DeLM: A Decoupled Training Framework for Wheel-Legged Bipedal Robots to Perform Loco-Manipulation Tasks Stably

Author[1]

***Abstract*— Integrating a manipulator can substantially enhance the task efficiency of wheel-legged bipedal robots. However, manipulator motions introduce time-varying disturbances that make mobile-base stability difficult to maintain. In this paper, we propose a learning-based Decoupled Loco-Manipulation (DeLM) training framework for wheel-legged bipedal robots to solve the issues. First, we present the Arm Randomization Curriculum (ARC), an easy-to-implement arm-motion generator that synthesizes diverse arm motions to inject realistic, time-varying manipulation disturbances into simulation, alleviating the lack of disturbance excitation during training. Second, we propose an Arm-Aware State Estimator (AASE) for locomotion that integrates latent arm representations to better compensate for these time-varying disturbances, improving both stability and robustness. Our experiments demonstrate over a 40% improvement in stability compared with a whole-body control baseline in simulation, and achieve a success rate above 80% in real-world experiments. Within DELM, the robot serves as a stable and robust command-tracking mobile base that supports diverse, precise manipulation tasks, thereby opening up new possibilities for wheel-legged bipedal robots to perform loco-manipulation reliably.**

I. Introduction

Wheel-legged robots combine the high mobility of wheeled platforms [1] with the terrain adaptability of legged systems [2], enabling robust performance across diverse real-world environments. When equipped with robotic arms, these platforms can efficiently perform loco-manipulation tasks [3], significantly enhancing their practical utility and productivity. However, fast and large-amplitude arm motions introduce significant time-varying disturbances to the mobile base, and the intrinsically dynamic wheeled-legged system is particularly sensitive to such perturbations, leading to base drift and degraded manipulation accuracy.

With the rapid progress of Reinforcement Learning (RL) in robotics, an increasing number of studies [5], [6] have adopted learning-based methods for loco-manipulation control. Most existing works [7], [8] address above loco-manipulation tasks through whole-body control (WBC) frameworks, in which the motions of the base and manipulator are jointly optimized under a unified formulation. While such approaches can achieve coordinated behaviors in specific scenarios, they often rely on carefully designed task-dependent objectives, rewards, constraints, or heuristics, which can limit their generalization capability across diverse manipulation tasks and environments.

Fig. 1: **Real-world deployment.** The robot tracks velocity commands to reach the target position, and the robotic arm performs manipulation tasks via two different control methods. (a) The robot grasps a roadside soda can and place it into the onboard bin via human remote teleoperation. (b) The robot picks up a plastic bottle from the ground via the finetuned VLA model GR00T N1 [4], details are in VI-D.

Inspired by the impressive performance of humanoid robot in [9], [10], we propose a learning-based **D**ecoupled **L**oco-**M**anipulation (DeLM) training framework for wheel-legged bipedal robots, which decouples the whole-body control tasks into the upper manipulator task and lower locomotion task. To perform diverse manipulation tasks, we introduce an **A**rm **R**andomization **C**urriculum (ARC) that simulates diverse possible arm motions in the real world. These arm motions serve as structured, time-varying disturbances during training, mimicking the pose-dependent effects of real manipulation (such as changes in inertial coupling and induced base torques) rather than simple domain randomization noise. This helps the lower-level locomotion policy learn to anticipate and compensate for arm-induced dynamics. To achieve stable locomotion under manipulation-induced disturbances, we propose an **A**rm-**A**ware **S**tate **E**stimator (AASE) that uses a history-based state estimator to encode arm-related information for base-policy learning. Leveraging this latent representation together with stability-oriented rewards, the robot attains more robust and stable locomotion performance. Through the robust locomotion, the upper manipulation can be independently realized through human teleoperation or unified frameworks such as vision language action (VLA)

*This work was not supported by any organization

[1]This work was not supported by any organization, This work was not supported by any organization, This work was not supported by any organization `This work was not supported by any organization`

[2]This work was not supported by any organization, This work was not supported by any organization, This work was not supported by any organization `This work was not supported by any organization`

Manuscript 1345 submitted to 2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 5, 2026.

models [11]. The method allows the operator to remotely control the robot's mobility using simple velocity commands while performing precise arm manipulation without explicitly considering base balance.

Finally, we validate the proposed framework in both simulation and real-world experiments. The results demonstrate that our method effectively reduces base pose drift and oscillations under highly dynamic coupling conditions, and achieves a high task success rate in real-world loco-manipulation scenarios in Figure 1.

In summary, our works are concluded as follows:

- We propose a decoupled loco-manipulation (DeLM) training framework for wheel–legged bipedal robots in loco-manipulation tasks. This framework enables the wheel-legged bipedal robot serve as a stable and robust command-tracking mobile base capable of performing diverse manipulation tasks.
- We introduce an arm randomization curriculum (ARC) for manipulation tasks to model manipulation-induces time-varying disturbances as a set of structured arm-motion primitives. This method alleviates insufficient disturbance excitation during training and improves diversity of manipulation tasks.
- We propose an arm-aware state estimator (AASE) that infers compact, dynamics-relevant arm-motion latents from short-horizon observations and provides them to the base controller, improving stability under time-varying disturbances while preserving decoupled control.

II. Related Work

A. Loco-Manipulation Tasks for Legged Robots

Arm-equipped mobile robots provide enhanced dexterity and mobility, enabling effective collaboration with humans. On both quadrupedal and humanoid platforms, loco-manipulation has been extensively investigated using whole-body control frameworks [5]–[9]. For quadrupedal robots, ROA [8] introduced a Regularized Online Adaptation approach to train whole-body controllers for loco-manipulation. UMI-L [7] leveraged both real-world and simulated data to train arm-equipped quadrupeds. For humanoid robots, [12], [13] have demonstrated whole-body control can not only achieve motion clone but also perform better extreme parkour through onboard sensor. PMP [9] introduces a decoupling method to enhance manipulation precision for humanoid robots. However, these methods are developed primarily for purely legged platforms with discrete foothold contacts and do not directly transfer to wheel-legged bipeds, where rolling contacts and underactuated dynamics make the base more sensitive to arm-induced, time-varying disturbances.

B. Stable Locomotion for Wheel-Legged Bipedal Robots

Compared with purely legged robots, maintaining *stable* locomotion on wheel-legged platforms is particularly challenging due to rolling contacts and stronger sensitivity to disturbances. Prior work has achieved robust stability using model-based methods for balance control [14] and for dynamic maneuvers such as jumping with planned contacts [15], [16]. More recently, learning-based approaches have shown strong stabilization capability on low-cost wheel-legged systems, enabling robust behaviors such as blind stair climbing [17] and wheel-legged loco-manipulation [3]. Despite this progress, existing methods still lack a simple and broadly applicable solution that can reliably preserve base stability under manipulation-induced, time-varying disturbances across diverse scenarios.

III. Preliminary

In RL, control problems are typically framed as a Markov Decision Process (MDP), which is represented as $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ with the time step t, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denotes the state transition probability, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denotes the reward functions and $\gamma \in [0, 1]$ denotes the discount factor. The goal of RL is to train a policy π that maximizes the cumulative reward, which is defined as:

$$J_r(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})\right], \tag{1}$$

where $\boldsymbol{s}_t, \boldsymbol{s}_{t+1} \in \mathcal{S}$, $r \in \mathcal{R}$ and $\boldsymbol{a}_t \in \mathcal{A}$. The expectation $\mathbb{E}[...]$ represents the expected discounted return.

To directly address the constrained problems when training, we extend this framework into a Constrained Markov Decision Process (CMDP) [18]. Constrained RL introduces a set $\mathcal{C}$ of cost functions $\{c_1, c_2, ..., c_n\}$ and the corresponding limits $\{\epsilon_1, \epsilon_2, ..., \epsilon_n\}$. Each $c_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denotes the cost of the state transition. The objective is to maximize the reward while keeping the discounted sum of costs c_i below their respective threshold ϵ_i [19], [20], which is formulated as follows:

$$\begin{aligned} \max_{\pi} \quad & J_r(\pi) \\ \text{s.t.} \quad & \forall i \in \{1, \ldots, n\}, \quad J_{c_i}(\pi) \leq \epsilon_i, \end{aligned} \tag{2}$$

where

$$J_{c_i}(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c_i(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})\right]. \tag{3}$$

Observation Space: We denote by $\boldsymbol{O}_t$ the proprioception observations and $\boldsymbol{O}_t^{priv}$ the privileged observations at time t. The proprioception observations $\boldsymbol{O}_t \in \mathbb{R}^{37}$ contain desired linear velocity command v_x^{cmd}, desired angular velocity command w_z^{cmd}, desired height command h_z^{cmd}, joint angles $\boldsymbol{q}$, joint velocities $\dot{\boldsymbol{q}}$, previous action $\boldsymbol{a}_{t-1}$, base angular velocity $\boldsymbol{w}$ and base Euler angles $\boldsymbol{\theta}$. The privileged observations $\boldsymbol{O}_t^{priv} \in \mathbb{R}^{93}$ contain arm observations $\boldsymbol{O}_t^{arm}$, proprioception observations $\boldsymbol{O}_t$ and other privileged information including joint accelerations $\ddot{\boldsymbol{q}}$, joint torques $\boldsymbol{\tau}$, base mass m, base center of mass (CoM), default leg joint positions $\boldsymbol{q}^{leg,d}$ and the joint stiffness and damping coefficients K_p and K_d. The arm observations $\boldsymbol{O}_t^{arm} \in \mathbb{R}^6$ contain the end effector (ee) position $\boldsymbol{P}_{ee}^{arm}$ in the base frame and the CoM of the robotic arm $\boldsymbol{P}_{com}^{arm}$ in the base frame, which are used to assist the

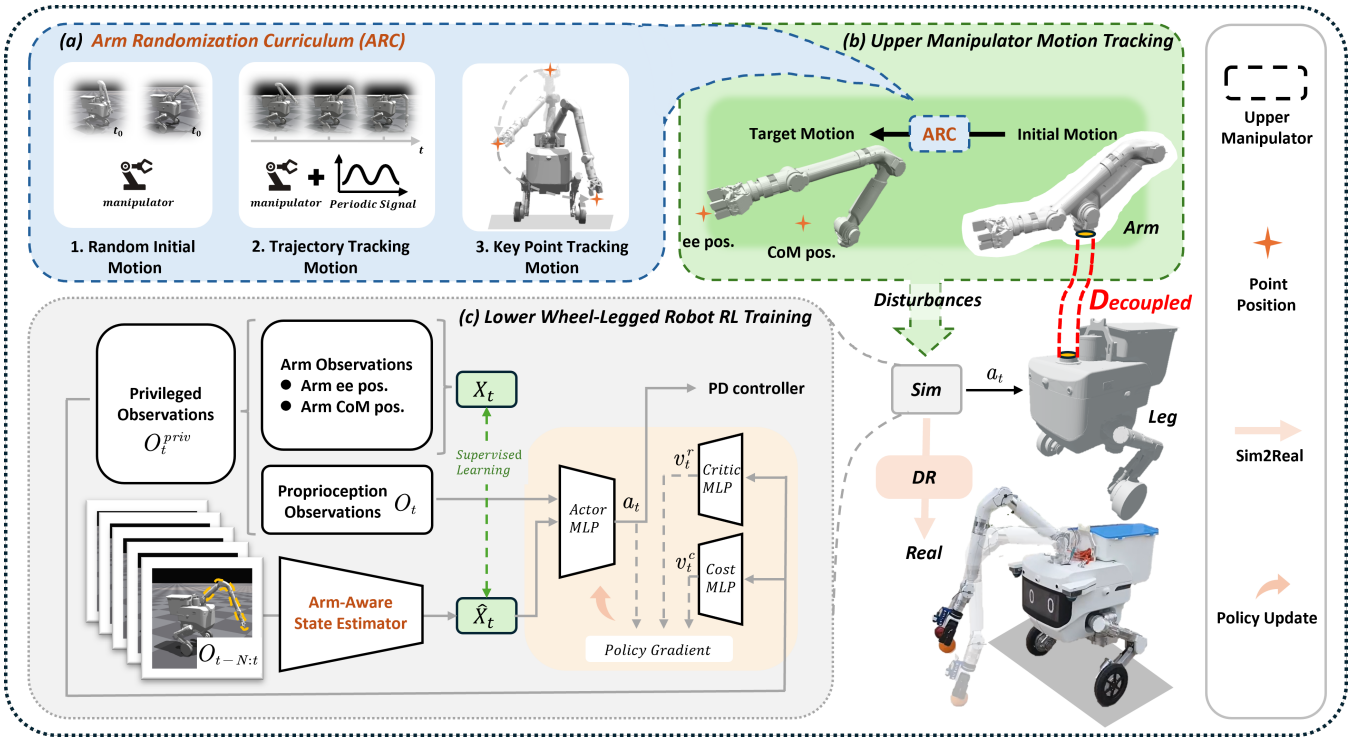

Fig. 2: **Overview of the framework DeLM.** (**a**) We propose an arm randomization curriculum. (**b**) During training, the manipulator moves to target positions according to the proposed randomization strategies. (**c**) Overall RL training framework for lower platform locomotion tasks.

base in inferring the real-time position of the robotic arm to improve the training results.

Action Space: For the lower platform actuated leg joints, $\boldsymbol{a}_t^{leg} \in \mathbb{R}^4$ represents the angular deviation of the robot's joint relative to its default position. For the wheel joints, $\boldsymbol{a}_t^{wheel} \in \mathbb{R}^2$ represents the velocity deviation of the robot's joint relative to its target joint velocity. The robot's joint PD controller reference can denoted as follows with the scale α

$$\boldsymbol{q}_t^{ref} = \alpha \boldsymbol{a}_t^{leg} + \boldsymbol{q}^{leg,d}, \tag{4}$$

$$\dot{\boldsymbol{q}}_t^{ref} = \boldsymbol{a}_t^{wheel}. \tag{5}$$

IV. Methodology

A. Overview of the Framework

As illustrated in Figure 2, Our framework decouples whole-body control into two parallel components: *upper manipulator motion tracking* in (b) and *lower platform RL training* in (c). In (b), we employ an arm randomization curriculum to generate diverse arm motions as disturbances. This curriculum categorizes arm motions into three types: *Random Initial Motion*, *Trajectory Tracking Motion*, and *Key Point Tracking Motion*. These randomized motions act as dynamic disturbances, forcing the lower platform to adapt to varying inertial changes. In (c), we adopt a one-phase training approach with an arm-aware state estimator to learn robust locomotion policies. The estimator leverages the most recent five steps of arm pose and proprioceptive observations to infer arm-related states. To improve training stability and enforce safety-related constraints [3], we adopt NP3O, a normalized penalized variant of PPO, which is particularly beneficial under strong, time-varying disturbances. To bridge the reality gap, we incorporate detailed domain randomization (DR) in simulation before deploying the policy on the real robot.

B. Arm Randomization Curriculum

As illustrated in Figure 4 (a), the robotic arm has 6 degrees of freedom (DoF) in total. We simplify the arm model by focusing on the first three arm joint angles as the target arm joint angles. Since the first three DoFs dominate the primary motion of the robotic arm, the last three DoFs only affect the attitude of the end-effector and have negligible influence on the inertial disturbance exerted on the robot base. We define $\boldsymbol{q}^{arm,g}$ as the target arm pose, $\boldsymbol{q}^{arm,d}$ as the default arm pose. All joint angles in this section are expressed in radians. The first three selected joint components are denoted by $\mathcal{S}$ when needed.

*1) **Random Initial Motion (RIM)**:* The RIM curriculum is designed to simulate a fully randomized initial arm posture $q_i^{RIM} \sim \mathcal{U}(\delta_i^{min}, \delta_i^{max})$, which is sampled uniformly within joint-wise bounds $[\boldsymbol{\delta}^{min}, \boldsymbol{\delta}^{max}]_{RIM}$. We introduce realistic static disturbances at the beginning of each episode. It is formulated as follows:

$$\boldsymbol{q}_{\mathcal{S}}^{arm,g}(t) = \boldsymbol{q}_{\mathcal{S}}^{RIM}(t) + \boldsymbol{q}_{\mathcal{S}}^{arm,d}. \tag{6}$$

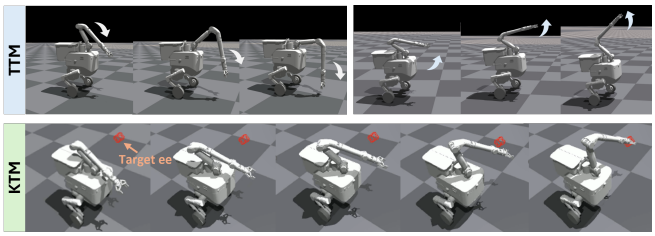

Fig. 3: **Demonstration of ARC.** TTM (top) generates randomized continuous joint-space trajectories, while KTM (bottom) commands task-space end-effector targets (red dots) that are mapped to joint configurations.

*2) **Trajectory Tracking Motion (TTM)**:* The TTM curriculum provides a simple and intuitive way to simulate fully randomized, continuous manipulator motions. The motion covers representative behaviors such as pick-and-place, turning, and placing into a bin (as shown in Figure 1). As shown in Figure 3, TTM generates continuous arm-joint-space trajectories $\boldsymbol{q}^{TTM}$ via joint-level randomization. Specifically, we design the joint motion profiles based on the arm's joint limits and phase scheduling to ensure smooth and continuous manipulator motions. It is formulated as follows:

$$\boldsymbol{q}_{\mathcal{S}}^{TTM}(t) = \begin{bmatrix} a_1 \sin(2\pi f t + \phi_1) \\ a_2 \cos(2\pi f t + \phi_2) + 1 \\ -a_3 \cos(2\pi f t + \phi_3) - 1 \end{bmatrix}, \tag{7}$$

where frequency $f \sim \mathcal{U}(0.1, 0.5)$ Hz, phase $\phi_i \sim \mathcal{U}(0, \pi/4)$ rad and amplitude $a_i \sim \mathcal{U}(\delta_i^{min}, \delta_i^{max})$ follows a uniform distribution with joint-wise bounds $[\boldsymbol{\delta}^{min}, \boldsymbol{\delta}^{max}]_{TTM}$. To overcome the limitations of strictly periodic signals, we synthesize randomized multi-frequency sinusoids using Fourier series to approximate arbitrary waveforms, formulated as:

$$\boldsymbol{q}_{\mathcal{S}}^{TTM}(t) = \sum_{n=1}^{N} \boldsymbol{a}_n \sin(2\pi n f_0 t + \boldsymbol{\phi}_n), \tag{8}$$

where f_0 denotes the fundamental frequency and A_n, ϕ_n are randomly sampled amplitudes and phase shifts, N denotes the total number. To model random temporal pauses at time t_{pause} in manipulator motions, we define:

$$\boldsymbol{q}_{\mathcal{S}}^{TTM}(t) = \boldsymbol{a} \sin(2\pi f t + \boldsymbol{\phi}) \times \mathbb{I}_{t<t_{pause}}, \tag{9}$$

where $\mathbb{I}$ indicates the signal function. Then the target joint positions can be formulated as follows:

$$\boldsymbol{q}_{\mathcal{S}}^{arm,g}(t) = \boldsymbol{q}_{\mathcal{S}}^{TTM}(t) + \boldsymbol{q}_{\mathcal{S}}^{arm,d}. \tag{10}$$

*3) **Keypoint Tracking Motion (KTM)**:* The KTM curriculum is designed to simulate end-effector keypoint tracking motions of the manipulator. For each motion segment, we randomly sample an end-effector target position on a hemisphere centered at the arm base, with the hemisphere radius set to the arm reach. The sampled target is mapped to a feasible joint configuration via inverse kinematics (IK). To improve numerical stability near singular configurations, we use a damped least-squares IK solver and apply feasibility checks. Targets that lead to ill-conditioned Jacobians or IK failure are rejected and resampled. Finally, we connect consecutive IK solutions using time-parameterized second-order spline interpolation to generate smooth, continuous joint trajectories, ensuring coherent manipulator motions throughout training, as shown in Figure 3. The target arm joint positions can be defined as:

$$\boldsymbol{q}_{\mathcal{S}}^{arm,g}(t) = \boldsymbol{q}_{\mathcal{S}}^{\text{KTM}}(t) + \boldsymbol{q}_{\mathcal{S}}^{arm,d}. \tag{11}$$

TABLE I
REWARD FUNCTION DESIGN.

Reward	Equation	Weight
Task		
Lin. vel. tracking (x)	$\exp(-4\|\|v_x^{cmd} - v_x\|\|^2)$	1.0
Ang. vel. tracking (z)	$\exp(-4\|\|w_z^{cmd} - w_z\|\|^2)$	1.0
Base height	$\exp(-1000\|\|h^{cmd} - h\|\|^2)$	1.0
Euler (y)	$\exp(-160\|\|\theta_y\|\|^2)$	0.8
Ang. vel. (y)	$\exp(-50\|\|w_y\|\|^2)$	0.8
Feet distance	$\exp(-100 d^{feet})$	0.2
Regularization		
Lin. vel. (z)	$\|\|v_z\|\|^2$	-1e-4
Ang. vel. (xy)	$\|\|w_{xy}\|\|^2$	-0.05
Joint vel.	$\|\|\dot{\boldsymbol{q}}^{leg}\|\|^2$	-5e-4
Joint acc.	$\|\|\ddot{\boldsymbol{q}}^{leg}\|\|^2$	-5e-7
Joint power	$\|\boldsymbol{\tau}^{leg}\dot{\boldsymbol{q}}^{leg}\|$	-1e-8
Action rate	$\|\|\boldsymbol{a}_{t-1} - \boldsymbol{a}_t\|\|^2$	-0.2
Action smoothness	$\|\|\boldsymbol{a}_{t-2} + \boldsymbol{a}_t - 2\boldsymbol{a}_{t-1}\|\|^2$	-0.5
Penalty		
Collision	$\mathbb{I}_{\|\|\boldsymbol{F}_{col}\|\|>0.1}$	-20.0
Stand still penalty	$\|\|\boldsymbol{v}\|\|^2 \times \mathbb{I}_{\|\|\boldsymbol{v}_{xy}^{cmd}\|\|<0.1}$	-50.0
Orientation mismatch	$\|\|\boldsymbol{g}_{xy}\|\|^2$	-10.0

TABLE II
COST FUNCTION DESIGN.

Cost	Equation	Weight
Joint pos.	$\|\|\boldsymbol{q}^{leg} - \boldsymbol{q}_{lim}^{leg}\|\| \times \mathbb{I}_{\|\|\boldsymbol{q}^{leg} - \boldsymbol{q}_{lim}^{leg}\|\|>0}$	0.3
Joint vel.	$\text{clip}(\|\|\dot{\boldsymbol{q}}^{leg}\|\| - 0.8\dot{\boldsymbol{q}}_{lim}^{leg}, 0, 1)$	0.3
Joint torque	$\text{clip}(\|\|\boldsymbol{\tau}^{leg}\|\| - 0.8\boldsymbol{\tau}_{lim}^{leg}, 0, 1)$	0.3
Acc. smoothness	$0.1\text{max}(\|\|\ddot{\boldsymbol{q}}^{leg}\|\| - \ddot{\boldsymbol{q}}_{lim}^{leg}, 0)$	0.1

C. Arm-Aware State Estimator

Under our decoupled framework, accurate base awareness of the manipulator pose is crucial for stable locomotion. Although the training observation $\boldsymbol{O}_t$ includes arm joint states, these local measurements are often insufficient to capture the arm's key pose variations that dominate the coupled dynamics, limiting the mobile base's ability to reason about manipulation-induced disturbances. We address this by introducing two stability-relevant arm position latent variables: $\boldsymbol{P}_{com}^{arm}$, which represents the arm's overall position awareness during motion, and $\boldsymbol{P}_{ee}^{arm}$, which represents end-effector position awareness during grasping and manipulation. Specifically, we encode a short history of arm joint states together with proprioceptive observations, and employ a state estimator to explicitly predict $\boldsymbol{P}_{com}^{arm}$ and $\boldsymbol{P}_{ee}^{arm}$. These

estimates are then used as inputs to the locomotion policy network.

V. Implementation Details

A. Robot Details

As shown in Figure 4, the wheel-legged bipedal robot has 5 DoFs for each leg, the manipulator has 6 DoFs. There is a one-DoF gripper equipped with a fisheye RGB camera at the end-effector. On the real robot, the manipulator is controlled via joint-angle mapping with a position-loop servo.

B. Reward Function Design

In Table I, we classify our reward functions into three categories based on their functionality: task rewards, regularization rewards and penalty rewards. We mainly use `euler_y` and `ang_vel_y` to maintain the base orientation and prevent large-scale swinging. The `stand_still_penalty` helps reduce unnecessary velocity when the agent is standing still. Additionally, we apply strong regularization on `action_rate` and `action_smoothness` to prevent large-scale collisions.

C. Cost Function Design

In Table II, we define four constraint limits for safety [3]: joint position limits $\boldsymbol{q}_{lim}^{leg}$, joint velocity limits $\dot{\boldsymbol{q}}_{lim}^{leg}$, joint torque limits τ_{lim}^{leg} and acceleration limits $\ddot{\boldsymbol{q}}_{lim}^{leg}$. All of the cost functions are defined as once the corresponding value exceeds its predefined limit, the excess will be recorded and accumulated as part of the final cost signal for cost critic network.

D. Domain Randomization

DR plays a crucial role in reducing the sim-to-real gap. By carefully selecting appropriate randomization ranges, real-world conditions can be more accurately approximated within the simulation environment, thereby improving the success rate of zero-shot transfer. The specific parameters and their DR ranges used in our work are summarized in Table III. Through empirical analysis, we identified three motor parameters (marked red) that critically impact real-world performance: joint motor friction, damping and armature. Despite some basic DR for RL training (marked black), we also use many delay DR (marked red) to possibly reduce real world time delay effects.

TABLE III
Domain Randomization.

Randomization Term	Range	Unit
Mass	[-5, 5]	Kg
CoM of base	[-0.05, 0.05]	m
Motor offset	[-0.03, 0.03]	rad
Friction	[0.1, 2]	-
Restitution	[0, 1]	-
Inertia	[0.8, 1.2]	-
K_p factor	[0.9, 1.1]	-
K_d factor	[0.9, 1.1]	-
Motor strength factor	[0.9, 1.1]	-
Torque delay	[0,10]	ms
Obs. delay	[0,5]	ms
Action delay	[10, 35]	ms
Joint delay	[0, 10]	ms
Imu delay	[25, 55]	ms
Joint friction	[0.9, 1.1]	-
Joint damping	[0.9, 1.1]	-
Joint armature	[0.9, 1.1]	-

Black: physical properties randomization variants.
Blue: system delays randomization variants.
Red: significant joint properties randomization variants.

VI. Experiments

A. Experiment Setup

1) ***Training Setup****:* We use Isaac Gym [21] as our simulator to train 4,096 parallel environments simultaneously, each one was trained for 10,000 iterations on a Nvidia RTX 4070 GPU. The simulation time step is set to 0.002 s and the environment time step is set to 0.01 s. The actor MLP has three hidden layers of (128,64,32), and reward critic and cost critic has three hidden layers of (256,128,64), and AASE has two hidden layers of (128,64). Core policy hyperparamaters are γ=0.99, lr=5.0e-4, λ_{GAE}=0.95, desired_kl=1.0e-2 and entropy_corf=1.0e-2. We evaluate all the methods across 2,000 environments with the same random seeds and training configuration. During training, the ARC curriculum allocates environments with proportions of $[\text{RIM}, \text{TTM}, \text{KTM}] = [20\%, 40\%, 40\%]$. We uniformly sample $a_i \sim \mathcal{U}(\delta_i^{min}, \delta_i^{max})$ with $[\boldsymbol{\delta}^{min}, \boldsymbol{\delta}^{max}]_{\text{RIM}} = [-0.3, 0.3]_{a_1}, [0, 2.78]_{a_2}, [0, 2.0]_{a_3}$ and $[\boldsymbol{\delta}^{min}, \boldsymbol{\delta}^{max}]_{\text{TTM}} = [-0.3, 0.3]_{a_1}, [1.125, 1.425]_{a_2}, [0.62, 1.02]_{a_3}$ for the arm joints.

2) ***Manipulation Setup****:* For manipulation tasks, we use TTM and KTM in ARC (Sec. IV-B) to emulate continuous arm motions and target-tracking arm motions in simulation, respectively. In real-world deployment, we evaluate two arm-control modes: teleoperated arm control and VLA-based autonomous grasping (Sec. VI-D).

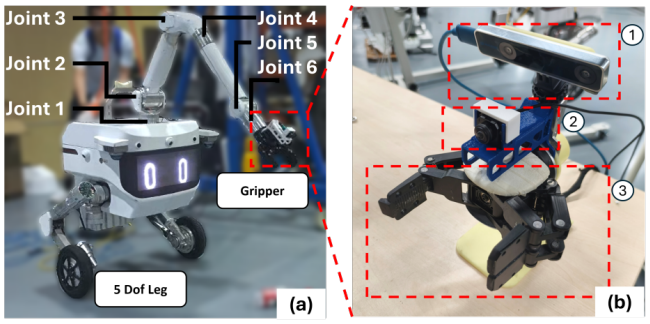

Fig. 4: **Real robot and data-collection gripper.** (a) The manipulator has 6 DoFs and each leg has 5 DoFs. A fisheye RGB camera is mounted at the manipulator end-effector. (b) The gripper is used for VLA data collection training, and its components are as follows: (1) a RealSense depth camera, (2) a fisheye camera, (3) a gripper identical to the manipulator end-effector. The gripper is actuated via a trigger to control its open or close.

TABLE IV
EXPERIMENT RESULTS ON DIFFERENT ALGORITHM.

Method	$D_{drift} \downarrow$	$D_{\theta_y} \downarrow$	$S_{w_y} \downarrow$	$D_{heading} \downarrow$	$E_{vel} \downarrow$	$E_{ang} \downarrow$	$E_{ee} \downarrow$
Continuous arm motions via TTM							
Baseline	1.296 ± 1.275	0.199	0.277	0.174	0.438 ± 0.351	0.053 ± 0.069	-
WBC	0.542 ± 0.287	0.041	0.297	0.103	0.166 ± 0.103	0.049 ± 0.048	-
HIM	0.295 ± 0.237	0.039	0.203	0.075	$\mathbf{0.143} \pm 0.095$	0.037 ± 0.032	-
Ours	$\mathbf{0.247} \pm 0.171$	**0.030**	**0.130**	**0.051**	0.152 ± 0.102	$\mathbf{0.035} \pm 0.031$	-
Ours w/o ARC	0.381 ± 0.327	0.038	0.228	0.129	0.169 ± 0.111	0.037 ± 0.033	-
Ours w/o AASE	0.313 ± 0.244	0.033	0.147	0.152	0.159 ± 0.107	0.049 ± 0.043	-
Target-tracking arm motions via KTM							
Baseline	1.061 ± 1.117	0.309	0.136	0.193	0.439 ± 0.362	0.061 ± 0.098	0.236 ± 0.161
WBC	0.440 ± 0.325	0.048	0.301	0.093	0.178 ± 0.112	0.048 ± 0.047	0.208 ± 0.091
HIM	0.369 ± 0.283	0.039	0.192	0.143	0.157 ± 0.095	0.044 ± 0.041	0.224 ± 0.079
Ours	$\mathbf{0.223} \pm 0.158$	**0.028**	**0.106**	**0.044**	$\mathbf{0.154} \pm 0.101$	$\mathbf{0.039} \pm 0.036$	$\mathbf{0.176} \pm 0.079$
Ours w/o ARC	0.304 ± 0.230	0.039	0.129	0.140	0.168 ± 0.108	0.413 ± 0.038	0.195 ± 0.089
Ours w/o AASE	0.288 ± 0.209	0.032	0.122	0.172	0.156 ± 0.103	0.050 ± 0.046	0.216 ± 0.078

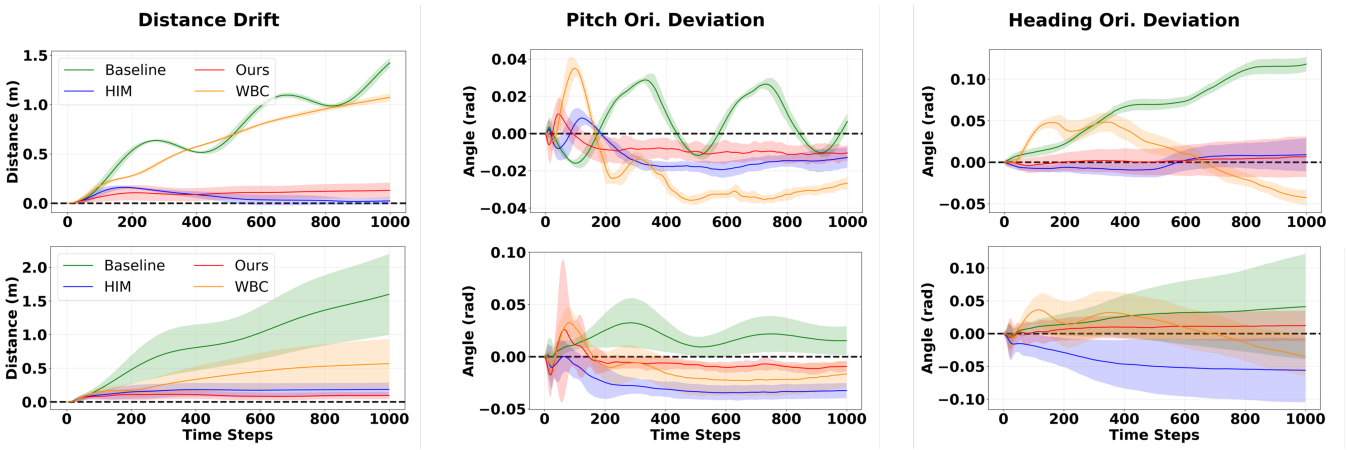

Fig. 5: **Comparative experiment evaluation curves**. Our method's curve is marked by red color. In the figure array, the top row is under manipulator motion by TTM and the bottom row is by KTM. The curves are obtained by averaging results across all environments, and the shaded regions indicate the variation range across trials. The horizontal axis represents time steps during the evaluation phase (the maximum is 1,000). The **black dashed line** denotes the ideal zero-reference level, with curves closer to this line indicating better performance. During testing, the robot is subjected to a sudden velocity increase of 0.05 m/s every 8 seconds to simulate external pushes for robustness. E_{ee} is not evaluated for TTM since the motion is specified in joint space in simulation.

B. Evaluation Metrics

To evaluate stabilization performance under manipulation-induced disturbances, we use the following metrics:

1) ***Stability****:* We evaluate *pre-grasp (quasi-static) stability*, which measures how well the robot keeps its base stationary with minimal oscillation or deviation while aligning the end-effector to the target before grasping. We use four metrics: (1) **base distance drift** $D_{\text{drift}} = \|\boldsymbol{P}_{t_n}^{\text{base}} - \boldsymbol{P}_{t_0}^{\text{base}}\|$ (m), where $\boldsymbol{P}_{t_0}^{\text{base}}$ and $\boldsymbol{P}_{t_n}^{\text{base}}$ are the initial and final base positions, (2) **max pitch orientation deviation** $D_{\theta_y} = \max\|\theta_y\|$ (rad), which captures the magnitude of pitch oscillations during the pre-grasp phase, (3) **pitch angular-velocity standard deviation** $S_{\omega_y} = \sigma(\omega_y)$ (rad/s), defined as the standard deviation of the pitch angular velocity, reflecting the magnitude of angular-velocity fluctuations and the intensity of pitch jitter, and (4) **max heading orientation deviation** $D_{\text{heading}} = \max||\theta_z||$ (rad), which measures undesired yaw drift while the robot is nominally stationary during pre-grasp, often caused by base twisting induced by time-varying arm motions. Notably, D_{drift} is an evaluation-only metric to assess the learned policy. While additional localization (for example, LiDAR-based odometry) could be used in real deployments, we omit it here to isolate the contribution of the learned policy.

2) ***Locomotion****:* We evaluate locomotion capability during mobile manipulation. We measure motion-tracking performance using the desired linear velocity tracking error E_{vel} (m/s) and desired angular velocity tracking error E_{ang} (rad/s).

3) ***Precision****:* The gripper end-effector position error $E_{ee} = ||\boldsymbol{P}_{ee}^{arm,g} - \boldsymbol{P}_{ee}^{arm}||$ (m) is defined to evaluate grasping accuracy.

Manuscript 1345 submitted to 2026 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Received March 5, 2026.

Fig. 6: **One demonstration of outdoor waste collection task**. The robot navigates to the target location, executes continuous and complex manipulation motions to place the waste into the onboard bin, and then continues to the next target.

C. Experiment Results

For the comparative experiment, we benchmark our method against the following baselines to evaluate stability:

- **Baseline**: The policy was trained using PPO.
- **HIM**: The policy was trained using Hybrid Internal Model, a method using contrast learning [22].
- **WBC**: The whole training framework was changed to whole-body control framework to track the end-effector target position [3].

For the ablation study, we benchmark the following ablated variants to evaluate the contribution of each component:

- **Ours w/o ARC**: In spite of using ARC (Section IV-B), the term applies larger disturbance DR such as random push for 0.1 m/s every 8 seconds and random base CoM for [-0.2,0.2] to simulate arm-induced disturbances.
- **Ours w/o AASE**: Policy was trained using our method but without AASE in the framework (Section IV-C).

*1) **Comparative Experiment**:* The results are shown in Table IV. Regarding *stability*, *Ours* achieves D_{drift} of 0.247 (TTM) and 0.223 (KTM), outperforming *WBC* by 44.1% and 49.3%, respectively. It also yields the lowest D_{θ_y}, S_{w_y}, and D_{heading} in both settings, reduce the deviation from the second-best by 23.1%, 31.7%, 32.0% in TTM and 28.2%, 22.1%, 52.7% in KTM, indicating consistently reduced oscillation and yaw drift. For *locomotion*, *Ours* attains the lowest E_{ang} (0.035 and 0.039) and the second-lowest E_{vel} (0.152 and 0.154). For *precision*, *Ours* achieves the lowest end-effector error E_{ee} of 0.176, improving over the second-best method by 9.7 percentage points. Overall, these results show that our method improves stability, robustness, and manipulation accuracy in loco-manipulation. In contrast, *Baseline* degrades substantially under time-varying arm-induced disturbances. *WBC* performs better under KTM than TTM, likely because it is designed around target-tracking objectives, while *HIM* maintains slightly better E_{vel} in TTM but is less stable than our method.

We further visualize the results in Figure 5. For *Distance Drift*, *Baseline* and *WBC* show clear drift accumulation over time, while *Ours* remains bounded with consistently small drift. For *Pitch Orientation Deviation*, *Ours* stays close to zero after the initial transient, whereas *Baseline* and *WBC* exhibit larger deviations and more pronounced oscillations. For *Heading Orientation Deviation*, *Baseline* and *HIM* gradually depart from the initial heading, while *Ours* maintains a near-constant heading with minimal drift. Overall, the curves indicate that our method yields a more stable policy under arm-induced disturbances, with reduced base drift and smaller orientation deviations.

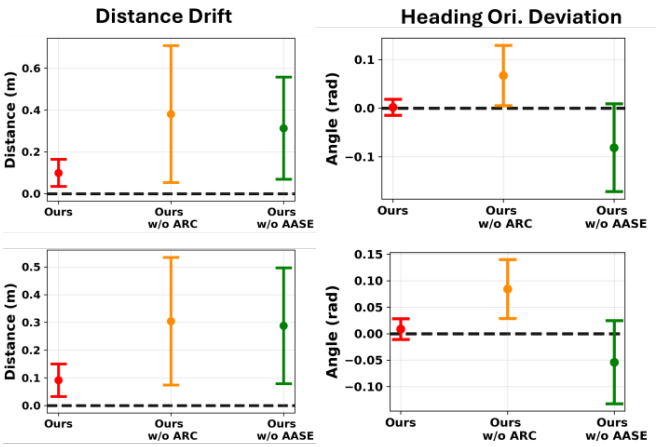

Fig. 7: **Ablation Study**. The top row is under TTM and the bottom row is under KTM.

*2) **Ablation Study**:* As shown in Figure 7, removing ARC results in substantially larger *Distance Drift* than *Ours*, with increases of 54.3% under TTM and 36.3% under KTM. This indicates weaker robustness to arm-motion scenarios and leads to more pronounced base displacement during manipulation. These results highlight that ARC is more effective than generic domain randomization alone, as synthesizing diverse *time-varying* arm-induced disturbances is crucial for training a stable mobile base. In contrast, removing AASE yields noticeably larger *Heading Orientation Deviation* and *Distance Drift*. This indicates that without arm-motion awareness, the policy cannot adequately compensate for arm–base dynamic coupling, leading to systematic yaw drift and residual base displacement even when the robot is nominally stationary.

D. Real Robot Experiment

In real-world deployment, the robot tracks velocity commands to approach a target object, while the manipulator executes grasp-and-place using two arm-control modes. Both modes share the same learned base policy for stable command tracking under arm motions.

*1) **Teleoperation**:* For teleoperation, we use the LeRobot master arm as an input device to control the onboard manipulator. This setup enables large-range arm motions and

TABLE V
REAL-WORLD GRASP SUCCESS RATES.

Method	Teleoperation	Autonomous
Ours	95% (38/40)	85% (34/40)
Baseline	80% (32/40)	65% (26/40)

precise pick-and-place operations, such as grasping objects at challenging locations and placing them into the onboard bin, as shown in Figure 1 (a).

*2) **VLA-based autonomous grasping**:* As shown in Figure 4 (b), we collect expert demonstrations using a gripper-centric teleoperation setup, where a depth camera provides gripper pose estimates and a fisheye RGB camera captures visual observations. We fine-tune the GR00T N1 VLA model on the collected dataset. The data-collection manipulator is identical to the onboard manipulator, ensuring kinematic consistency. During deployment, the VLA policy runs onboard at inference time on the robot GPU, as shown in Figure 1 (b).

As shown in Figure 6, we evaluate our method on a real robot under two arm-control modes: teleoperation and VLA-based autonomous grasping. We conduct 40 grasp trials in total, including indoors and outdoors. The success rate is defined as the fraction of trials in which the robot achieves a successful stable grasp. As shown in Table V, compared with the baseline, our method improves the success rate by 25% under teleoperation and by 20% under VLA-based grasping, demonstrating clear real-world gains. We observe two primary failure modes: accumulated quasi-static base drift and high-frequency base oscillations, both of which can cause residual misalignment at the moment of grasp. Teleoperation yields a higher success rate because the operator can compensate for residual base misalignment, whereas VLA relies on vision-based inference and is more sensitive to outdoor lighting changes and terrain conditions.

VII. CONCLUSION

In this paper, we presented DeLM, a learning-based decoupled training framework that improves mobile-base stability for wheel-legged bipedal robots during loco-manipulation. By introducing ARC, an easy-to-implement arm-motion curriculum that injects realistic, time-varying manipulation disturbances in simulation, and AASE, an arm-aware state estimator that incorporates latent arm representations, DeLM significantly enhances balance robustness and command-tracking performance under manipulator motions. Extensive evaluations in simulation and real-world deployment demonstrate consistent stability gains and high grasping success rates in both indoor and outdoor settings. Limitations remain on highly unstructured terrains, and future work will address more challenging terrain conditions and tighter loco-manipulation coordination.

REFERENCES

[1] L. Bruzzone and G. Quaglia, "Locomotion systems for ground mobile robots in unstructured environments," *Mechanical sciences*, vol. 3, no. 2, pp. 49–62, 2012.

[2] C. D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3359–3365.

[3] Z. Wang, Y. Jia, L. Shi, H. Wang, H. Zhao, X. Li, J. Zhou, J. Ma, and G. Zhou, "Arm-constrained curriculum learning for loco-manipulation of a wheel-legged robot," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 10 770–10 776.

[4] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[5] Y. Ma, A. Cramariuc, F. Farshidian, and M. Hutter, "Learning coordinated badminton skills for legged manipulators," *Science Robotics*, vol. 10, no. 102, p. eadu3922, 2025.

[6] Y. Ma, Y. Liu, K. Qu, and M. Hutter, "Learning accurate whole-body throwing with high-frequency residual policy and pullback tube acceleration," *arXiv preprint arXiv:2506.16986*, 2025.

[7] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song, "Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers," *arXiv preprint arXiv:2407.10353*, 2024.

[8] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning*. PMLR, 2023, pp. 138–149.

[9] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang, "Mobile-television: Predictive motion priors for humanoid whole-body control," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 5364–5371.

[10] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang, "Exbody2: Advanced expressive humanoid whole-body control," *arXiv preprint arXiv:2412.13196*, 2024.

[11] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[12] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *arXiv preprint arXiv:2406.08858*, 2024.

[13] Z. Zhuang, S. Zhu, M. Zhao, and H. Zhao, "Deep whole-body parkour," *arXiv preprint arXiv:2601.07701*, 2026.

[14] S. Wang, L. Cui, J. Zhang, J. Lai, D. Zhang, K. Chen, Y. Zheng, Z. Zhang, and Z.-P. Jiang, "Balance control of a novel wheel-legged robot: Design and experiments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6782–6788.

[15] H. Chen, B. Wang, Z. Hong, C. Shen, P. M. Wensing, and W. Zhang, "Underactuated motion planning and control for jumping with wheeled-bipedal robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 747–754, 2020.

[16] V. Klemm, A. Morra, C. Salzmann, F. Tschopp, K. Bodie, L. Gulich, N. Küng, D. Mannhart, C. Pfister, M. Vierneisel, *et al.*, "Ascento: A two-wheeled jumping robot," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 7515–7521.

[17] S. Chamorro, V. Klemm, M. d. L. I. Valls, C. Pal, and R. Siegwart, "Reinforcement learning for blind stair climbing with legged and wheeled-legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8081–8087.

[18] E. Altman, *Constrained Markov decision processes*. Routledge, 2021.

[19] J. Lee, L. Schroth, V. Klemm, M. Bjelonic, A. Reske, and M. Hutter, "Evaluation of constrained reinforcement learning algorithms for legged locomotion," *arXiv preprint arXiv:2309.15430*, 2023.

[20] Y. Kim, H. Oh, J. Lee, J. Choi, G. Ji, M. Jung, D. Youm, and J. Hwangbo, "Not only rewards but also constraints: Applications on legged robot locomotion," *IEEE Transactions on Robotics*, vol. 40, pp. 2984–3003, 2024.

[21] 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.

[22] J. Long, Z. Wang, Q. Li, J. Gao, L. Cao, and J. Pang, "Hybrid internal model: Learning agile legged locomotion with simulated robot response," *arXiv preprint arXiv:2312.11460*, 2023.