

Learning Humanoid Arm Motion via Centroidal Momentum Regularized Multi-Agent Reinforcement Learning

Ho Jae Lee¹, Se Hwan Jeon¹, and Sangbae Kim¹

Abstract—Humans naturally swing their arms during locomotion to regulate whole-body dynamics, reduce angular momentum, and help maintain balance. Inspired by this principle, we present a limb-level multi-agent reinforcement learning (RL) framework that enables coordinated whole-body control of humanoid robots through emergent arm motion. Our approach employs separate actor-critic structures for the arms and legs, trained with centralized critics but decentralized actors that share only base states and centroidal angular momentum (CAM) observations, allowing each agent to specialize in task-relevant behaviors through modular reward design. The arm agent guided by CAM tracking and damping rewards promotes arm motions that reduce overall angular momentum and vertical ground reaction moments, contributing to improved balance during locomotion or under external perturbations. Comparative studies with single-agent and alternative multi-agent baselines further validate the effectiveness of our approach. Finally, we deploy the learned policy on the MIT Humanoid, achieving robust performance across diverse locomotion tasks, including flat-ground walking, rough terrain traversal, and stair climbing. (see project page)

Index Terms—Humanoid and Bipedal Locomotion, Reinforcement Learning, Whole-Body Motion Control, Physics-Guided Learning

I. INTRODUCTION

ARM swing is a natural and characteristic feature of human locomotion, but its fundamental role remains unclear. Biomechanical studies have offered a variety of explanations and consequences of arm swing during locomotion. Some argue that it reduces the induced angular momentum of the legs, ensuring smoother movement [1]. Consequently, the required ground reaction moments for locomotion are lower [2], improving energy economy of gait patterns [3]. For humanoid platforms, however, it is less clear how to effectively and intentionally coordinate the arms during locomotion, especially with typical optimization or learning-based control methods. Designing such controllers is difficult due to the complexities of whole-body dynamics, the variability of locomotion tasks, and uncertainty on how arm motion should be guided with respect to the task, stylistically or otherwise.

Prior work has typically approached controlling arm and leg motion in a unified manner. Imitation-based approaches try to match distributions of reference data, as in [4], [5], [6]. These controllers are able to exhibit human-like behavior, but it is

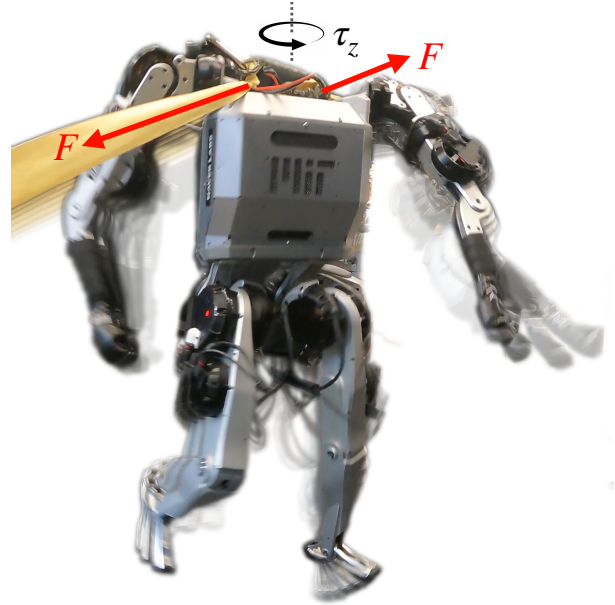


Fig. 1: Coordinated arm response under external torque disturbance. Two external forces F are applied from opposite sides to induce a positive yaw torque disturbance τ_z about the vertical axis. The robot reacts with emergent arm motions that counteract the induced momentum, helping to recover and maintain balance.

unclear how (or if) mimicking these references offers advantages for locomotion—the primary reward signal is to imitate trajectories from higher degree-of-freedom systems, which is not necessarily aligned with robustness or performance for the specific platform. With reinforcement learning (RL) methods, potentially conflicting rewards make it challenging to plan arm and leg motions simultaneously, and arm motion is often regularized with heuristics that may have little physical basis other than stylistic tuning [7]. This balance becomes even more sensitive with loco-manipulation, such as in [8], where different sets of limbs can be responsible for entirely different tasks.

Model-based controllers have typically leveraged *full-body* optimization schemes to generate arm motion. Via centroidal momentum tracking [9], [10], [11], relative angular-acceleration control [12], or full-body model predictive control (MPC) [13], [14] arm motion can arise naturally with a physically-consistent basis. However, solving these full-body motion plans becomes difficult to scale for online, closed-loop evaluation, often requiring approximations or special treatment for sufficient bandwidth. Additional considerations, such as avoiding self-collisions, state estimation noise, and environmental constraints, can further complicate these model-based controllers. While reasoning about the full dynamics of

Manuscript received July 5, 2025; Accepted September 5, 2025.

This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by NAVER Labs.

¹The authors are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: hjlee201@mit.edu; sehwan@mit.edu; sangbae@mit.edu).

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

the legs and arms is ideal, the computational expense can make these control schemes intractable.

In contrast to these whole-body approaches, there is evidence that the arms and legs are *not* planned jointly at the biological level. Commands for the arms and legs arise from entirely separate cortical areas, and are only coupled for stabilization elsewhere in the cerebellum [15], [16]. It is unclear whether this is simply vestigial. However, this separation in control can present several advantages from an engineering perspective. Optimizations for complex systems can be greatly accelerated by "sectioning" the physics of the model, such as the front and rear legs of a quadruped, and ensuring consensus downstream [17]. Similarly, locomotion can be trained separately from manipulation tasks by abstracting arm motion as random inertial wrenches on the body, as in [18]. This controller design enables the limbs to be reasoned about separately without mutual interference, allowing for coordinated motion without intractable computation or meticulous tuning.

Inspired by this architecture, we approach this challenge of coordinating locomotion with arm motion as a *multi-agent RL problem*—the arms and legs are treated as separate agents in the learning environment, connected only by the centroidal dynamics of the robot. Specifically, we adopt a centralized training and decentralized execution (CTDE) approach [19], [20], wherein decentralized actors for the arms and legs observe only their respective proprioceptive information, while centralized critics have access to global state information from both limb groups. This architecture enables faster convergence during training to a higher asymptotic task reward, and we analyze the effect of having centralized/decentralized observations for both the actor and critic.

Additionally, we introduce a centroidal angular momentum reward to guide arm motion, motivated by prior biomechanical studies of locomotion and physically-grounded principles. The resultant controller demonstrates natural arm swing while walking and improved recovery from external disturbances, which we validate on hardware for the MIT Humanoid. We analyze the ground reaction moments and momentum during execution and find they corroborate the findings from [1], [2], [3].

We summarize our main contributions as follows:

- 1) We introduce a CAM reward based on biomechanical studies of human walking, and find that it guides the emergence of natural arm swing for stable locomotion and effective push recovery for our policy.
- 2) We propose a multi-agent RL framework employing separate actor-critic networks for the arms and legs, trained centrally but executed in a decentralized manner.
- 3) We demonstrate the effectiveness and practicality of our controller by validating its performance on the MIT Humanoid, both in simulation and hardware experiments.

II. BACKGROUND

A. Centroidal Dynamics and Momentum

The centroidal dynamics of a system [10], [11], [13] offers a balanced representation of robot dynamics, bridging the gap

between simplified point-mass models and computationally expensive full-body dynamics. It is particularly valuable in humanoid control, as it captures whole-body interactions in a compact form, enabling emergent upper-body motion and improving robustness within a whole-body control framework. Formally, centroidal dynamics describes the linear and angular momentum dynamics around the robot's center of mass (CoM) as follows:

$$\dot{\mathbf{l}}_G = m\ddot{\mathbf{r}} = \sum_j \mathbf{F}_j + m\mathbf{g} \quad (1)$$

$$\dot{\mathbf{k}}_G = \sum_j (\mathbf{c}_j - \mathbf{r}) \times \mathbf{F}_j + \boldsymbol{\tau}_j \quad (2)$$

where m denotes the total mass of the robot, $\mathbf{r} \in \mathbb{R}^3$ is the CoM position, $\mathbf{F}_j \in \mathbb{R}^3$ and $\boldsymbol{\tau}_j \in \mathbb{R}^3$ are the external force and moment at contact point $\mathbf{c}_j \in \mathbb{R}^3$, respectively, and $\mathbf{g} \in \mathbb{R}^3$ represents gravitational acceleration. Additionally, $\mathbf{l}_G \in \mathbb{R}^3$ denotes the centroidal linear momentum (CLM), $\mathbf{k}_G \in \mathbb{R}^3$ denotes the centroidal angular momentum (CAM), and the subscript G indicates quantities expressed with respect to the CoM frame.

Centroidal momentum \mathbf{h}_G , comprising both CAM and CLM, is expressed compactly as:

$$\mathbf{h}_G(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \mathbf{k}_G \\ \mathbf{l}_G \end{bmatrix} = \mathbf{A}_G(\mathbf{q})\dot{\mathbf{q}} \quad (3)$$

where $\mathbf{A}_G \in \mathbb{R}^{6 \times (n+6)}$ is the centroidal momentum matrix (CMM), $\mathbf{q} \in \mathbb{R}^{(n+7)}$ and $\dot{\mathbf{q}} \in \mathbb{R}^{(n+6)}$ denote the generalized positions and velocities of the robot, respectively.

The generalized velocity vector $\dot{\mathbf{q}}$ can be partitioned as:

$$\dot{\mathbf{q}} = [\boldsymbol{\omega}_B^\top \quad \mathbf{v}_B^\top \quad \dot{\mathbf{q}}_{\text{legs}}^\top \quad \dot{\mathbf{q}}_{\text{arms}}^\top]^\top \quad (4)$$

where $\boldsymbol{\omega}_B \in \mathbb{R}^3$, $\mathbf{v}_B \in \mathbb{R}^3$ represent the body angular and linear velocities in base frame, respectively, and $\dot{\mathbf{q}}_{\text{legs}} \in \mathbb{R}^{n_{\text{legs}}}$ and $\dot{\mathbf{q}}_{\text{arms}} \in \mathbb{R}^{n_{\text{arms}}}$ are the joint velocities of legs and arms, respectively.

By time-differentiating Eq (3), we obtain the time-rate of change of centroidal momentum as:

$$\dot{\mathbf{h}}_G(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \dot{\mathbf{k}}_G \\ \dot{\mathbf{l}}_G \end{bmatrix} = \mathbf{A}_G(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{A}}_G(\mathbf{q})\dot{\mathbf{q}} \quad (5)$$

Since external disturbances directly affect centroidal momentum, we exploit this relationship for reward shaping: specifically, we incentivize arm policies to actively reduce the rate of change of CAM, thereby enhancing overall locomotion stability.

B. Multi-Agent Reinforcement Learning

Traditional policy gradient methods [21], [22] are poorly suited to multi-agent environments due to the inherent non-stationarity induced by simultaneously learning agents. As each agent's policy evolves during training, the environment appears non-stationary from the perspective of any individual agent, which undermines the convergence of standard RL algorithms. To address this, the Centralized Training with Decentralized Execution (CTDE) paradigm [19], [20] was

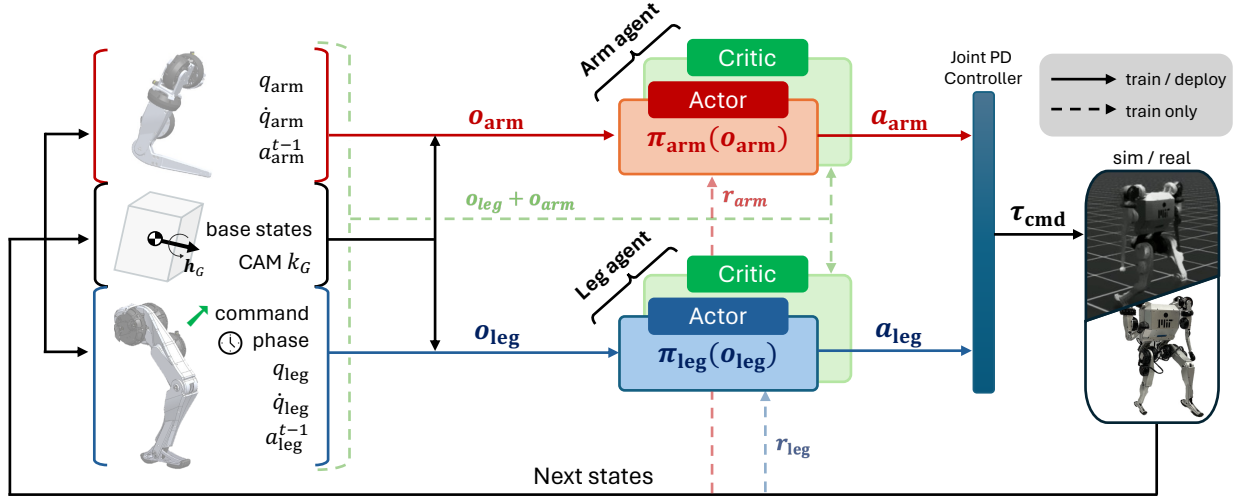


Fig. 2: Overview of our limb-level multi-agent reinforcement learning framework with CAM regularization. Separate actor-critic policies are assigned to the arm and leg agents, each operating on distinct observation spaces and reward structures. During training, each critic is provided global information for centralized training, while actors execute using local observations, sharing only base states and CAM. This design enables coordinated whole-body behavior, as the arm and leg agents are implicitly coupled through the shared CAM signal, allowing the arm to respond to momentum generated by leg motion.

introduced. Under CTDE, each agent maintains its own actor that operates solely on local observations, while the corresponding critic is trained with access to additional global information, such as the actions and observations of other agents. This centralized critic formulation stabilizes training by making the environment stationary with respect to each agent’s critic, even as policies change.

Formally, for any policy $\pi_i \neq \pi'_i$, the transition probability satisfies $P(s'|s, a_1, \dots, a_N, \pi_1, \dots, \pi_N) = P(s'|s, a_1, \dots, a_N) = P(s'|s, a_1, \dots, a_N, \pi'_1, \dots, \pi'_N)$ where s denotes the underlying environment state and a_i denotes the action taken by agent i . In the general multi-agent policy gradient formulation, the estimate of the policy gradient \hat{g} based on the advantage function takes the form:

$$\hat{g} = \mathbb{E}_t[\nabla_{\theta_i} \log \pi_i(a_i | o_i) \hat{A}_i^t(\mathbf{x}, a_1, \dots, a_N)] \quad (6)$$

where \hat{A}_i^t is the advantage estimate for agent i computed using the centralized critic. Here, \mathbf{x} denotes a global state representation (typically a concatenation of all agents’ observations) and o_i represents the local observation available to agent i .

This framework is also compatible with any actor-critic algorithm. A variant of Proximal Policy Optimization (PPO) [22] known as Multi-Agent PPO (MAPPO), has shown strong empirical performance in multi-agent settings [23].

III. CENTROIDAL MOMENTUM MODELING AND COMPUTATION

Our experimental platform is the MIT Humanoid [24] (Fig. 3), which weighs 24.89 kg in total, with the arms accounting for 5.71 kg—approximately 22.8% of the robot’s mass. The robot has 10 actuated joints for the legs and 8 for the arms. To compute dynamic quantities such as the CMM and its derivatives, we use `Pinocchio` [25], a modern rigid-body dynamics library that provides efficient algorithms for computing forward dynamics, kinematics, and momentum mappings. We implement a `casadi` [26] symbolic version of `Pinocchio`

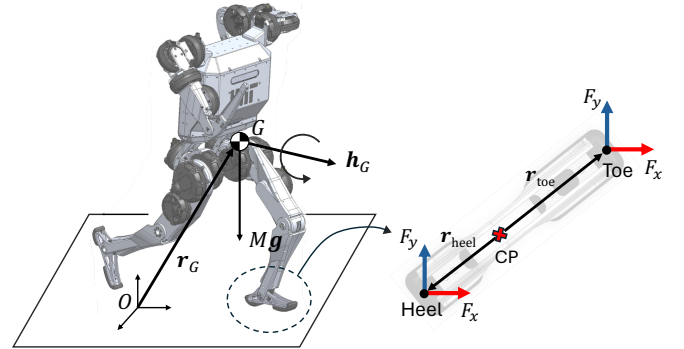


Fig. 3: *Left*: Illustration of the MIT Humanoid with indicated total centroidal momentum \mathbf{h}_G at the center of mass (G). *Right*: Top-down view of the foot with the cylindrical collision primitive, which permits only two contact points at the toe and heel. The corresponding horizontal contact forces F_x, F_y are depicted at each point. The vertical ground reaction moment (GRM) is then computed around the stance foot’s center of pressure (CP).

for analytical expression generation, and parallelize these expressions using `CusADi` [27], enabling high-throughput batched evaluation across multiple simulation environments.

We compute the total centroidal momentum and its time derivative using Eq. 3 and Eq. 5. To obtain the limb-wise contributions, we decompose $\mathbf{h}_G = \mathbf{h}_{G, \text{base}} + \mathbf{h}_{G, \text{legs}} + \mathbf{h}_{G, \text{arms}}$ into additive components from the base, legs, and arms. This is achieved by multiplying the relevant block of the CMM with the corresponding segment of the generalized velocity vector, effectively isolating each limb’s contribution to the total centroidal momentum.

We also define a reference centroidal momentum $\hat{\mathbf{h}}_G$ by projecting the desired base linear and angular velocities through the CMM evaluated at the current generalized position \mathbf{q} :

$$\hat{\mathbf{h}}_G(\mathbf{q}, \hat{\mathbf{q}}) = \begin{bmatrix} \hat{\mathbf{k}}_G \\ \hat{\mathbf{l}}_G \end{bmatrix} = \mathbf{A}_G(\mathbf{q}) \hat{\mathbf{q}} \quad (7)$$

where $\hat{\mathbf{q}} = [0 \ 0 \ \hat{\omega}_B^z \ \hat{v}_B^x \ \hat{v}_B^y \ 0 \ \mathbf{0}_\text{leg}^\top \ \mathbf{0}_\text{arm}^\top]^\top$ denotes

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

the reference generalized velocity embedding the user command $\hat{v}_B = (\hat{v}_B^x, \hat{v}_B^y, \hat{\omega}_B^z)$, corresponding to the desired base linear velocity in the x and y directions, and angular velocity around the z -axis. This reference serves as a target momentum profile in our reward shaping, encouraging the emergence of natural arm motions that contribute to coordinated whole-body locomotion.

IV. LIMB-LEVEL MULTI-AGENT REINFORCEMENT LEARNING FRAMEWORK

Taking inspiration from the separation of control present in the cortex, we approach learning whole-body control as a multi-agent RL problem, and train policies that separately control the arms and legs of the humanoid system (Fig. 2). By assigning distinct agents for the legs and arms of the platform, we can tailor observations and rewards for these subsystems individually. This ensures that policy gradient updates from the leg agent do not interfere with the arm agent, and vice versa.

Each agent is trained using the PPO algorithm [22] in the IsaacLab simulation environment [28], leveraging input normalization and parallel rollouts across 4096 environments. Both arm and leg policies operate at a control frequency of 100 Hz, and their outputs are added to nominal joint positions $\mathbf{q}^{\text{ref}} \in \mathbb{R}^{18}$, which are then passed to a low-level joint PD controller running at 1 kHz to compute motor torques.

A. Arm agent

The primary role of the arms in this work is to act as momentum generators, reacting to adjust the CAM to desired quantities. The observation space of the arm policy, denoted as $\mathcal{O}_{\text{arm}} \in \mathbb{R}^{41}$, includes both arm-specific and base-related information. It consists of the base height, base heading, base linear and angular velocities in the base frame, the projection of the normalized gravity vector in the base frame, arm joint positions \mathbf{q}_{arm} and velocities $\dot{\mathbf{q}}_{\text{arm}}$, the previous arm action $\mathbf{a}_{\text{arm}}^{t-1}$, as well as the current and target CAM—where the x and y components are expressed in the base frame, and the z component is expressed in the world frame. The critic of the arm agent receives a global observation of size \mathbb{R}^{76} , which includes the full state of both the arm and leg joints, all base information, and both agents' actions, enabling centralized training. The action space $\mathcal{A}_{\text{arm}} \in \mathbb{R}^8$ corresponds to the residual arm joint position commands $\Delta \hat{\mathbf{q}}_{\text{arm}}$.

We design the arm policy's reward structure to encourage arm movements that regulate CAM, consisting of two primary components: a vertical CAM tracking term (Eq. 8) and a horizontal CAM damping term (Eq. 9). The tracking reward r_{CAM} encourages anti-phase arm swing relative to leg motion by aligning the vertical CAM with a reference CAM (Eq. 7), while the damping reward r_{dCAM} penalizes undesirable build-up of horizontal angular momentum, thereby promoting arm responses that dissipate perturbation-induced momentum. In addition, we include the arm regularization terms that follow the same structure as [29].

$$r_{\text{CAM}} = \exp\left(-\frac{\hat{\mathbf{k}}_G^z - \mathbf{k}_G^z}{1 + |\hat{\mathbf{k}}_G^z|}\right)^2 / \sigma \quad (8)$$

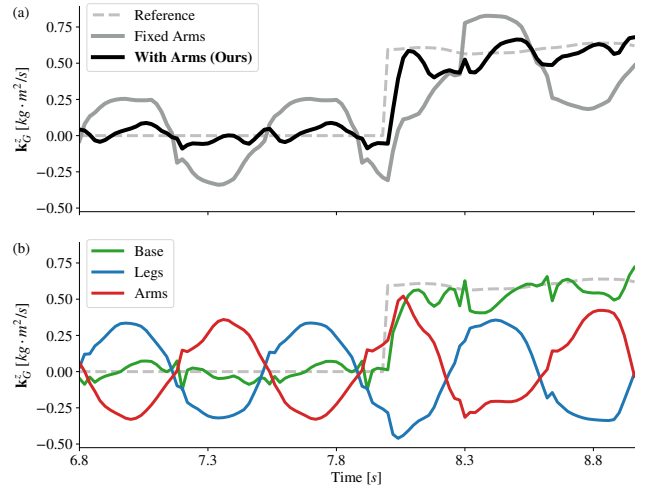


Fig. 4: (a) Comparison of total vertical CAM between our method (*With Arms*), *Fixed Arms* baseline, and the reference. Our method is able to track the desired CAM more closely by leveraging momentum from the arms. (b) Decomposition of the total vertical CAM of our method into contributions from the base, legs, and arms. The angular momentum from the arms and legs effectively cancels each other out, resulting in improved tracking of the reference vertical CAM.

$$r_{\text{dCAM}} = -\min\left(0, \sum_{i \in \{x,y\}} \mathbf{k}_G^i \cdot \dot{\mathbf{k}}_G^i\right) \quad (9)$$

B. Leg agent

The primary objective of the leg agent is to track user-specified base velocity commands while adhering to the desired contact frequency. The observation space of the leg policy, denoted as $\mathcal{O}_{\text{leg}} \in \mathbb{R}^{49}$, includes leg-specific and base-related information. It shares the same observations of the base and CAM as the arm policy, in addition to the velocity command \hat{v}_B , the contact phase signal ϕ , leg joint positions \mathbf{q}_{leg} , velocities $\dot{\mathbf{q}}_{\text{leg}}$, and previous leg action $\mathbf{a}_{\text{leg}}^{t-1}$. Its critic shares the same global observations as the arm agent critic. The action space $\mathcal{A}_{\text{leg}} \in \mathbb{R}^{10}$ represents the residual leg joint position commands $\Delta \hat{\mathbf{q}}_{\text{leg}}$.

We formulate the leg policy's reward to encourage accurate tracking of the user-defined base velocity commands. It is comprised of two primary components: a velocity tracking reward (Eq. 10) and a contact schedule reward (Eq. 11). The velocity tracking term r_{vt} encourages the base link to follow the commanded velocity, while the contact schedule term r_{cs} incentivizes adherence to the specified stepping frequency.

$$r_{vt} = \exp\left(-\frac{(\hat{v}_B - \tilde{v}_B)^2}{1 + |\hat{v}_B|}\right) / \sigma \quad (10)$$

$$r_{cs} = (\mathbb{I}_{r,\text{contact}} - \mathbb{I}_{l,\text{contact}}) \phi_{\text{contact}} \quad (11)$$

Here, \hat{v}_B denotes the commanded base velocity defined in Section III, while $\tilde{v}_B = (v_B^x, v_B^y, \omega_B^z)$ represents the current base linear velocity in the x and y directions, and angular velocity around the z -axis, expressed in the base frame. The terms $\mathbb{I}_{r,\text{contact}}$, $\mathbb{I}_{l,\text{contact}}$, ϕ , and ϕ_{contact} follow the definitions introduced in [29]. As with the arm policy, we also include additional leg regularization terms to ensure smooth motions.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

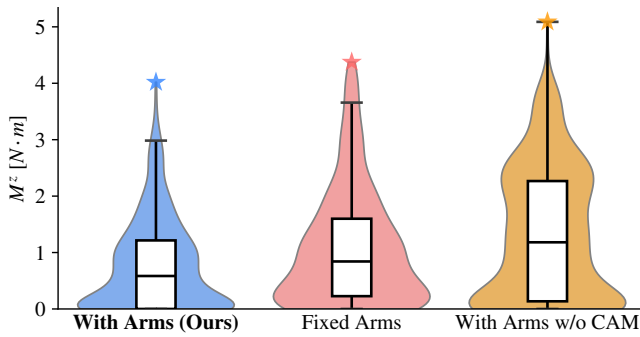


Fig. 5: Comparison of vertical ground reaction moment (GRM) M^z distributions across three controllers: our proposed method, *Fixed Arms*, and *With Arms w/o CAM*, evaluated over 10 walking steps. Our method exhibits a narrower GRM distribution and the lowest peak moment, aligning with the conclusions from [3]. Notably, the passive arm-swing policy (*With Arms w/o CAM*) results in higher GRM peaks and a wider distribution, suggesting that uncontrolled arm motion can negatively impact locomotion stability.

V. RESULTS

In this section, we investigate the benefits of our architecture and reward shaping. We find that leveraging CAM-based rewards induces coordinated and naturalistic arm motion, improving the system’s ability to recover from disturbances. Analyses of the ground reaction moments from the policy reflect those from Collins et al., suggesting that arm swing can reduce undesirable wrenches on the body [3]. Moreover, our limb-level multi-agent RL framework improves learning stability and overall task performance compared to standard single-agent learning. Our results are evaluated on the MIT Humanoid in both simulation and hardware experiments.

A. CAM-regularized arm motion

In this section, we investigate the emergent arm behavior induced by CAM-based reward shaping and analyze its contribution to the improved whole-body locomotion. To highlight the effect of CAM regularization, we compare our methods with two baselines:

- **Fixed Arms:** A baseline where only the legs are actuated and arms remain fixed.
- **With Arms w/o CAM:** A full-body policy that actuates both arms and legs, using the same architecture and training setup as our method, but excluding CAM-related rewards (Eq. 8, Eq. 9).

As shown in Fig. 4, our method tracks the reference CAM far more accurately compared to the *Fixed Arms* baseline. The decomposition of the vertical CAM contributions from the base, legs, and arms shows that the momentum from the arms and legs is consistently out of phase, effectively canceling each other and resulting in better tracking of the reference vertical CAM, consistent with the findings from [3].

But why is minimizing fluctuations in vertical CAM crucial? Increased fluctuations in vertical CAM can be attributed to larger contact wrenches (Eq. 1, Eq. 2), expressed as increased vertical ground reaction moment (GRM) around the stance foot’s center of pressure (CP) as depicted in Fig. 3. Fig. 5 compares the distribution of vertical GRM M^z for our method

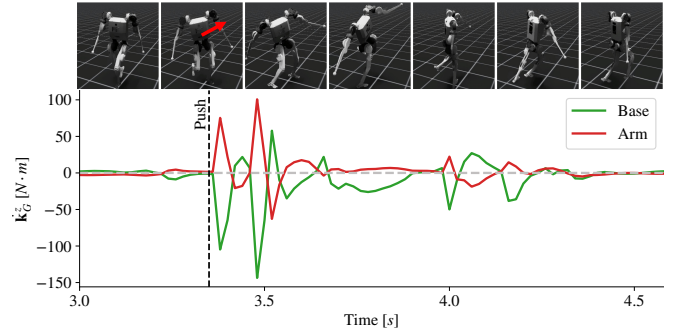


Fig. 6: Emergent arm behavior under external perturbation. A simultaneous torque disturbance of $3 N \cdot m$ along the x, y, z -axes is applied to the robot base at approximately 3.4 s (dashed line). The resulting abrupt time-rate of change of vertical CAM (\dot{k}_G^z , green) in the base is quickly compensated by the emergent arm motion (red), producing opposite-direction momentum that stabilizes the overall CAM. The robot naturally recovers and returns to stable walking within approximately one second after the disturbance.

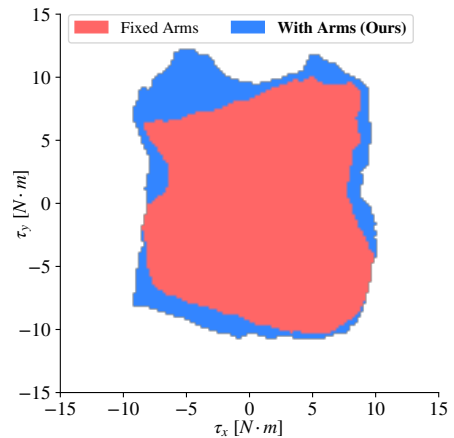


Fig. 7: Comparison of disturbance recovery performance under random torque disturbances applied at $t = 1s$, sampled from the range $[-15N \cdot m, 15N \cdot m]$ along the x, y -axes. The shaded regions represent successful recovery zones over the subsequent 4 seconds for the *Fixed Arms* (red) and our method (blue). The extended coverage of the blue region indicates that our controller remains balanced under a wider range of disturbances, approximately a 23% improvement over the baseline range.

against *Fixed Arms* and *With Arms w/o CAM* over 10 walking steps. Our method shows a narrower GRM distribution and achieves the smallest peak vertical moment, consistent with observations in [3]. This result highlights that properly controlled arm motion guided by CAM observations effectively reduces vertical GRM, thereby decreasing foot slippage and improving overall locomotion stability. Interestingly, the *With Arms w/o CAM* policy performs the worst, exhibiting both the highest peak vertical moment and consistently elevated GRMs throughout the gait cycle. This result suggests that passively swinging arms without intentional CAM-based control may introduce greater instability compared to intentionally fixed arms during locomotion.

Additionally, we investigate how the arm policy can assist with recovery from unexpected disturbances. Such perturbations manifest as abrupt changes in the momentum rate of the robot’s base, requiring rapid compensatory arm motion to stabilize the induced momentum. Fig. 6 demonstrates this emergent arm behavior under external disturbances. In this experiment, we apply a $3 N \cdot m$ torque disturbance to the base

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

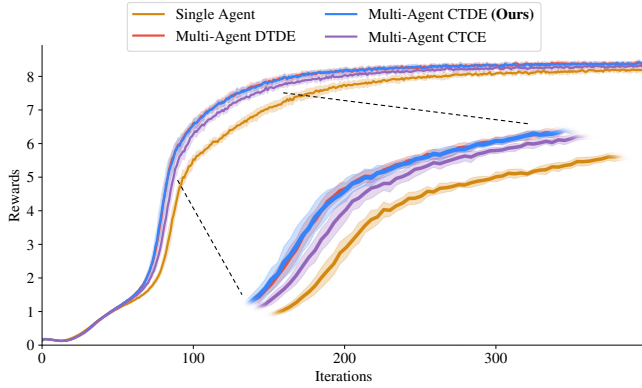


Fig. 8: Reward convergence comparison of our method (*Multi-Agent CTDE*) against three baselines over 400 training iterations. The plotted reward corresponds to the velocity tracking term defined in Eq. 10. Both our approach and *Multi-Agent DTDE* exhibit faster and higher convergence compared to the *Single Agent* baseline and the *Multi-Agent CTCE* variant. These results underscore the advantage of the CTDE design: using local observations for actors while sharing only base state and CAM enables stable and coordinated policy learning.

along the x , y , z -axes.

The resulting jump in k_G^z of the base induced by this perturbation is quickly countered by the emergent arm motion, which generates momentum in the opposite direction and effectively stabilizes the overall CAM. Consequently, the robot swiftly recovers to a stable walking gait within approximately one second after the disturbance.

We further evaluate the robustness of our method by testing its ability to recover from a range of external disturbances. Specifically, we apply random torque disturbances at time $t = 1s$, sampled uniformly from the range $[-15N \cdot m, 15N \cdot m]$ along the x , y -axes, and evaluate the policy’s ability to remain balanced over the subsequent 4 seconds. As shown in Fig. 7, our method achieves approximately 23 % higher success rate compared to the *Fixed Arms* baseline, demonstrating that the emergent arm motion contributes significantly to disturbance rejection, particularly under extreme perturbation conditions.

B. Effectiveness of Limb-Level Multi-Agent RL

In this section, we evaluate the effectiveness of our limb-level multi-agent RL framework, with an emphasis on the importance of the CTDE architecture. To isolate the contribution of this framework, we compare our method against the following three baselines:

- **Single Agent:** A single actor-critic policy that receives the full global observation and outputs both arm and leg actions jointly.
- **Multi-Agent DTDE:** A **D**ecentralized **T**raining and **D**ecentralized **E**xecution setup, where both actor and critic for each agent observe only local (limb-specific) and base information.
- **Multi-Agent CTCE:** A **C**entralized **T**raining and **C**entralized **E**xecution setup, where both actor and critic for each agent observe the full global observation.

We begin by comparing the reward convergence of our approach (*Multi-Agent CTDE*) against the three baselines over 400 training iterations, corresponding to approximately

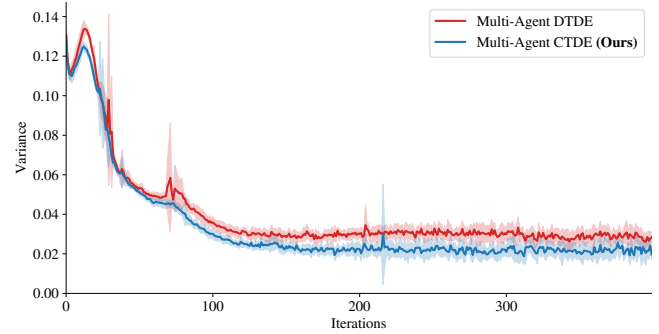


Fig. 9: Variance of the advantage function for the arm agent during training, comparing *Multi-Agent DTDE* and our method. Our approach exhibits faster convergence and consistently lower variance, indicating more stable and reliable advantage estimates. This supports the effectiveness of centralized critics, which leverage global observations to mitigate the multi-agent credit assignment problem and improve policy learning stability.

8 minutes of wall-clock time, averaged over 10 different random seeds. The plotted reward corresponds to the velocity tracking term defined in Eq. 10, which serves as the dominant objective for locomotion. As shown in Fig. 8, both our method and *Multi-Agent DTDE* exhibit the fastest and highest reward convergence. In contrast, the *Single Agent* baseline achieves significantly lower final reward, with noticeably slower convergence. The *Multi-Agent CTCE* variant also performs slightly worse than our method.

In the *Single Agent* setting, all arm and leg rewards are aggregated into a single scalar return, which can lead to conflicting objectives across limbs. As the policy attempts to optimize all reward terms simultaneously, balancing these objectives becomes a sensitive tuning problem, resulting in degraded performance across all components. In contrast, our approach decouples the return for the arms and legs separately, allowing each policy to be updated without mutual interference.

In the case of *Multi-Agent CTCE*, each actor receives the full global observation, increasing the input dimension to 76. This includes limb-specific information from the other agent, which is largely irrelevant and can introduce noise during policy learning. As a result, learning becomes less sample-efficient and convergence is hindered.

These findings support the CTDE design: actors should be restricted to local, limb-specific observations while sharing only base state and CAM, since CAM implicitly captures whole-body dynamics and provides sufficient latent information for coordination without unnecessary input noise.

To better understand the distinction between *Multi-Agent DTDE* and our approach (CTDE), we examine how policy gradients are estimated in RL. As shown in Eq. 6, the policy update relies on the advantage function, which quantifies how good an action is relative to the expected value. This means that the consistency of the advantage estimate given a state is crucial for reliable policy gradient updates. High variance in the advantage function leads to noisy gradient estimates, potentially destabilizing learning. Conversely, lower variance suggests more stable and consistent learning signals. By measuring the variance of

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

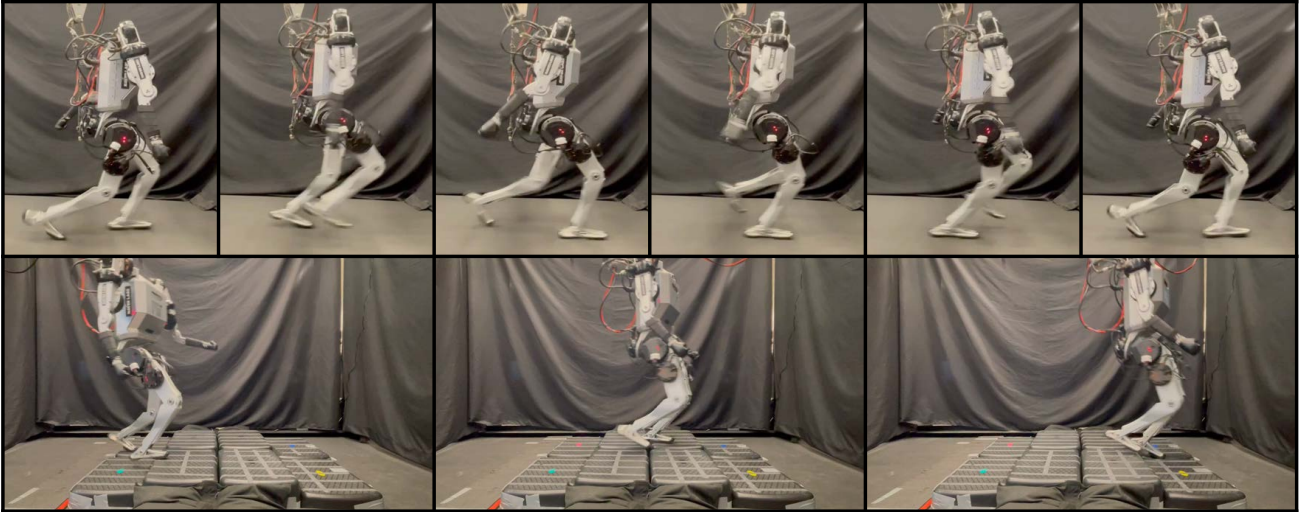


Fig. 10: Hardware demonstrations of the learned policy on the MIT Humanoid. *Top Row*: Forward walking on flat ground with coordinated out-of-phase arm swing relative to the legs, reaching speeds up to 1.3 m/s . The sequence captures one complete gait cycle. *Bottom Row*: Vision-free stair traversal, where adaptive arm motions assist in recovering from missteps and maintaining balance. See supplementary video for full motion sequences.

the advantage function throughout training, we can evaluate how effectively each framework mitigates the multi-agent credit assignment problem and promotes stable, consistent learning signals. To this end, we compare the advantage variance of the arm agent between *Multi-Agent DTDE* and our approach in Fig. 9. Our method exhibits significantly lower final variance and converges more quickly, suggesting that providing the critic with global information is critical for producing reliable advantage estimates and facilitating more stable policy updates.

C. Hardware Experiment

We demonstrate the real-world applicability of our approach by successfully transferring the learned policy to the MIT Humanoid across a variety of locomotion tasks (Fig. 1, Fig. 10). By extending the training environments from flat terrain to more diverse conditions—including rough surfaces, stairs, and inclined planes—we show that the emergent whole-body motion generalizes without modification. Our policy enables the robot to perform various tasks such as walking on flat ground, navigating uneven terrain, and ascending and descending stairs.

The top row of Fig. 10 illustrates forward walking with clear out-of-phase arm swing relative to the legs, achieving speeds up to 1.3 m/s . The bottom row shows the robot crossing stairs without vision, using adaptive arm motion to help recover from missteps and maintain balance. Detailed motion sequences can be found in the supplementary video.

VI. CONCLUSION

In this work, we presented a multi-agent RL framework that leverages CAM objectives to learn the emergent arm swing and coordinated whole-body locomotion. To generate naturalistic arm motion while maintaining stable locomotion, we design separate reward structures for each agent: the leg agent is primarily guided by velocity tracking and contact rewards,

while the arm agent is driven by CAM tracking and damping objectives. Our analysis shows that the learned arm behavior effectively cancels out the CAM generated by the rest of the body, reducing whole-body angular momentum. We further demonstrate that this CAM-driven arm motion lowers vertical GRM during walking and improves the system’s ability to recover from external disturbances. We also provide comparative evaluations showing that our multi-agent CTDE formulation outperforms alternative baselines, including single-agent and centralized actor configurations. These findings underscore the benefit of modular limb-level policies for stable, coordinated control. Finally, we validate the practicality of our approach on the MIT Humanoid, demonstrating successful transfer to diverse real-world locomotion tasks such as walking on flat ground, navigating rough terrain, and climbing stairs.

In future work, we plan to explore more dynamic and contact-intensive scenarios, including one-foot balancing, running, and complex tasks such as dribbling or kicking. We believe that combining our control framework with human motion datasets will further enhance the naturalness and transferability of learned behaviors to real humanoid robots.

ACKNOWLEDGMENT

We thank the members of the Biomimetic Robotics Laboratory at MIT for their insightful discussions and feedback on this work. We especially thank Seungwoo, Charles, Yonghyeon, and Kendrick for their valuable advice on idea development and discussions, and David, Ethan, Justin, Andrew, and Ronak for their support with hardware setup and experiments.

REFERENCES

- [1] H. Elftman, “The function of the arms in walking,” *Human biology*, vol. 11, no. 4, p. 529, 1939.
- [2] H. Herr and M. Popovic, “Angular momentum in human walking,” *Journal of experimental biology*, vol. 211, no. 4, pp. 467–481, 2008.
- [3] S. H. Collins, P. G. Adamczyk, and A. D. Kuo, “Dynamic arm swinging in human walking,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 276, no. 1673, pp. 3679–3688, 2009.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

- [4] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters," *ACM Transactions On Graphics (TOG)*, vol. 41, no. 4, pp. 1–17, 2022.
- [5] A. Tang et al., "Humanmimic: Learning natural locomotion and transitions for humanoid robot via wasserstein adversarial imitation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 13 107–13 114.
- [6] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne, "Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning," *Acm transactions on graphics (tog)*, vol. 36, no. 4, pp. 1–13, 2017.
- [7] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, eadi9579, 2024.
- [8] J.-P. Sleiman, F. Farshidian, and M. Hutter, "Versatile multicontact planning and control for legged loco-manipulation," *Science Robotics*, vol. 8, no. 81, eadg5014, 2023.
- [9] S. Kajita et al., "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(cat. no. 03ch37453)*, IEEE, vol. 2, 2003, pp. 1644–1650.
- [10] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, vol. 35, pp. 161–176, 2013.
- [11] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application for whole-body control," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1 550 039, 2016.
- [12] A. Miyata, S. Miyahara, and D. N. Nenchev, "Walking with arm swinging and pelvis rotation generated with the relative angular acceleration," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 151–158, 2019.
- [13] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2014, pp. 295–302.
- [14] C. Khazoom and S. Kim, "Humanoid arm motion planning for improved disturbance recovery using model hierarchy predictive control," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 6607–6613.
- [15] M. S. Graziano and T. N. Aflalo, "Mapping behavioral repertoire onto the cortex," *Neuron*, vol. 56, no. 2, pp. 239–251, 2007.
- [16] V. Dietz, K. Fouad, and C. Bastiaanse, "Neuronal coordination of arm and leg movements during human locomotion," *European Journal of Neuroscience*, vol. 14, no. 11, pp. 1906–1914, 2001.
- [17] L. Amatucci, G. Turrisi, A. Bratta, V. Barasuol, and C. Semini, "Accelerating model predictive control for legged robots through distributed optimization," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2024, pp. 12 734–12 741.
- [18] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter, "Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2377–2384, 2022.
- [19] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.
- [20] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.
- [21] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
- [22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [23] C. Yu et al., "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in neural information processing systems*, vol. 35, pp. 24 611–24 624, 2022.
- [24] A. SaLoutos, E. Stanger-Jones, Y. Ding, M. Chignoli, and S. Kim, "Design and development of the mit humanoid: A dynamic and robust research platform," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, IEEE, 2023, pp. 1–8.
- [25] J. Carpentier et al., "The pinocchio c++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *2019 IEEE/SICE International Symposium on System Integration (SII)*, IEEE, 2019, pp. 614–619.
- [26] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.
- [27] S. H. Jeon, S. Hong, H. J. Lee, C. Khazoom, and S. Kim, "Cusadi: A gpu parallelization framework for symbolic expressions and optimal control," *IEEE Robotics and Automation Letters*, 2024.
- [28] M. Mittal et al., "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.
- [29] H. J. Lee, S. Hong, and S. Kim, "Integrating model-based footstep planning with model-free reinforcement learning for dynamic legged locomotion," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2024, pp. 11 248–11 255.