REBot: Reflexive Evasion Robot for Instantaneous Dynamic Obstacle Avoidance

Zihao Xu[1], **Ce Hao**[*1], **Chunzheng Wang**[1], **Kuankuan Sima**[1], **Fan Shi**[1], **Jin Song Dong**[1]
[1] National University of Singapore
[*] Corresponding to `cehao@u.nus.edu`

Abstract: Dynamic obstacle avoidance (DOA) is critical for quadrupedal robots operating in environments with moving obstacles or humans. Existing approaches typically rely on navigation-based trajectory replanning, which assumes sufficient reaction time and leading to fails when obstacles approach rapidly. In such scenarios, quadrupedal robots require reflexive evasion capabilities to perform instantaneous, low-latency maneuvers. This paper introduces Reflexive Evasion Robot (REBot), a control framework that enables quadrupedal robots to achieve real-time reflexive obstacle avoidance. REBot integrates an avoidance policy and a recovery policy within a finite-state machine. With carefully designed learning curricula and by incorporating regularization and adaptive rewards, REBot achieves robust evasion and rapid stabilization in instantaneous DOA tasks. We validate REBot through extensive simulations and real-world experiments, demonstrating notable improvements in avoidance success rates, energy efficiency, and robustness to fast-moving obstacles. Videos are available on https://rebot-2025.github.io/.

Keywords: Reflexive Evasion, Dynamic Obstacle Avoidance, Quadruped Locomotion

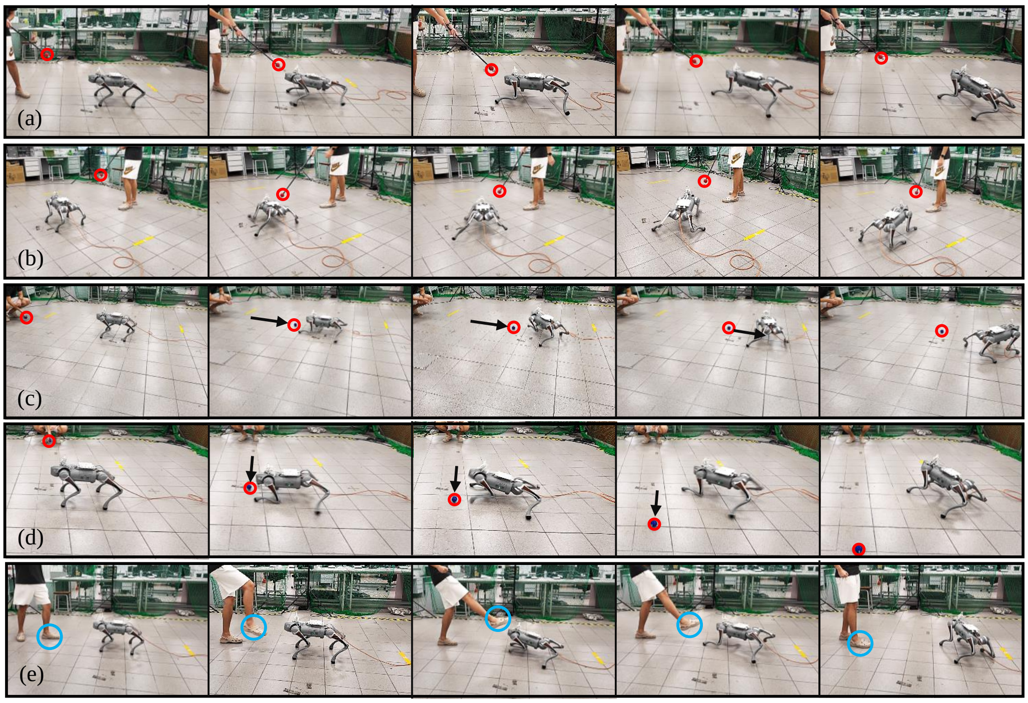


Figure 1: The Reflexive Evasion Robot (**REBot**) system achieves instantaneous dynamic obstacle avoidance. When the fast-moving obstacles approach the quadrupedal robots (reaction time<1.5s), REBot switches to the avoidance policy and performs reflexive evasion maneuvers. In (a) and (b), the robot is poked on the frontal and dorsal sides using a stick; in (c) and (d), a ball is launched toward the robot from both frontal and lateral directions; in (e), to evaluate robustness, the quadrupedal robot was subjected to intentional kicks from multiple directions. Experimental results demonstrate that the REBot system successfully controlled the quadrupedal robot to avoid all obstacles.

Workshop on Safe and Robust Robot Learning for Operation in the Real World at CoRL 2025.

1 Introduction

Ensuring the safety of a robot is essential during task execution [1]. For instance, a mobile robot must not only perform its primary tasks but also perceive surrounding obstacles and take appropriate evasive actions [2, 3, 4]. When encountering slow-moving obstacles (reaction time >2s), the robot typically has sufficient time to stop its current actions and replan a new trajectory using a decision-making model to avoid collisions [5, 6]. This type of behavior is commonly referred to as dynamic obstacle avoidance (DOA) via navigation-based trajectory replanning [7]. For legged robots, such as quadrupedal robots, DOA involves both high-level navigation decision-making and low-level locomotion control [8]. For example, in the Agile but Safe (ABS) framework [9], a quadrupedal robot encountering a quasi-static obstacle during high-speed locomotion can reduce its speed and replan a navigation trajectory to safely bypass the obstacle (Additional related works are provided in the appendix).

However, when obstacles approach at high speeds, the robot is left with extremely limited reaction time (<1.5s), necessitating immediate evasive maneuvers [10]. Due to limitations in mechanical structure and motor power, the robot often fails to generate sufficient velocity within the available time to accurately track a replanned navigation trajectory [11]. To achieve instantaneous DOA, we draw inspiration from the spinal reflex systems of vertebrates. Unlike decision-making processes governed by the brain, spinal reflexes enable rapid, localized decisions through neural circuits in the spinal cord, allowing animals to execute unconventional evasive actions instantaneously [12]. For example, an antelope might execute a sudden backward leap to evade an ambush from an underwater crocodile while drinking, relying entirely on reflexive evasion.

In this paper, we propose the Reflexive Evasion Robot (REBot) system for instantaneous dynamic obstacle avoidance. Using the quadrupedal robot Unitree Go2 as an example platform [13, 14], REBot demonstrates real-time evasion of high-speed obstacles with a reaction time of less than 1.5 seconds. The REBot system is structured as a finite-state machine with three behavioral stages. During the normal stage, the robot performs its primary functional tasks. When an approaching obstacle is detected, REBot transitions to the avoidance stage, executing reflexive evasion maneuvers. During evasion, a PPO [15, 16, 17] reinforcement learning policy enables rapid avoidance while preserving the robot's safety, balance, and energy efficiency. After an evasive maneuver, the robot may become unstable. REBot then enters the recovery stage, during which a policy stabilizes the robot and restores normal function.

We trained the REBot system for quadrupedal robots in Isaac Gym simulator [18], evaluated its performance, and deployed it on a real robot for demonstration. REBot achieved the highest avoidance and recovery success rates in both static and dynamic obstacle scenarios, while reducing maximum joint power and avoidance distance. We observed that the robot's reflexive evasion performance varied with obstacle direction and speed; it performed best when avoiding frontal obstacles due to its structural advantages in backward maneuvers. Ablation studies confirmed that the recovery policy, curriculum learning, and adaptive reward design significantly improved avoidance success rates. Finally, real-world experiments (Fig. 1) validated REBot's capability for real-time, instantaneous dynamic obstacle avoidance, offering insights into robot safety system design.

In summary, the contributions of this paper are as follows:

- We formally identify and formulate the reflexive evasion problem for dynamic obstacle avoidance in quadrupedal robots.
- We design the REBot system as a finite-state machine integrating avoidance and recovery policies to achieve robust, real-time reflexive evasion.
- We conduct comprehensive simulations and real-world experiments with thorough analysis to validate REBot's effectiveness across various obstacle scenarios.

2 Preliminary

Problem formulation. Fig. 2 shows that the dynamic obstacle avoidance (DOA) system has two entities: dynamic obstacles and quadruped robots. The dynamic obstacles O are modeled as a rigid sphere with states (r^O, p_t^O, v_t^O) of radius, position, velocity and acceleration in 3D space. The quadruped robot R is a high-dimensional articulated system with a robot base and four independently actuated legs. The states s_t^R consist of base position $p_t^R \in \mathbb{R}^3$, linear velocity $v_t^R \in \mathbb{R}^3$, angular velocity $\omega_t^R \in \mathbb{R}^3$, joint position $q_t^R \in \mathbb{R}^{12}$, joint velocity $\dot{q}_t^R \in \mathbb{R}^{12}$, joint torque $\tau_t^R \in \mathbb{R}^{12}$, contact force $f_t^R \in \mathbb{R}^{4\times 3}$ and orientation angle $\theta_t^R \in \mathbb{R}^3$. The robot is driven by servo motors on the joints to move on the ground via action a_t^R, where $a_t^R \in \mathbb{R}^{12}$ denotes the joint target angles. In this work, we utilize the Unitree Go2 robot to conduct experiments. We define a successful dynamic obstacle avoidance (DOA) as the robot maintaining collision-free motion throughout the task duration. A collision is considered to occur if the signed distance function (SDF) from the obstacle center p_t^O to the robot's oriented bounding box (OBB), denoted as $\mathcal{B}^R$, is smaller than the obstacle's radius r^O; that is, if $d(p_t^O, \mathcal{B}^R) < r^O$.

Dynamic obstacle Avoidance Strategies. The robot adopts a control system with observation $(p_t^R, \omega_t^R, q_t^R, \dot{q}_t^R, \tau_t^R, f_t^R, p_t^O, v_t^O, r^O)$ and action (a_t^R), enabling the robot to avoid approaching obstacles while maintaining balance. When faced with static or slow-moving obstacles, the robot can temporarily stop and replan its trajectory, a behavior classified as **navigation avoidance** [9]. However, when the obstacles approach instantaneously, the robot must react immediately. We categorize such behavior as **reflexive evasion** (Fig. 2). The reaction time T_{react} determines the reaction of navigation or reflex, and we distinguish these two behaviors by the maximum joint power of the robot. In the following sections, we design the REBot system to achieve the reflexive DOA and analyze the behaviors in both simulation and real-world experiments.

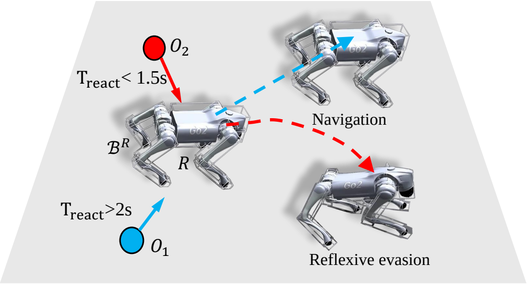


Figure 2: Robot dynamic obstacle avoidance.

3 Method: Reflexive Evasion Robot

In this section, we design the reflexive evasion robot (**REBot**) system to achieve DOA in Fig. 3(a). As shown in Fig. 3, the REBot system consists of three behavioral stages organized as a finite state machine (FSM). In the following sections, we introduce the FSM stages and transition criteria (Sec. 3.1), the training strategies of the avoidance policy (Sec. 3.2) and the recovery policy (Sec. 3.3). Finally, we delineate the training and deployment of the REBot system in simulation and on real quadruped robots, respectively (Sec. 3.4).

3.1 REBot Stages and Transition Criteria

The robot initially stays in the **normal stage** with functional behaviors such as standing, walking, or trotting. During this stage, the robot continuously observes the environment and potential obstacles. In this work, we define the normal behavior as standing still while maintaining balance via the PD controller. When the obstacle is approaching the robot, (i.e., $\langle v_t^O,\, p_t^R - p_t^O \rangle > 0$), REBot switches to the **avoidance stage** to execute reflexive evasion, in which the avoidance policy performs reactive maneuvers under constrained reaction time.

However, severe evasion may cause the robots' instability. Therefore, the REBot correspondingly switches to the **recovery stage**, where a recovery policy drives the robot back to normal functions. REBot judges the instability with three criteria: (i) body orientation exceeds a safe range $\|\theta_t^R\| > \theta_{\text{th}}^R$; (ii) joint velocity surpasses a stability limit $\|\dot{q}_t^R\| > \dot{q}_{\text{th}}^R$; (iii) base height drops below a threshold value $h_t^R < h_{\text{th}}^R$. Here, θ_t^R, $\dot{q}_t^R$, and $h_t^R \in \mathbb{R}$ denote the robot's orientation, joint velocity, and base height, respectively. The corresponding thresholds are θ_{th}^R, $\dot{q}_{\text{th}}^R$, and h_{th}^R.

3.2 Avoidance Policy

The avoidance policy is trained via RL to achieve reflexive evasion under constrained reaction time. The objective is to avoid collisions while maintaining postural stability and minimizing energy consumption. The reward function consists of three parts: $r = r_{\text{avoidance}} + r_{\text{regularization}} + r_{\text{adaptive}}$.

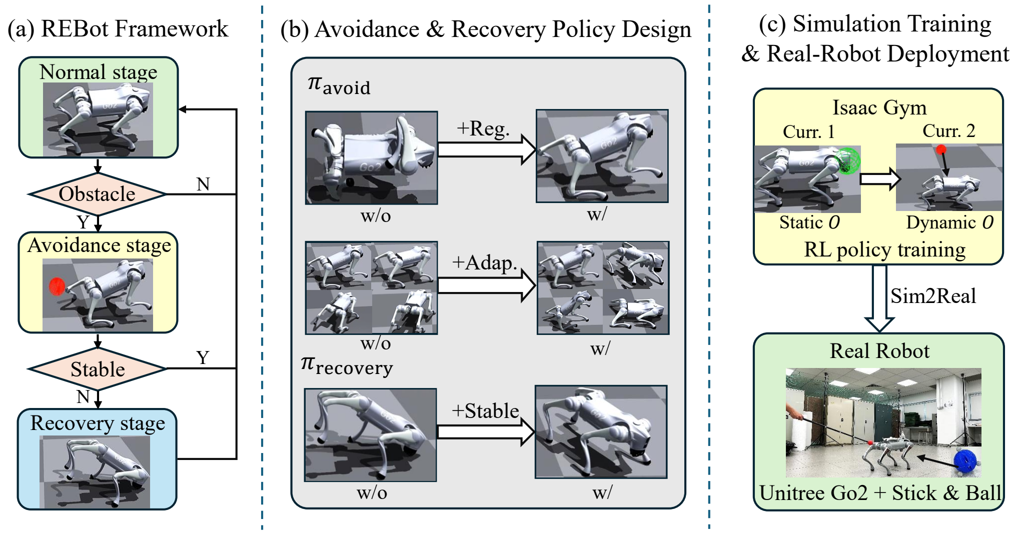


Figure 3: **(a) REBot Framework:** A finite-state machine (FSM) governs transitions between the Normal, Avoidance, and Recovery stages. The quadrupedal robot performs reflexive evasion to avoid obstacles (red balls). **(b) Policy Design:** The avoidance policy is trained not only for successful obstacle avoidance but also incorporates regulation rewards for state stabilization and adaptive rewards to encourage diverse evasive behaviors. **(c) Training and Deployment:** REBot is trained in Isaac Gym using a two-stage curriculum from static to dynamic obstacles, and deployed on a real Unitree Go2 robot.

Avoidance reward is composed of $r_{\text{avoidance}} = r_{\text{distance}} + r_{\text{collision}}$. The distance reward is defined as $r_{\text{distance}} = -\exp(-(d(p_t^O, \mathcal{B}^R) - r^O))$ to encourage the robot to maintain a safe distance from the moving obstacle throughout the entire task. The collision penalty is defined as $r_{\text{collision}} = \mathbf{1}(c = 0) - \mathbf{1}(c = 1)$ to penalize any contact event, where $c \in \{0, 1\}$ denotes the collision. We adopt a two-stage curriculum to improve policy training efficiency. In the first stage, a static obstacle appears instantaneously in a random location near the robot at a predefined activation time. In the second stage, the obstacle follows a directed trajectory toward the robot at varying speeds, simulating realistic dynamic threats.

Regularization reward consists of three terms $r_{\text{regularization}} = r_{\text{walk}} + r_{\text{energy}} + r_{\text{contact}}$. It is designed to ensure the learned evasive maneuvers remain both stable and natural (Fig. 3(b)). To promote natural and coordinated motion, we encourage symmetric limb phasing consistent with a trot gait. The term $r_{\text{walk}} = \frac{1}{2}\left(\mathbf{1}(c_{\text{FL}} = c_{\text{RR}}) + \mathbf{1}(c_{\text{FR}} = c_{\text{RL}})\right)$ rewards synchronized contact patterns between diagonal leg pairs, where $c_{i,j}$ denotes different contact leg. We penalize the product of joint torque and joint velocity across all actuated degrees of freedom to reduce excessive power consumption through $r_{\text{energy}} = -\sum_i |\tau_t^{R,i} \cdot \dot{q}_t^{R,i}|$, where $\tau_t^{R,i}$ and $\dot{q}_t^{R,i}$ represent the torque and angular velocity of each joint respectively. We also penalize the temporal fluctuation of vertical foot contact forces to reduce instability via $r_{\text{contact}} = -\sum_i \left(f_t^{R,i,z} - f_{t-1}^{R,i,z}\right)^2$, where $f_t^{R,i,z}$ denotes the vertical foot contact force of each leg.

Adaptive reward is designed as $r_{\text{adaptive}} = r_{\text{diversity}} + r_{\text{threat}} + r_{\text{direction}}$ to encourage motion diversity, speed modulation, and direction efficiency [19]. RL policy tends to converge toward a single locally optimal behavior (Fig. 3(b)). In this task, it manifests as the robot repeatedly using a fixed evasion gait regardless of obstacle state. We defined a diversity reward to encourage the policy to appropriate behaviors $r_{\text{diversity}} = \text{Var}_{s^R \sim \mathcal{D}_{s^R}}\left[\pi(a_t^R|s_t^R)\right]$. We define the threat level of the obstacle through the reaction time and the robot learns to adapt its speed in response to the levels of perceived threat as $r_{\text{threat}} = -\|v_t^{\text{R}} - v_t^{\text{R,safe}}\|, \quad v_t^{\text{R,safe}} = v_t^{\text{R,cmd}} + \lambda \exp(-\eta T_{\text{reaction}})$, where $v_t^{\text{R,cmd}}$ denotes the command velocity, λ and η denote the hyperparameters. To discourage evasive movements that deviate unnecessarily from the ideal escape direction, we penalize wrong the robot movement direction via $r_{\text{direction}} = -\langle v_t^R, p_t^O - p_t^R \rangle$.

3.3 Recovery Policy

The recovery policy ensures a smooth transition from the avoidance stage back to the normal stage, allowing the robot to regain balance (Fig. 3(b)). Therefore, the reward function $r = r_{\text{orientation}} + r_{\text{stable}} + r_{\text{position}} + r_{\text{additional}}$ is designed corresponding to the instability criteria. The orientation reward defined as $r_{\text{orientation}} = -\sum_i (\theta_t^{R,i} - \theta_0^{R,i})^2$ penalizes excessive tilt of the robot, where $\theta_0^{R,i} \in \mathbb{R}$ denotes the default orientation angles. The stable reward $r_{\text{stable}} = \sum_i \exp(-|\dot{q}_t^{R,i}|)$ encourages low joint velocities via exponential decay. And the position term $r_{\text{position}} = -\|p_t^R - p_0^R\|^2$ penalizes too slow base height, where p_0^R denotes the default position. The additional reward term $r_{\text{additional}}$ includes penalties on large joint torque and action discontinuities. These components are designed to reduce abrupt joint movements and encourage smoother transitions during recovery.

3.4 Training in Simulation and Real-Robot Deployment

We implement the REBot system on the Unitree Go2 quadrupedal robot (Fig. 3(c)). We train the avoidance and recovery policies in the Isaac gym simulator [20] with PPO algorithm [21, 22, 23]. Specifically, the avoidance policy is trained in two curricula. First, a stationary obstacle is randomly placed around the robot and activates after a delay; Go2 must react within the available response time. Second, a moving obstacle approaches with fixed velocity from a random direction, requiring real-time evasion. Both curricula randomize obstacle parameters to prevent overfitting and encourage generalization across planning-based and reflexive behaviors. Then we deploy the REBot system to the real Unitree Go2 robot. A motion capture system was used to provide real-time ground truth position data for both the robot and the dynamic obstacle. To emulate dynamic obstacles, we used a rigid rod with a lightweight ball attached to its tip, serving as a physical proxy for an incoming object. The obstacle's position was continuously tracked via reflective markers.

4 Simulation Experiments

We validate and estimate the performance of REBot in the simulation system. In this section, we answer three questions: **Q1** Can REBot achieve successful evasion under instantaneous DOA? (Sec. 4.2) **Q2** What are the robots' reactions under different obstacle conditions? (Sec. 4.3) **Q3** How can the rewards' design and recovery stage influence DOA performance? (Sec. 4.4)

4.1 Experiment Settings

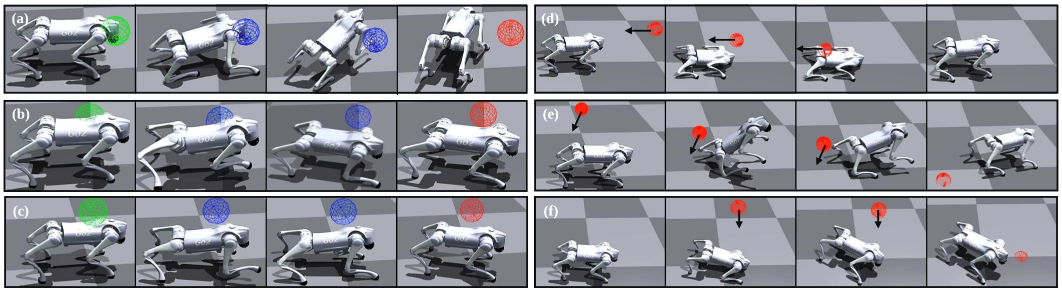
Tasks. We conducted simulation experiments in the Isaac gym [20] to evaluate the DOA ability of the REBot system. During testing, the obstacle approaches from diverse directions within a 180° arc in the XZ, YZ, and XY planes of the robot's body frame (Fig. 6), covering frontal, lateral, overhead, and ground-level threats. The response time is expanded beyond training, with $T_{\text{react}} \in [0.1, 4.0]$ s, allowing evaluation across both immediate reaction and delayed planning scenarios.

Metrics. The systems are evaluated with five metrics. The avoidance success rate (*ASR*): $N_{\text{avoid}}/N_{\text{total}}$; the recovery stability rate (*RSR*): $N_{\text{recover}}/N_{\text{avoid}}$, indicating the proportion of trials where the robot successfully stabilizes after avoidance; maximum joint power (*MJP*); avoidance moving distance (*AMD*): the base displacement between the robot's initial and final positions; and gait diversity index (*GDI*): $\mathbb{E}_{s^R \sim \mathcal{D}(s^R)} \left[\mathrm{Var}_{a^R \sim \pi(a^R|s^R)} [a^R] \right]$ the expected action variance under the learned policy, where $\pi(a^R|s^R)$ denotes the policy distribution over actions at state s^R, and $\mathcal{D}(s^R)$ is the state distribution collected during execution.

Baselines. 1) Agile But Safe (ABS) [9] achieved robust static obstacle avoidance with high-speed navigation motions, without the capability for dynamic obstacles. 2) Reactive RL (RRL) [24] is developed for dynamic obstacle avoidance in the UAV system. The avoidance strategy is based on simplified rigid-body dynamics, which do not generalize to legged whole-body systems.

4.2 Main Experimental Results

The REBot system is trained in static obstacle and dynamic obstacle avoidance curricula. Fig. 4 visualizes the simulation experimental results with different obstacle conditions and avoidance behaviors. When obstacles appear in front or on the sides, the robot tends to jump away for evasion (Fig. 4(a)(b)(e)). When obstacles appear on top, the robot will crouch down (Fig. 4(c)(d)(f)). These



Curriculum 1: static obstacle

Curriculum 2: moving obstacle

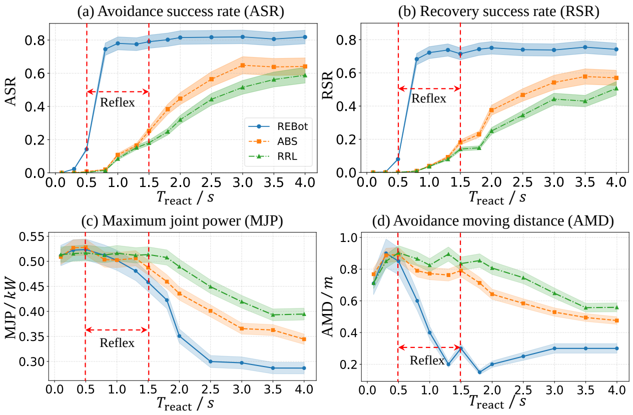
Figure 4: Illustration of simulation experiments. **Curriculum 1**: Static obstacle appears with a delayed time. 🟢 robot in normal stage; 🔵 robot avoiding the obstacle; 🔴 the obstacle appears and the robot evades the forbidden region. **Curriculum 2**: The robot avoids the red fast-moving obstacle at all times.

Table 1: Simulation Experiment Results

T_{react} / s	Metric	ABS$^{\diamond}$	RRL$^{\diamond}$	REbot
$0.1 \sim 0.5$	ASR$\uparrow^*$	0.00	0.00	**0.05**
	RSR$\uparrow^*$	0.00	0.00	**0.03**
	MJP$\downarrow^*$	0.51	0.52	**0.50**
	AMD$\downarrow^*$	0.84	0.85	**0.82**
$0.5 \sim 1.5$	ASR$\uparrow$	0.11	0.09	**0.65**
	RSR$\uparrow$	0.06	0.05	**0.59**
	MJP$\downarrow$	0.52	0.51	**0.49**
	AMD$\downarrow$	0.80	0.86	**0.47**
$1.5 \sim 4.0$	ASR$\uparrow$	0.51	0.41	**0.81**
	RSR$\uparrow$	0.42	0.32	**0.74**
	MJP$\downarrow$	0.40	0.45	**0.34**
	AMD$\downarrow$	0.60	0.70	**0.26**

[*] ASR: avoidance success rate; RSR: recovery success rate; MJP: maximum joint power; AMD: avoidance moving distance. [◇] ABS: Agile But Safe method [9]; RRL: Reactive RL policy [24].

Figure 5: Performances over Reaction Time



results demonstrate that REBot enables the robot to select appropriate avoidance strategies based on the obstacle state.

We categorize the reaction time range into three intervals (Tab. 1 and Fig. 5(a)(b)): 0.1 $\sim$ 0.5s, 0.5 $\sim$ 1.5s and 1.5 $\sim$ 4.0s. In the first interval, REBot and both baselines obtain nearly 0 success rates due to insufficient reaction time. In the second interval with moderately short reaction times, REBot exhibits reflexive evasion behaviors, while the two baselines still take low-speed gaits for navigation. This difference leads to much higher ASR and RSR of REBot compared to both baselines. In the third interval, REBot remains high ASR and RSR. ABS and RRL also improve their performance due to the longer reaction time, but still fall short of REBot because they are not specialized for active DOA. The overall trend, as illustrated in Fig 5, indicates that REBot effectively addresses instantaneous DOA challenges.

We also explore the relationship between MJP, AMD and reaction time (Fig. 5(c)(d)). When the reaction time is extremely short (e.g., below 0.5s), we observe a high MJP over 500 W and a large AMD, caused by intense evasion behaviors such as jumping away. In the moderately short reaction time interval, both MJP and AMD decrease, where REBot can adopt more appropriate avoidance behaviors such as crouching down. With longer reaction time, both MJP and AMD converge to lower values, as REBot has enough time to execute smoother navigation-based avoidance behaviors. The trends show that REBot adapts avoidance behaviors based on time constraints and avoidance efficiency.

4.3 Analysis of Avoidance Ability

We evaluate the effect of the obstacle direction on the robot's avoidance ability by applying impacts from different angles within the X-Z, Y-Z and X-Y planes of the robot's body frame (Fig. 6). Based on ASR and MJD, we divide the robot's avoidance behavior space into three regions: region I, where the robot fails to avoid; region II, where the robot adopts reflexive evasion; and region III, where the

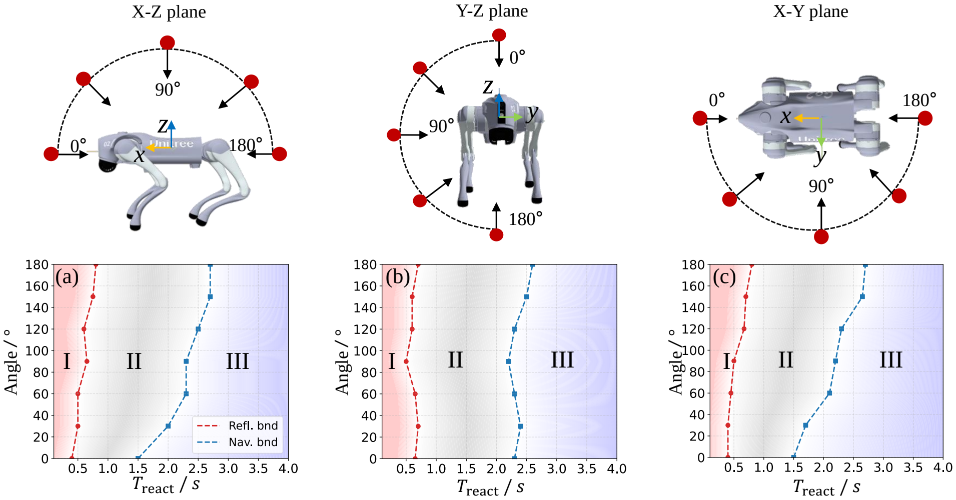


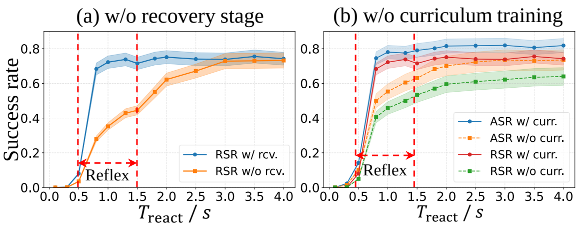
Figure 6: Robot avoids obstacles in various directions and reaction time. **Top row** shows the obstacles' approaching directions in three planes. **Bottom row** figures indicate the different avoidance behaviors. Region I: avoidance failure, II: reflexive evasion, III: navigation avoidance.

Table 2: Ablation Performances of REBot

T_{react} / s	Metric	w/o rcv.[1]	w/o curr.[2]	w/o adp.[3]	REBot
0.5∼1.5	ASR↑*	0.63	0.48	0.59	**0.65**
	RSR↑*	0.31	0.39	0.51	**0.59**
	GDI↑*	2.46	2.41	1.43	**2.51**
1.5∼4.0	ASR	0.80	0.71	0.78	**0.81**
	RSR	0.63	0.60	0.69	**0.74**
	GDI	2.06	2.24	1.36	**2.13**

* ASR: avoidance success rate; RSR: recovery success rate; GDI: gait diversity index. [1] w/o recovery stage; [2] w/o curriculum one learning; [3] w/o adaptive reward in avoidance policy.

Figure 7: Success Rate of Ablation Studies



robot adopts navigation-based avoidance. The boundary between region I and II is defined by ASR over 30%, while the boundary between region II and III is defined by MJD below 300 W.

In the X-Z and X-Y planes (Fig. 6(a)(c)), we observe that obstacles appearing in front of the robot are easier to avoid, and navigation-based avoidance can be achieved with shorter reaction time. In contrast, obstacles approaching from the back require longer reaction time and make navigation-based avoidance more difficult. This asymmetry is attributed to the mechanical design of Unitree Go2, where the leg structure facilitates faster backward motion but makes forward jumping more challenging. In the Y-Z plane (Fig. 6(b)), we find that obstacles approaching from the sides are easier to avoid compared to those from the top or bottom, with a shorter transition time from reflexive evasion to navigation-based avoidance. These results highlight the robot's avoidance capability varies significantly depending on the obstacle direction.

4.4 Ablation Studies of REBot System

The recovery stage ensures a stable standing posture after rapid reflexive evasion. To validate its effectiveness, we compare the performance of the REBot system with and without the recovery policy. As shown in Table 2 and Fig. 7(a), removing the recovery stage leads to a drop (20%) in the success rate within the reflex region. Additionally, as reaction time increases, avoidance behaviors shift from reflex to navigation, reducing the influence of the recovery stage on robot stabilization.

Curriculum learning enables a smooth transition from normal stage to fast-moving reflexive evasion. In the first stage, the robot learns to avoid obstacles that appear suddenly at varying positions; in the second stage, it generalizes to obstacles approaching from different directions. An ablation study reveals the importance of this progressive training: when policies are trained directly on the second curriculum (bypassing the first), Table 2 and Fig. 7(b) show removing the first curriculum causes a 5% decline in both ASR and RSR for both reflexive evasion and navigation avoidance. This performance gap stems from the more moderate and diverse gaits learned during static obstacle avoidance, which prove beneficial for a stabilized start when facing fast-moving obstacles.

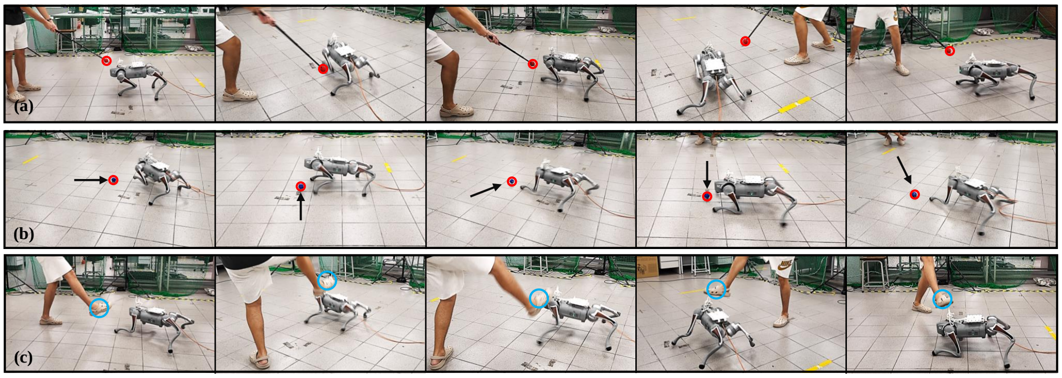


Figure 8: REBot system real-robot demonstrations on Unitree Go2 Robot. (See video)

The adaptive reward encourages diversified avoidance gaits that improve avoidance robustness. Without this term, the reinforcement learning algorithm converges to a single avoidance behavior (e.g., consistently jumping backward) regardless of obstacle variations. We conduct an ablation study to evaluate the adaptive reward's effectiveness. As shown in Tab. 2, removing the adaptive reward results in: (1) a significant 40% decrease in gait diversity index (GDI), demonstrating reduced behavioral diversity, and (2) moderate declines in both ASR and RSR. These findings indicate that the gait diversity promoted by the adaptive reward contributes directly to improved avoidance performance.

5 Real-Robot Demonstration

We deploy the REBot system on a Unitree Go2 robot to demonstrate its reflexive evasion capabilities in real-world scenarios. The robot and obstacles are tracked using an OptiTrack motion capture system to provide accurate position information. To generate diverse dynamic obstacles, we test three interaction types: poking with a stick (Fig. 8(a)), throwing a ball (Fig. 8(b)), and kicking (Fig. 8(c)), each targeting the robot from different directions, including front, left, right, left-front, and right-front.

When an obstacle approaches, the robot triggers the avoidance mode to perform reflexive evasion. The primary avoidance actions include jumping away from the obstacle and crouching down. After completing the avoidance maneuver, the robot switches to the recovery mode to regain a stable standing posture. Additionally, we observe that when the poking motion is relatively slow, the robot tends to adopt navigation-based avoidance strategies instead of reflexive actions, leveraging the longer available reaction time to perform smoother behaviors.Under the real-world test conditions, the REBot system achieves an ASR of 56% and an RSR of 53%. The performance gap compared to simulation is mainly attributed to Sim2Real challenges such as unmodeled actuator dynamics, latency in control execution, and surface friction variability, which particularly affect fast reflexive responses requiring precise torque delivery.

6 Conclusion

We initiate the study of reflexive evasion as a critical capability for dynamic obstacle avoidance in quadrupedal robots, where traditional navigation-based methods fall short under tight reaction constraints. To address this challenge, we develop REBot, a unified control system that couples rapid avoidance and stability recovery through reinforcement learning and structured training strategies. Extensive experiments in simulation and on real hardware confirm REBot's ability to perform reliable, adaptive evasive maneuvers, while revealing important characteristics of reflexive responses shaped by robot morphology and obstacle dynamics. Our results point toward new directions for building more agile, resilient, and safety-aware legged robotic systems in dynamic environments.

Limitation

The REBot system now has three limitations.

1. While we focus on the reflexive evasion policy, we leave precise obstacle position perception as an assumption. The real robot relies on the motion capture system to sense the obstacles, which provides accurate centimeter-level position observation and a 10-millisecond-level delay. We are actively implementing the ego-centric observation system via RGBD cameras and Lidar, which enables independent measuring and avoidance.
2. The Unitree Go2 robots have limitations in certain avoiding directions. As observed in both simulation and real-robot experiments, the REBot prefers to avoid by jumping backwards rather than forwards. Even if the obstacles approach the hip of the quadruped robot, the REBot system frequently drives the robot to move backwards. This might be because of the hardware configuration of the Go2 robot. Since the elbow's direction of all four legs is backwards, the robot is inherently suitable to jump backward, especially in high-speed reflex evasion. We will conduct more in-depth studies on such perspective of reflexive behaviors of quadrupedal robots.
3. An important Sim2Real gap in the REBot system is the servo motor control. In the simulation, the actions are the joints' angular velocity, while the direct control of the servo motors is the electric current and torque. Although the servo can internally address this gap, it is indeed influential in the very high-speed reflexive behaviors happening within 1 second, especially when the motors start from zero speed. In the future, we will study better preparation states for reflexive avoidance (e.g., active trot).

References

[1] F. Shi, C. Zhang, T. Miki, J. Lee, M. Hutter, and S. Coros. Rethinking robustness assessment: Adversarial attacks on learning-based quadrupedal locomotion controllers. *arXiv preprint arXiv:2405.12424*, 2024.

[2] Y. Tao, M. Li, X. Cao, and P. Lu. Mobile robot collision avoidance based on deep reinforcement learning with motion constraints. *IEEE Transactions on Intelligent Vehicles*, 2024.

[3] Y. Sun, R. Chen, K. S. Yun, Y. Fang, S. Jung, F. Li, B. Li, W. Zhao, and C. Liu. Spark: A modular benchmark for humanoid robot safety, 2025. URL https://arxiv.org/abs/2502.03132.

[4] D. Falanga, K. Kleber, and D. Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40):eaaz9712, 2020.

[5] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter. Learning a state representation and navigation in cluttered and dynamic environments. *IEEE Robotics and Automation Letters*, 6 (3):5081–5088, 2021.

[6] T. Dudzik, M. Chignoli, G. Bledt, B. Lim, A. Miller, D. Kim, and S. Kim. Robust autonomous navigation of a small-scale quadruped robot in real-world environments. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3664–3671. IEEE, 2020.

[7] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. *arXiv preprint arXiv:2107.03996*, 2021.

[8] X. Huang, Z. Li, Y. Xiang, Y. Ni, Y. Chi, Y. Li, L. Yang, X. B. Peng, and K. Sreenath. Creating a dynamic quadrupedal robotic goalkeeper with reinforcement learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2715–2722. IEEE, 2023.

[9] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi. Agile but safe: Learning collision-free high-speed legged locomotion. *arXiv preprint arXiv:2401.17583*, 2024.

[10] M. Lu, X. Fan, H. Chen, and P. Lu. Fapp: Fast and adaptive perception and planning for uavs in dynamic cluttered environments. *IEEE Transactions on Robotics*, 2024.

[11] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Robust and versatile bipedal jumping control through reinforcement learning. *arXiv preprint arXiv:2302.09450*, 2023.

[12] T. Umeda, O. Yokoyama, M. Suzuki, M. Kaneshige, T. Isa, and Y. Nishimura. Future spinal reflex is embedded in primary motor cortex output. *Science Advances*, 10(51):eadq4194, 2024.

[13] M. Liu, J. Xiao, and Z. Li. Deployment of whole-body locomotion and manipulation algorithm based on nmpc onto unitree go2quadruped robot. In *2024 6th International Conference on Industrial Artificial Intelligence (IAI)*, pages 1–6. IEEE, 2024.

[14] F. Xiao, T. Chen, and Y. Li. Egocentric visual locomotion in a quadruped robot. In *Proceedings of the 2024 8th International Conference on Electronic Information Technology and Computer Engineering*, pages 172–177, 2024.

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[16] M. Gurram, P. K. Uttam, and S. S. Ohol. Reinforcement learning for quadrupedal locomotion: Current advancements and future perspectives. In *2025 9th International Conference on Mechanical Engineering and Robotics Research (ICMERR)*, pages 28–38. IEEE, 2025.

[17] Z. Xu, A. H. Raj, X. Xiao, and P. Stone. Dexterous legged locomotion in confined 3d spaces with reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11474–11480. IEEE, 2024.

[18] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[19] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[20] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[21] X. Han and M. Zhao. Learning quadrupedal high-speed running on uneven terrain. *Biomimetics*, 9(1):37, 2024.

[22] Y. Zhao, T. Wu, Y. Zhu, X. Lu, J. Wang, H. Bou-Ammar, X. Zhang, and P. Du. Zsl-rppo: Zero-shot learning for quadrupedal locomotion in challenging terrains using recurrent proximal policy optimization. *arXiv preprint arXiv:2403.01928*, 2024.

[23] J. W. Mock and S. S. Muknahallipatna. A comparison of ppo, td3 and sac reinforcement algorithms for quadruped walking gait generation. *Journal of Intelligent Learning Systems and Applications*, 15(1):36–56, 2023.

[24] X. Fan, M. Lu, B. Xu, and P. Lu. Flying in highly dynamic environments with end-to-end learning approach. *IEEE Robotics and Automation Letters*, 10(4):3851–3858, Apr. 2025. ISSN 2377-3774. doi:10.1109/lra.2025.3547306. URL http://dx.doi.org/10.1109/LRA.2025.3547306.

[25] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, et al. Cerberus in the darpa subterranean challenge. *Science Robotics*, 7(66):eabp9742, 2022.

[26] S.-j. Sun, S.-h. Jiang, S.-h. Cui, Y. Kang, and Y.-t. Chen. Path planning of forest fire-fighting robots based on deep learning. 2020.

[27] X. Meng, W. Liu, L. Tang, Z. Lu, H. Lin, and J. Fang. Trot gait stability control of small quadruped robot based on mpc and zmp methods. *Processes*, 11(1):252, 2023.

[28] J. Chen, K. Xu, and X. Ding. Adaptive gait planning for quadruped robot based on center of inertia over rough terrain. *Biomimetic Intelligence and Robotics*, 2(1):100031, 2022.

[29] Z. Zhou, B. Wingo, N. Boyd, S. Hutchinson, and Y. Zhao. Momentum-aware trajectory optimization and control for agile quadrupedal locomotion. *IEEE Robotics and Automation Letters*, 7(3):7755–7762, 2022.

[30] W. Chi, X. Jiang, and Y. Zheng. A linearization of centroidal dynamics for the model-predictive control of quadruped robots. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4656–4663. IEEE, 2022.

[31] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis. Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control. *IEEE Transactions on Robotics*, 38(5):2908–2927, 2022.

[32] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. sci. *Robotics*, 4:26, 2019.

[33] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park. Representation-free model predictive control for dynamic motions in quadrupeds. *IEEE Transactions on Robotics*, 37(4):1154–1171, 2021.

[34] X. Chang, H. Ma, and H. An. Quadruped robot control through model predictive control with pd compensator. *International Journal of Control, Automation and Systems*, 19(11):3776–3784, 2021.

[35] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1–9. IEEE, 2018.

[36] J.-R. Chiu, J.-P. Sleiman, M. Mittal, F. Farshidian, and M. Hutter. A collision-free mpc for whole-body dynamic locomotion and manipulation. In *2022 international conference on robotics and automation (ICRA)*, pages 4686–4693. IEEE, 2022.

[37] J. Wu, Y. Xue, and C. Qi. Learning multiple gaits within latent space for quadruped robots. *arXiv preprint arXiv:2308.03014*, 2023.

[38] Z. Luo, Y. Dong, X. Li, R. Huang, Z. Shu, E. Xiao, and P. Lu. Moral: Learning morphologically adaptive locomotion controller for quadrupedal robots on challenging terrains. *IEEE Robotics and Automation Letters*, 2024.

[39] X. Zhang, Z. Xiao, Q. Zhang, and W. Pan. Synloco: Synthesizing central pattern generator and reinforcement learning for quadruped locomotion. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 2640–2645. IEEE, 2024.

[40] L. Smith, I. Kostrikov, and S. Levine. A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *arXiv preprint arXiv:2208.07860*, 2022.

[41] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research*, 43(4):572–587, 2024.

[42] Y. Song, S. Kim, and D. Scaramuzza. Learning quadruped locomotion using differentiable simulation. *arXiv preprint arXiv:2403.14864*, 2024.

[43] J. Bagajo, C. Schwarke, V. Klemm, I. Georgiev, J.-P. Sleiman, J. Tordesillas, A. Garg, and M. Hutter. Diffsim2real: Deploying quadrupedal locomotion policies purely trained in differentiable simulation. *arXiv preprint arXiv:2411.02189*, 2024.

[44] R. Azzam, M. Chehadeh, O. A. Hay, M. A. Humais, I. Boiko, and Y. Zweiri. Learning-based navigation and collision avoidance through reinforcement for uavs. *IEEE Transactions on Aerospace and Electronic Systems*, 60(3):2614–2628, 2023.

[45] Y. Wang, X. Li, J. Zhang, S. Li, Z. Xu, and X. Zhou. Review of wheeled mobile robot collision avoidance under unknown environment. *Science Progress*, 104(3):00368504211037771, 2021.

[46] D. Kim, M. Srouji, C. Chen, and J. Zhang. Armor: Egocentric perception for humanoid robot collision avoidance and motion planning. *arXiv preprint arXiv:2412.00396*, 2024.

[47] K.-T. Song and C.-H. Lin. Mpc-based optimization design for 3d collision avoidance of a mobile manipulator based-on obstacle velocity estimation. In *2024 International Automatic Control Conference (CACS)*, pages 1–6. IEEE, 2024.

[48] Y. Zhang, T. Liang, Z. Chen, Y. Ze, and H. Xu. Catch it! learning to catch in flight with mobile dexterous hands. *arXiv preprint arXiv:2409.10319*, 2024.

[49] R. Ramadan, H. Geyer, J. Jeka, G. Schöner, and H. Reimann. A neuromuscular model of human locomotion combines spinal reflex circuits with voluntary movements. *Scientific Reports*, 12(1):8189, 2022.

[50] T. S. Pulverenti, M. Zaaya, M. Grabowski, E. Grabowski, M. A. Islam, J. Li, L. M. Murray, and M. Knikou. Neurophysiological changes after paired brain and spinal cord stimulation coupled with locomotor training in human spinal cord injury. *Frontiers in Neurology*, 12:627975, 2021.

A Related Works

A.1 Quadrupedal Robot Locomotion

Quadrupedal robots have achieved significant breakthroughs over the past decade [25, 26], enabled by advances in both model-based and learning-based control frameworks. Model-based methods, such as zero-moment point (ZMP) planning [27, 28], centroidal dynamics optimization [29, 30], and Model Predictive Control (MPC) [31, 32], allow precise trajectory tracking and robust locomotion over structured terrains. By leveraging accurate physical models, these approaches enable real-time foot placement adjustments and stability control [33, 34], supporting dynamic maneuvers like trotting, galloping, and bounding under known conditions.

In parallel, reinforcement learning has emerged as a key enabler for quadrupedal locomotion [16], demonstrating impressive adaptability across diverse environments and tasks [35, 36]. RL-trained policies have successfully produced a wide range of gaits [37], robustly traversed rough terrains [38], and even adapted to changes in morphology or sensory conditions [39]. By learning directly from trial-and-error interactions, these methods can capture complex, nonlinear locomotion behaviors that are difficult to design analytically [40, 41]. Several works have also explored RL-based static obstacle avoidance, integrating navigation strategies to enable quadrupeds to plan collision-free paths around known obstacles [9].

Recent developments in differentiable simulation (DiffSim) [42] further enhance locomotion research by providing sample-efficient learning through differentiable physics engines [43]. While promising, most applications of DiffSim currently focus on improving locomotion in structured or semi-structured environments.

A.2 Dynamic Obstacle Avoidance

Dynamic obstacle avoidance has been actively explored across UAVs [44], mobile robots [45], humanoid robots [46] and manipulators [47], each leveraging platform-specific capabilities to achieve rapid and adaptive reactions. For UAVs, Falanga et al. [4] combined event cameras with model predictive control to enable fast evasive maneuvers in cluttered, dynamic environments. Lu et al. [10] proposed a fast and adaptive perception-planning framework that integrates global and local maps for high-frequency collision avoidance, while Fan et al. [24] introduced a lidar-driven deep rl system capable of avoiding highly dynamic obstacles without explicit mapping, even at high relative speeds.

For mobile robots, Tao et al. [2] developed a deep rl framework with embedded motion constraints, enabling wheeled robots to navigate dynamic, dense environments using onboard RGB-D cameras, outperforming traditional velocity obstacle or MPC-based methods. For humanoid robots and manipulators, Zhang et al. [48] designed a rl approach that coordinates base and arm motion to intercept and avoid moving objects, while the SPARK benchmark [3] introduced a modular evaluation framework for assessing the generalization and robustness of humanoid locomotion controllers under diverse tasks and disturbances.

Despite these advances, quadrupedal robots currently lack general-purpose frameworks for dynamic obstacle avoidance. Existing approaches often rely on predefined motion primitives such as sidestepping or jumping, without the adaptive, reflexive strategies seen in other platforms. Bridging this gap remains an open challenge and is critical for deploying legged robots in fast-changing environments.

A.3 Reflexive Neural System

In biological systems, reflexes are rapid, involuntary responses triggered by local sensory inputs, allowing animals to react instantly to sudden stimuli without engaging central decision-making pathways [49]. These reflexive pathways, often described as part of a master-slave system, assign the "slave" or subordinate system to handle immediate, protective reactions—such as limb withdrawal or postural correction—while the "master" system, typically the brain, focuses on slower, deliberate processing and higher-order decision-making. This hierarchical division ensures survival under

Table 3: Summary of PPO hyperparameters used for training

Hyperparameter	Value
Actor hidden layers	[512, 256, 128]
Critic hidden layers	[512, 256, 128]
Activation	ELU
Learning rate (α)	1×10^{-3}
Clip parameter (ϵ)	0.2
Value loss coefficient (c_1)	1.0
Entropy coefficient (c_2)	0.01
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
Desired KL divergence	0.01
Max gradient norm	1.0
Steps per env per iter	24
Mini-batches per iter	4
Learning epochs per iter	5
Max iterations	5000
Parallel environments	4096

unexpected disturbances [50], enabling organisms to balance fast, reactive control with complex, goal-directed behaviors.

B Experiment Details

B.1 RL Policy Training Details

We trained two separate RL policies using PPO: an avoidance policy focused on dynamic obstacle evasion and a recovery policy responsible for post-disturbance stabilization. Both policies used actor-critic architectures implemented as multilayer perceptrons (MLPs) with hidden layers of 512, 256, and 128 units, using ELU activations.

For optimization, we applied a clipped surrogate objective with a clip parameter $\epsilon = 0.2$ (Tab. 3), an entropy coefficient $c_2 = 0.01$, and a value loss coefficient $c_1 = 1.0$. We used the Adam optimizer with an initial learning rate $\alpha = 1 \times 10^{-3}$, combined with adaptive learning rate scheduling. The discount factor was set to $\gamma = 0.99$, and the generalized advantage estimation (GAE) parameter was set to $\lambda = 0.95$. Each PPO iteration collected 24 steps per environment across 4096 parallel environments, followed by 5 learning epochs over 4 mini-batches. The maximum gradient norm was clipped at 1.0, and the desired KL divergence threshold was set to 0.01. Training was conducted using Isaac Gym on an NVIDIA RTX 4090 GPU.

Building on the PPO optimization framework, the avoidance policy is guided by a reward structure that combines three main components: avoidance rewards that encourage maintaining safe distances from dynamic obstacles and penalize collisions, regularization rewards that promote stable, symmetric, and energy-efficient gait patterns, and adaptive rewards that foster motion diversity, speed adaptation, and directional efficiency under varying threat levels. This combination ensures that the robot can execute timely evasive maneuvers while maintaining locomotion stability and natural gait coordination.

In parallel, the recovery policy employs a dedicated reward design focused on regaining upright posture, minimizing joint velocities, returning to the nominal base position, and ensuring smooth, low-torque recovery transitions after disturbances. This structure enables the robot to rapidly restore balance and seamlessly transition back to its default locomotion behaviors after a disturbance or evasive event.

Table 4: Summary of auxiliary regularization terms

Term	Purpose
$\|v_t^{R,z}\|^2$	Penalize vertical velocity
$\|\|\dot{\theta}_t^{R,xy}\|\|$	Penalize horizontal angular velocity
$\|\|\theta_t^{R,xy}\|\|$	Penalize non-flat orientation
$\sum_i (a_t^{R,i} - a_{t-1}^{R,i})^2$	Penalize abrupt action changes
$\sum_i \mathbf{1}_c^i$	Penalize body collisions
$\|\|v_t^{R,xy} - v_t^{R,xy,\text{cmd}}\|\|$	Track command linear velocity
$\|\omega_t^{R,z} - \omega_t^{R,z,\text{cmd}}\|$	Track command angular velocity
$\sum_i \left(t_{\text{air}}^i - 0.5\right)$	Reward long foot swing phases
$\mathbf{1}\left(\max_i \left(\left\\|\mathbf{f}_t^{R,xy,i}\right\\| / \left\|f_t^{R,z,i}\right\|\right) > 5\right)$	Penalize stumbling events
$\sum_i \\|f_t^{R,i} - f_{\text{th}}^{R,i}\\|^2$	Penalize excessive foot contact forces

Table 5: Domain Randomization Settings for Policy Training

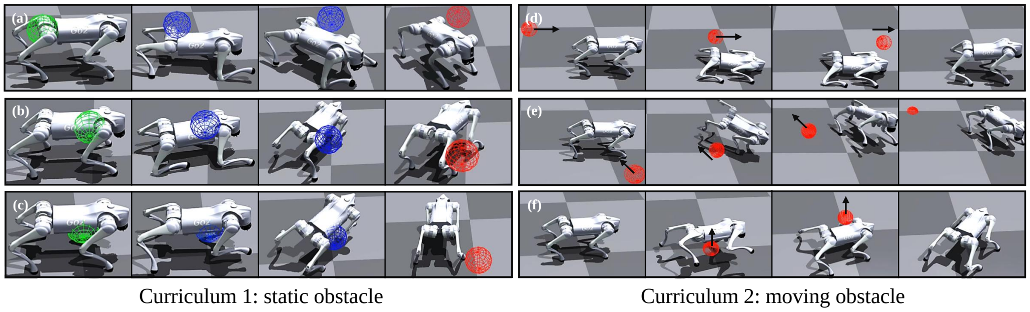
Term	Value
Observation	
Joint position noise	$\mathcal{U}(-0.01, 0.01)$ rad
Joint velocity noise	$\mathcal{U}(-1.5, 1.5)$ rad/s
Angular velocity noise	$\mathcal{U}(-0.2, 0.2)$ rad/s
Projected gravity noise	$\mathcal{U}(-0.05, 0.05)\ \mathrm{m/s^2}$
Height measure noise	$\mathcal{U}(-0.1, 0.1)$ m
Dynamics	
Friction factor	$\mathcal{U}(0.5, 1.25)$
Added base mass	$\mathcal{U}(-1.0, 1.0)$ kg
Obstacle position (per axis)	$\mathcal{U}(-0.4, 0.4)$ m
Obstacle radius	$\mathcal{U}(0.05, 0.3)$ m
Obstacle velocity	$\mathcal{U}(1.0, 6.0)$ m/s
Reaction time	$\mathcal{U}(0.1, 4.0)$ s
Episode	
Episode length	$\mathcal{U}(8.0, 10.0)$ s
Command robot yaw	$\mathcal{U}(-1.0, 1.0)$ rad
Command robot velocity	$\mathcal{U}(-1.0, 1.0)$ m/s
Command robot heading	$\mathcal{U}(-\pi, \pi)$ rad

In addition to these task-specific rewards, we incorporate a set of auxiliary regularization terms to enforce physical plausibility, smoothness, and mechanical safety. These terms play a critical role in constraining the robot's low-level dynamics, preventing unrealistic or unsafe behaviors, and improving the overall robustness and hardware transferability of the learned policies. The complete set of these auxiliary terms is summarized in Tab. 4.

B.2 Domain Randomization & Curricula

To enhance policy generalization and robustness, we applied domain randomization and a staged curriculum strategy during training. Domain randomization (Tab. 5) introduces variability across observation parameters (e.g., joint position and velocity noise), dynamics parameters (e.g., ground friction, added base mass, and obstacle velocity), and episode-level parameters (e.g., commanded yaw and episode duration). By sampling these parameters uniformly within predefined ranges at the start of each episode, the policy is exposed to a diverse set of conditions, improving its ability to handle modeling uncertainties, mitigate overfitting to narrow simulation settings, and transfer reliably to real-world deployment.

The curriculum learning strategy is implemented to gradually increase task complexity. Initially, the policy learns to avoid static obstacles that suddenly appear at specific positions near the robot, without any external perturbations. In the next stage, dynamic obstacles with varying speeds and



Curriculum 1: static obstacle

Curriculum 2: moving obstacle

Figure 9: Additional results of simulation experiments. (a) and (d) show the obstacle hits from the back; (b) and (e) hit from the right; (c) and (f) hit from the bottom.

trajectories are introduced, requiring the robot to perform rapid, adaptive avoidance maneuvers. Finally, disturbances and environmental uncertainties are incorporated to ensure the policy remains stable and robust under real-world deployment conditions. Additional experimental results are provided in Fig. 9.

B.3 Real-Robot Experiment Settings

To evaluate the real-world feasibility of the proposed avoidance-recovery strategy, experiments were conducted on a Unitree Go2 quadrupedal robot. Prior to deployment, the learned policies were validated through sim-to-sim transfer from Isaac Gym to MuJoCo to ensure robustness under a higher-fidelity simulation environment (Fig.10). Only after passing these intermediate robustness tests were the policies transferred to the real robot.

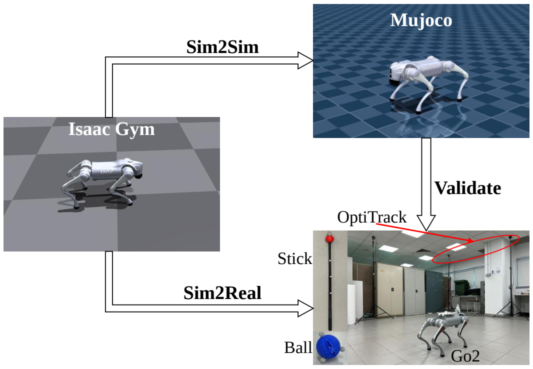


Figure 10: Transfer pathways from Isaac Gym to MuJoCo (sim2sim) and to the real robot (sim2real). Real-robot experiments involve dynamic obstacles including a ball, a stick, and a human foot, representing diverse scenarios.

For the real-robot setup, we employed an OptiTrack motion capture system to provide precise localization of both the robot and the dynamic obstacles (Fig. 10). The system offers sub-millimeter positioning accuracy, enabling high-fidelity state tracking without relying on onboard perception (e.g., cameras or LiDAR). This design ensures that the robot directly receives ground truth position and velocity information for both itself and the obstacles at each control step.

To simulate various dynamic threats, we used three types of physical obstacles: a rigid ball, a stick with a marker-attached tip, and a human foot. Marker points were affixed to each obstacle, allowing their motion to be tracked and fed into the robot's control pipeline. During experiments, the Go2 executed the trained avoidance and recovery policies in real time, responding to incoming obstacles by performing reflexive evasion and subsequent stabilization maneuvers.

B.4 Additional Experiment Results

We present additional qualitative results to illustrate the effectiveness and versatility of the proposed avoidance-recovery strategy across diverse scenarios. Figure 11 shows an example where the robot employs navigation-based avoidance strategies, relying on trajectory adjustment rather than reflexive maneuvers. This is feasible because the approaching obstacles are slow, providing sufficient reaction time for planned avoidance.

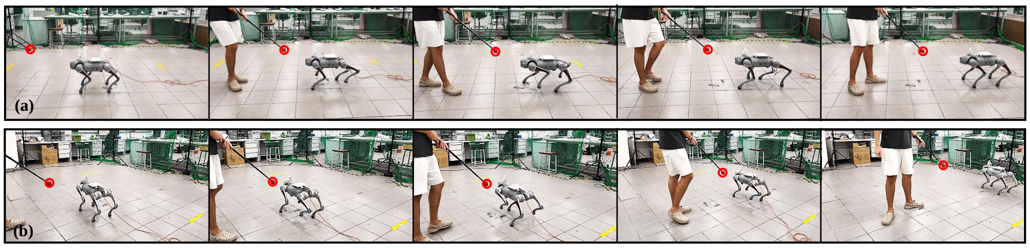


Figure 11: Navigation-based avoidance under slow-moving stick disturbances, providing sufficient reaction time. (a) Poking from the front; (b) poking from the left.

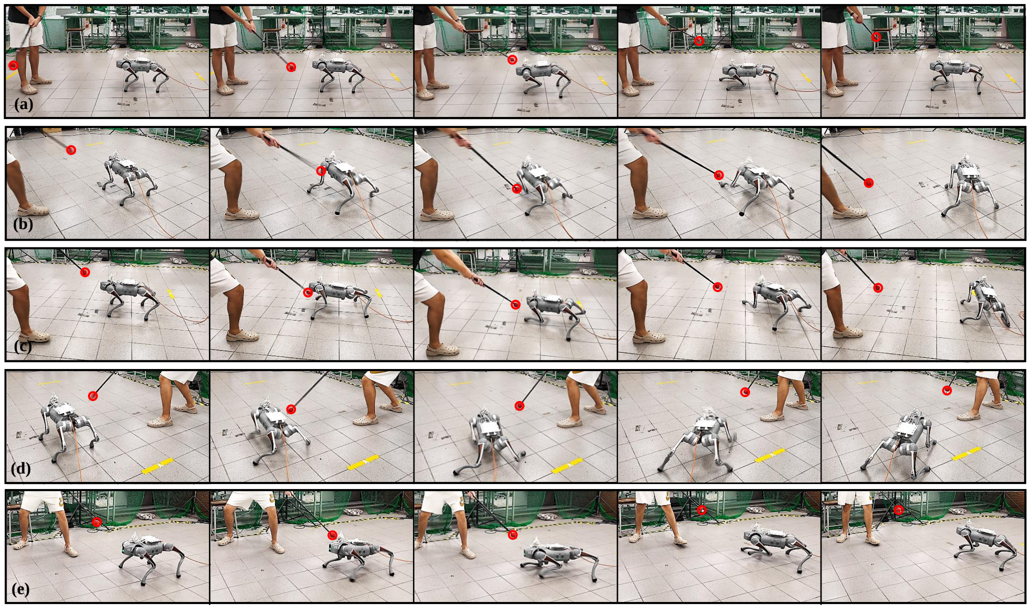


Figure 12: Reflexive evasion under fast stick disturbances with short reaction time. (a) From the front; (b) from the left; (c) from the left front; (d) from the right; (e) from the right front.

Figure 12 demonstrates the robot's reflexive response under stick-induced prodding attacks from various directions, showcasing its ability to rapidly adjust posture and evade external physical disturbances.

Figure 13 presents results where a ball is thrown toward the robot from multiple angles, testing the policy's capacity to execute fast evasive maneuvers under short reaction times.

Finally, Figure 14 highlights experiments where the robot faces unexpected kicks from a human foot at different approach angles, demonstrating the policy's robustness in handling unstructured, real-world disturbances.

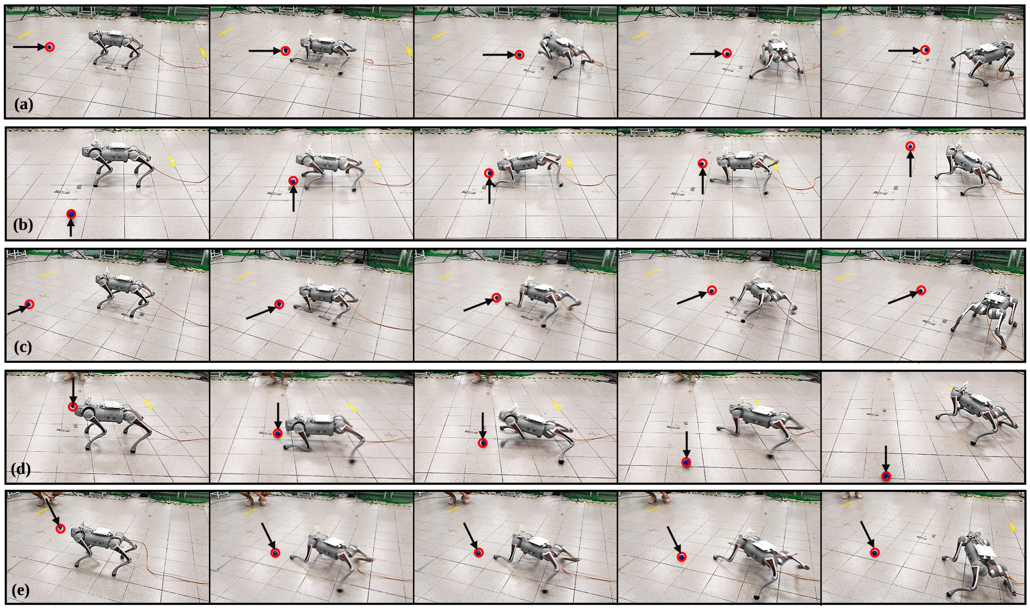


Figure 13: Reflexive evasion under ball-throw impacts from different directions. (a) From the front; (b) from the left; (c) from the left front; (d) from the right; (e) from the right front.

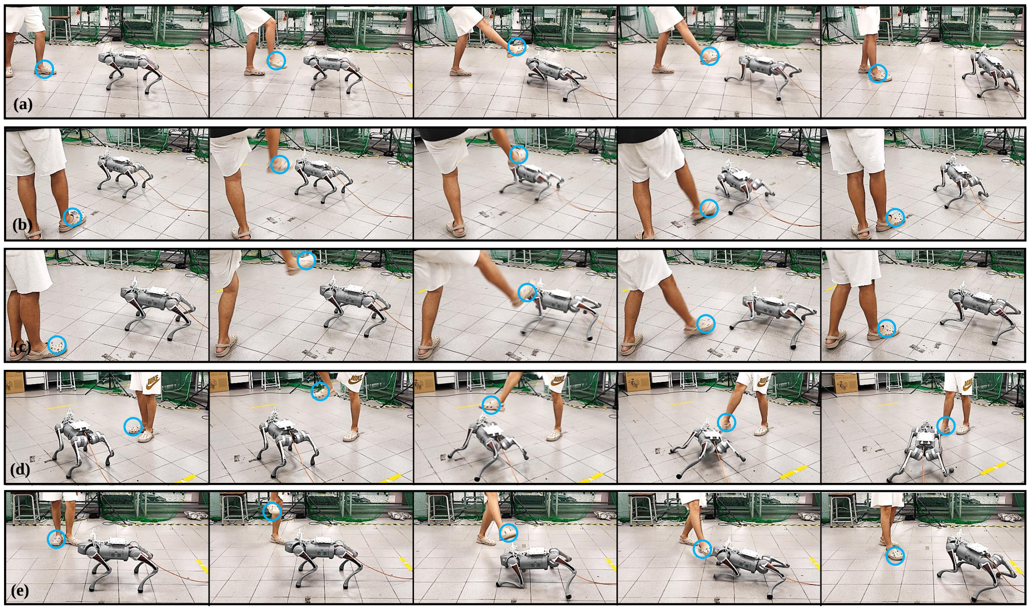


Figure 14: Reflexive evasion under foot-kick disturbances from different directions. (a) From the front; (b) from the left; (c) from the left front; (d) from the right; (e) from the right front.