and preemptive knowledge. Enhanced sensor data included higher-quality versions of existing observations, such as noiseless base and joint states, as well as the EE states. However, the EE states were also derivable via forward kinematics from joint states, albeit with noise. Preemptive knowledge referred to information used to define the MDP that was unavailable during deployment because it depended on the opponent's action, such as the number of remaining targets and the distance between the current target and the next target.

Both categories of observations enhanced the critic network's accuracy in estimating the state value function. Enhanced sensor data reduced stochasticity in the N-P3O policy gradient, and preemptive knowledge completed the set of state variables upon which the value function relied. As detailed in the following section, the training environment was structured around consecutive target swings, with rewards primarily based on EE state tracking. The number of remaining swings in an episode substantially influenced the expected return from the current state. In addition, the distance between

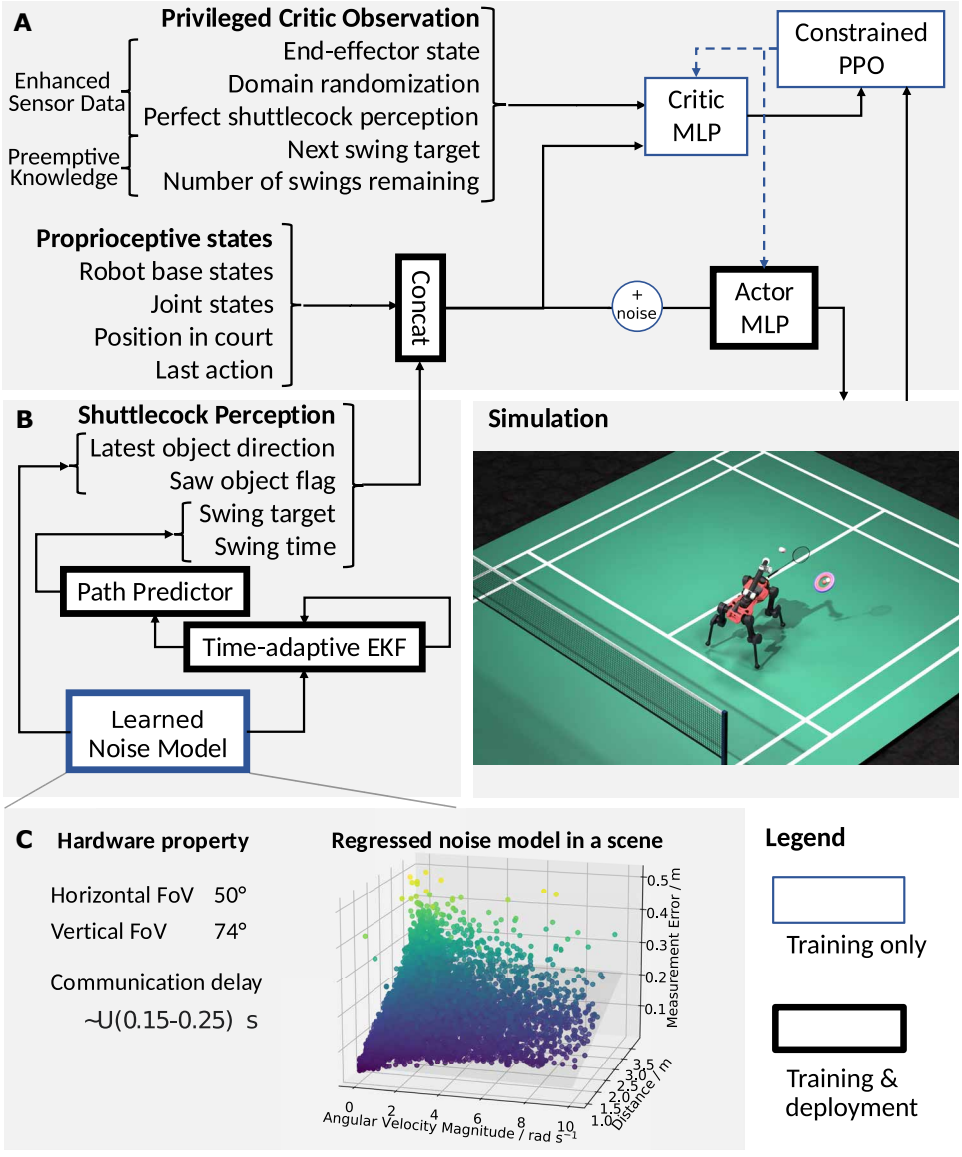


Fig. 7. Overview of the training method. (**A**) Joint control policy training with RL. The policy was trained with an asymmetric actor-critic, with privileged environment states and MDP information given only to the critic network. The policy received the noised proprioceptive states and shuttlecock perception with simulated noise. (**B**) Shuttlecock perception module in simulation. The time-adaptive EKF and the path predictor were reused during the deployment. During the training, we used a regressed perception noise model based on real camera noise collected from the hardware. (**C**) Object perception noise model. The object was in the simulation with the regressed detection probability and measurement noise if it was in the camera FOV.

consecutive targets provided the critic with insight into the anticipated motion vigor and tracking precision required, supplying predictive information that further refined return estimation and, consequently, improved policy training.

Individual ablation studies (Fig. 6, A and B) show that preemptive knowledge contributed to both the learning process and the final policy performance in EE tracking. The proposed observation format outperformed symmetric privileged training [teacher training in privileged learning (*33*)]. This improvement was due to the actor policy in symmetric training developing unnecessary time-dependent behaviors, which introduced additional challenges for value function estimation.

The critic's value function predictions closely matched the trajectory return computed from the discounted rewards cumulatively summed from the end, validating the effectiveness of the enhanced critic training (Fig. 6C). In contrast, removing the additional critic observations resulted in larger value function errors, leading to worse learning outcomes, as shown in Fig. 6B.

During training, we modeled the perception detection probability and noise as functions of the distance to the shuttlecock, the robot's angular velocity, and whether the shuttlecock was within the robot's FOV. This perception model was regressed from data collected using the robot hardware. We then used the same EKF and shuttlecock trajectory prediction module both during training and deployment, ensuring consistency, as depicted in Fig. 7B.

Swing tracking

We used a time-based swing reward mechanism to incentivize accurate and timely racket swings. This included rewards for position, orientation, and velocity, activated for a single time step per swing. The orientation reward specifically targeted the angle difference from the normal direction to the racket face.

During training, the robot observed the swing target position as the relative position between the EE and swing target, expressed in the base frame of the robot. This observation formulation helped maintain robust tracking when domain randomization was applied to the arm dynamics, which suggested smaller sim-to-real transfer challenges. Given that we expected flat terrain during the deployment, we did not implement advanced terrain types or curriculum during the training. An overall flat terrain with small unevenness was used to encourage higher foot-lifting, which helped the sim-to-real transfer. The Supplementary Materials provide additional training details, including other observations, network architecture, and hyperparameters.

Perception noise model

To fit the perception noise model, we collected data where the camera moved around while observing a fixed shuttlecock at a known location. The camera's position was tracked using a motion capture system, allowing us to compute the distance and angular velocity between the camera and the shuttlecock. Movie S4 shows the data collection procedure. The detection probability and the shuttlecock position error were regressed as a linear function of shuttle distance

and angular velocity, providing a noise model that could be deployed during the training with negligible computation overhead. An example of the regression is depicted in Fig. 7C.

Both our shuttlecock trajectory generation and the EKF followed the shuttlecock dynamics model (*41*) with a measured aerodynamic length L of 4.1 m. This aerodynamic length is defined as $L = 2\,m/\rho S C_D$, where m is the projectile mass, ρ is the air density, S is the cross-sectional area, and C_D is the air drag coefficient

$$m\frac{dv}{dt} = mg - m\|v\|\frac{v}{L} \tag{2}$$

where m is the mass of the shuttlecock, v is its velocity, and g is the gravitational acceleration. The process noise and measurement noise configurations are available in the Supplementary Materials.

During training, we integrated the perception noise model, the EKF, and shuttlecock trajectories into the learning loop. We first generated shuttlecock trajectories with random initial states and aerodynamic lengths. For each target swing in the training environment, we sampled a target swing height and a shuttle trajectory from the saved trajectory pool. We padded and translated the trajectory so that the shuttle reached the target swing height at the commanded swing time. The shuttle detection and measurement error was sampled using the regressed noise model (Fig. 7C) and given to the EKF. Because the same EKF and trajectory prediction module were used during training and on hardware, the single-frame noise introduced was filtered identically in both cases, capturing not only per-step measurement errors but also the final interception prediction error. This enabled direct penalization of observation error, rather than imposing hard-coded FOV constraints, allowing the RL algorithm to naturally balance perception accuracy and motion control for active perception learning.

To avoid the computation cost required to rollout the full shuttle prediction, we computed the final target offset linearized with respect to the current state estimation error based on the EKF estimation and the noiseless shuttlecock trajectory. Details on the trajectory distribution and the target offset approximation are presented in the Supplementary Materials.

This approach modeled the perception noise and reused the filter and prediction modules deployed on the hardware to reflect real-world conditions. Note that the perception noise level was subject to the testing site's light conditions and ambient color. We acknowledge that this was a limitation of our approach and that there was an expected perception error difference when deploying the policy in an untrained environment. However, the learned active perception behavior shown in Fig. 4 would still qualitatively transfer to different environments and decrease the perception error.

Training

The training process used the IsaacGym simulator (*48*) with the legged_gym framework (*48*, *49*). The training used the N-P3O (*42*), a constrained variant of the PPO algorithm (*50*). The policy approached the maximum training reward after around 7500 iterations (Fig. 6A), corresponding to 4.81 hours of wall-clock time on a single RTX 2080Ti GPU. For deployment, the policy was usually trained for 1 to 2 days for better convergence.

Perception deployment

In the deployment phase, we implemented several key components to ensure the accurate tracking and striking of the shuttlecock. We used a color-based filtering approach for effective shuttlecock tracking in the camera frame. Specifically, we used the hue-saturation-value (HSV) scale to filter out the shuttlecock's orange color by setting an upper and lower range for the HSV values. This enabled us to isolate the shuttlecock from the background effectively. Using the stereo information provided by the ZED X camera, we then transformed the filtered positions from two-dimensional image coordinates to the robot's map frame.

The map frame—a globally consistent world frame generated by the robot's SLAM pipeline, distinct from the odometry frame, which can accumulate drift over time—is essential for accurate localization and tracking and was derived through the integration of modular sensor fusion (MSF) (*44*) and CompSLAM (*43*). A stable and accurate map frame that properly accounted for the robot's movements was critical, given that the shuttlecock's state estimation was filtered within this frame. Any drift in the map frame would have resulted in a noisy shuttlecock estimation or directly led to incorrect interception positions when transformed into the base frame, thereby affecting the swing command observations.

Because of notable base angular velocity during the swing preparation phase, accurate timing information on the shuttle's position was required to determine its position in the world frame. For this purpose, the ZED X camera firmware provided synchronized image timestamps. Camera selection also played a key role in our perception system. We opted for a narrower FOV instead of a wide-angle camera to enhance angular resolution. This choice reduced measurement noise and improved the accuracy of shuttlecock tracking, particularly during high-speed motions where precise angular data were critical. For our perception system, we measured a total of 60- to 160-ms delay between the camera shutter time and when the shuttle's positions could be computed.

Once the shuttlecock's position in the map frame was obtained, it was processed by an EKF with parameters identical to those used during training. The EKF output a filtered shuttlecock state estimate, enabling trajectory prediction and interception point computation. In both simulation and real-world deployment, if the shuttlecock exited the robot's FOV, the system maintained the last predicted interception position for up to 2 s, during which the robot attempted to strike on the basis of this estimate.

Sim-to-real practicalities

The arm current consumption was limited to 8 A on the ANYmal robot by a fuse. The constrained RL technique N-P3O (*42*) was used to enforce arm current consumption constraints on the robot

$$|I_{\text{total}}| < 8\ \text{A} \tag{3}$$

where the total current I_{total} is calculated by summing the contributions from resistive power, derived from the motor constant K_{m} and the mechanical power, which was computed from the torque τ_i and motor velocity ω_i and dividing by the voltage V

$$I_{\text{total}} = \sum_{i=1}^{N}\left(\frac{\tau_i}{K_{\text{m}}}\right)^2\frac{1}{V} + \sum_{i=1}^{N}\frac{\tau_i \cdot \omega_i}{V} \tag{4}$$

The policy avoided the current limit with the N-P3O constraint implementation; in contrast, the baseline policy without the constraint violated the constraint even with soft overcurrent penalties. During the hardware deployment, the policy trained with N-P3O never violated the constraint.

The actuator torque and velocity constraints were included as soft penalties in the reward function because these were also treated as soft constraints on the hardware. Although exceeding these limits would not cause immediate failures, frequent violations could lead to wear or damage. The distributions of maximum arm drive velocity and leg drive torque in our test scenario are shown in Fig. 6 (E and F), respectively. Figure 6E indicates that fast motion was observed in the wrist actuators both in the backswing phase and during the swing, whereas the leg torque usage remained more consistent throughout the phases. During all phases, these values approached their limits but remained within them.

Statistical analysis

Statistical analyses were performed in Python using the NumPy library to compute means and SDs. Data were sampled at 100 Hz for simulated experiments and 400 Hz for hardware experiments. The analyses shown in Fig. 2 (A and B), Fig. 4 (A to C), and Fig. 6 (D to F) used 1650 trajectories per target position, with perturbed initial robot joint configurations. For the normalized mechanical power computation (Fig. 4C), trajectories with zero target distance were excluded to prevent division by zero. The mean training rewards shown in Fig. 6 (A and B) were averaged across 4096 parallel training environments using three random seeds. All plots in Fig. 6 were generated from simulated data using Matplotlib, with a convolution filter of window size 50 applied to Fig. 6A for smoothing.

Supplementary Materials

The PDF file includes:
Nomenclature
Methods
Figs. S1 to S7
Tables S1 to S5
Legends for movies S1 to S5
References (*51*, *52*)

Other Supplementary Material for this manuscript includes the following:
Movies S1 to S5

REFERENCES AND NOTES

1. M. H. Raibert, *Legged Robots That Balance* (MIT Press, 1986).
2. S. Le Cleac'h, T. A. Howell, S. Yang, C.-Y. Lee, J. Zhang, A. Bishop, M. Schwager, Z. Manchester, Fast contact-implicit model predictive control. *IEEE Trans. Robot.* **40**, 1617–1629 (2024).
3. B. Katz, J. Di Carlo, S. Kim, "Mini Cheetah: A platform for pushing the limits of dynamic quadruped control," in *IEEE International Conference on Robotics and Automation* (IEEE, 2019), pp. 6295–6301.
4. Y. Liu, A. Billard, Tube acceleration: Robust dexterous throwing against release uncertainty. *IEEE Trans. Robot.* **40**, 2831–2849 (2024).
5. J.-R. Chiu, J.-P. Sleiman, M. Mittal, F. Farshidian, M. Hutter, "A collision-free MPC for whole-body dynamic locomotion and manipulation," in *IEEE International Conference on Robotics and Automation* (IEEE, 2022), pp. 4686–4693.
6. D. Crowley, J. Dao, H. Duan, K. Green, J. Hurst, A. Fern, "Optimizing bipedal locomotion for the 100m dash with comparison to human running," in *IEEE International Conference on Robotics and Automation* (IEEE, 2023), pp. 12205–12211.
7. G. Ji, J. Mun, H. Kim, J. Hwangbo, Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robot. Autom. Lett.* **7**, 4630–4637 (2022).
8. G. B. Margolis, G. Yang, K. Paigwar, T. Chen, P. Agrawal, Rapid locomotion via reinforcement learning. *Int. J. Robot. Res.* **43**, 572–587 (2024).
9. T. He, C. Zhang, W. Xiao, G. He, C. Liu, G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," in *Robotics: Science and Systems XX*, D. Kulic, G. Venture, K. Bekris, E. Coronado, Eds. (RSS Foundation, 2024).
10. D. Hoeller, N. Rudin, D. Sako, M. Hutter, Anymal parkour: Learning agile navigation for quadrupedal robots. *Sci. Robot.* **9**, eadi7566 (2024).
11. X. Cheng, K. Shi, A. Agarwal, D. Pathak, "Extreme parkour with legged robots," in *IEEE International Conference on Robotics and Automation* (IEEE, 2024), pp. 11443–11450.
12. Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, H. Zhao, "Robot parkour learning," in *Proceedings of the 7th Conference on Robot Learning*, J. Tan, M. Toussant, K. Darvish, Eds. (PMLRPress, 2023), pp. 73–92.
13. K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang, N. Batchelor, S. Bohez, F. Casarini, J. E. Chen, O. Cortes, E. Coumans, A. Dostmohamed, G. Dulac-Arnold, A. Escontrela, E. Frey, R. Hafner, D. Jain, B. Jyenis, Y. Kuang, E. Lee, L. Luu, O. Nachum, K. Oslund, J. Powell, D. Reyes, F. Romano, F. Sadeghi, R. Sloat, B. Tabanpour, D. Zheng, M. Neunert, R. Hadsell, N. Heess, F. Nori, J. Seto, C. Parada, V. Sindhwani, V. Vanhoucke, J. Tan, Barkour: Benchmarking animal-level agility with quadruped robots. arXiv:2305.14654 [cs.RO] (2023).
14. Y. Ji, G. B. Margolis, P. Agrawal, "Dribblebot: Dynamic legged manipulation in the wild," in *IEEE International Conference on Robotics and Automation* (IEEE, 2023), pp. 5155–5162.
15. T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, J. Humplik, M. Wulfmeier, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner, M. Bloesch, K. Hartikainen, A. Byravan, L. Hasenclever, Y. Tassa, F. Sadeghi, N. Batchelor, F. Casarini, S. Saliceti, C. Game, N. Sreendra, K. Patel, M. Gwira, A. Huber, N. Hurley, F. Nori, R. Hadsell, N. Heess, Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Sci. Robot.* **9**, eadi8022 (2024).
16. D. B. D'Ambrosio, S. Abeyruwan, L. Graesser, A. Iscen, H. B. Amor, A. Bewley, B. J. Reed, K. Reymann, L. Takayama, Y. Tassa, K. Choromanski, E. Coumans, D. Jain, N. Jaitly, N. Jaques, S. Kataoka, Y. Kuang, N. Lazic, R. Mahjourian, S. Q. Moore, K. Oslund, A. Shankar, V. Sindhwani, V. Vanhoucke, G. Vesom, P. Xu, P. R. Sanketi, Achieving human level competitive robot table tennis. arXiv:2408.03906 [cs.RO] (2024).
17. T. Ding, L. Graesser, S. Abeyruwan, D. B. D'Ambrosio, A. Shankar, P. Sermanet, P. R. Sanketi, C. Lynch, "Learning high speed precision table tennis on a physical robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2022), pp. 10780–10787.
18. H. Ma, D. Büchler, B. Schölkopf, M. Muehlebach, Reinforcement learning with model-based feedforward inputs for robotic table tennis. *Auton. Robots* **47**, 1387–1403 (2023).
19. S. W. Abeyruwan, L. Graesser, D. B. D'Ambrosio, A. Singh, A. Shankar, A. Bewley, D. Jain, K. M. Choromanski, P. R. Sanketi, "i-sim2real: Reinforcement learning of robotic policies in tight human-robot interaction loops," in *Proceedings of the 6th Conference on Robot Learning*, K. Liu, D. Kulic, J. Ichnowski, Eds. (PMLRPress, 2023), pp. 212–224.
20. H. Ferrolho, V. Ivan, W. Merkt, I. Havoutis, S. Vijayakumar, Roloma: Robust loco-manipulation for quadruped robots with arms. *Auton. Robots* **47**, 1463–1481 (2023).
21. S. Zimmermann, R. Poranne, S. Coros, "Go fetch! Dynamic grasps using Boston Dynamics Spot with external robotic arm," in *IEEE International Conference on Robotics and Automation* (IEEE, 2021), pp. 4488–4494.
22. Y. Ma, F. Farshidian, T. Miki, J. Lee, M. Hutter, Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators. *IEEE Robot. Autom. Lett.* **7**, 2377–2384 (2022).
23. J.-P. Sleiman, F. Farshidian, M. V. Minniti, M. Hutter, A unified MPC framework for whole-body dynamic locomotion and manipulation. *IEEE Robot. Autom. Lett.* **6**, 4688–4695 (2021).
24. N. Yokoyama, A. Clegg, J. Truong, E. Undersander, T.-Y. Yang, S. Arnaud, S. Ha, D. Batra, A. Rai, ASC: Adaptive skill coordination for robotic mobile manipulation. *IEEE Robot. Autom. Lett.* **9**, 779–786 (2024).
25. M. Liu, Z. Chen, X. Cheng, Y. Ji, R. Qiu, R. Yang, X. Wang, "Visual whole-body control for legged loco-manipulation," in *Proceedings of the 8th Conference on Robot Learning*, P. Agrawal, O. Kroemer, W. Burgard, Eds. (PMLRPress, 2024), pp. 234–257.
26. J. Dao, H. Duan, A. Fern, "Sim-to-real learning for humanoid box loco-manipulation," in *IEEE International Conference on Robotics and Automation* (IEEE, 2024), pp. 16930–16936.
27. Z. Fu, X. Cheng, D. Pathak, "Deep whole-body control: Learning a unified policy for manipulation and locomotion," in *Proceedings of the 6th Conference on Robot Learning*, K. Liu, D. Kulic, J. Ichnowski, Eds. (PMLRPress, 2023), pp. 138–149.
28. Y. Ma, F. Farshidian, M. Hutter, "Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators," in *IEEE International Conference on Robotics and Automation* (IEEE, 2023), pp. 12149–12155.
29. H. Zhang, Y. Yuan, V. Makoviychuk, Y. Guo, S. Fidler, X. B. Peng, K. Fatahalian, Learning physically simulated tennis skills from broadcast videos. *ACM Trans. Graph.* **42**, 1–14 (2023).
30. Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, C. Finn, "HumanPlus: Humanoid Shadowing and imitation from humans," in *Proceedings of the 8th Conference on Robot Learning*, P. Agrawal, O. Kroemer, W. Burgard, Eds. (PMLRPress, 2024), pp. 2828–2844.
31. X. Cheng, Y. Ji, J. Chen, R. Yang, X. Wang, Expressive whole-body control for humanoid robots. arXiv:2402.16796 [cs.RO] (2024).
32. H. Ha, Y. Gao, Z. Fu, J. Tan, S. Song, UMI on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. arXiv:2407.10353 [cs.RO] (2024).
33. J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.* **5**, eabc5986 (2020).

34. T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.* **7**, eabk2822 (2022).
35. A. Agarwal, A. Kumar, J. Malik, D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Proceedings of the 6th Conference on Robot Learning*, K. Liu, D. Kulic, J. Ichnowski, Eds. (PMLRPress, 2023), pp. 403–415.
36. D. Hoeller, N. Rudin, C. Choy, A. Anandkumar, M. Hutter, Neural scene representation for locomotion on structured terrain. *IEEE Robot. Autom. Lett.* **7**, 8667–8674 (2022).
37. C. Schwarke, V. Klemm, M. Van der Boon, M. Bjelonic, M. Hutter, "Curiosity-driven learning of joint locomotion and manipulation tasks," in *Proceedings of the 7th Conference on Robot Learning*, J. Tan, M. Toussant, K. Darvish, Eds. (PMLRPress, 2023), pp. 2594–2610.
38. Open-Ended Learning Team, A. Stooke, A. Mahajan, C. Barros, C. Deck, J. Bauer, J. Sygnowski, M. Trebacz, M. Jaderberg, M. Mathieu, N. McAleese, N. Bradley-Schmieg, N. Wong, N. Porcel, R. Raileanu, S. Hughes-Fitt, V. Dalibard, W. M. Czarnecki, Open-ended learning leads to generally capable agents. arXiv:2107.12808 [cs.LG] (2021).
39. N. Rudin, D. Hoeller, M. Bjelonic, M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2022), pp. 2497–2503.
40. L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, P. Abbeel, "Asymmetric actor critic for image-based robot learning," in *Robotics: Science and Systems XIV* (RSS Foundation, 2018).
41. C. Cohen, B. D. Texier, D. Quéré, C. Clanet, The physics of badminton. *New J. Phys.* **17**, 063001 (2015).
42. J. Lee, L. Schroth, V. Klemm, M. Bjelonic, A. Reske, M. Hutter, Evaluation of constrained reinforcement learning algorithms for legged locomotion. arXiv:2309.15430 [cs.RO] (2023).
43. S. Khattak, H. Nguyen, F. Mascarich, T. Dang, K. Alexis, "Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments," in *International Conference on Unmanned Aircraft Systems* (IEEE, 2020), pp. 1024–1029.
44. S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2013), pp. 3923–3929.
45. J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, M. Hutter, Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* **4**, eaau5872 (2019).
46. J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, V. Vanhoucke, Sim-to-real: Learning agile locomotion for quadruped robots. arXiv:1804.10332 [cs.RO] (2018).
47. M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, M. Hoepflinger, "ANYmal—A highly mobile and dynamic quadrupedal robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2016), pp. 38–44.
48. V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, G. State, "Isaac Gym: High performance GPU based physics simulation for robot learning," in *Proceedings of the Neural Information Processing Systems Track* on *Datasets and Benchmarks*, J. Vanschoren, S. Yeung, Eds. (NeurIPS, 2021).
49. N. Rudin, D. Hoeller, P. Reist, M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proceedings of the 5th Conference on Robot Learning*, A. Faust, D. Hsu, G. Neumann, Eds. (PMLRPress, 2022), pp. 91–100.
50. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms. arXiv:1707.06347 [cs.LG] (2017).
51. G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 8 November 2000.
52. M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, S. Gelly, O. Bachem, What matters in on-policy reinforcement learning? A large-scale empirical study, paper presented at ICLR 2021: The Ninth International Conference on Learning Representations, Vienna, Austria, 3 to 7 May 2021 (virtual).

Acknowledgments: We thank C. Chen and K. Qu for dedicating extensive time as the robot's opponents. We also thank N. Rudin for initial project discussions and F. Tischhouser for extensive hardware engineering support. We are grateful to D. Vogal for insightful feedback on Movie 1. Additional thanks to J. He, T. An, J. Preisig, F. Yang, M. Mittal, Y. Shen, T. Miki, F. Bjelonic, and J.-R. Chiu for assistance in experiments and data collection and E. Sako for hardware insights. We acknowledge K. Kawaharazuka, D. Hoeller, and J. Lee for project discussions and E. Elbir, E. Schnieder, A. Binkert, A. Graf, J. Schwabe, F. Schindele, and L. Schmid for computer-aided design and infrastructure support. We also used ChatGPT and DeepSeek to assist with revising and refining the language in this paper. **Funding:** This work was supported by Intel Labs, the Max Planck ETH Center for Learning Systems, and the National Centre of Competence in Research Robotics (NCCR Robotics). In addition, this work was conducted as part of ANYmal Research, a community dedicated to advancing legged robotics. **Author contributions:** Conceptualization, simulation, sensor selection, data collection, policy training, experiments, investigation, analysis, visualization, and writing: Y.M. Sensor selection and substantial revision: A.C. Initial project discussions and revision: F.F. Initial task scope, sensor selection, resources, supervision, and revision: M.H. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** The data for this study have been deposited in the Zenodo database: https://doi.org/10.5281/zenodo.15242151. All other data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials.

Submitted 5 November 2024
Accepted 29 April 2025
Published 28 May 2025
10.1126/scirobotics.adu3922