Learning Decoupled Training Methods for High-Inertia Wheel-Legged Robot to Move and Grasp

***Abstract*— It is always a challenging task for arm-equipped wheel-legged bipedal robots to stably move and grasp. However, when performing loco-manipulation tasks, how to keep the lower-body mobile platform stable is always a critical problem. Current research primarily focuses on low-cost, lightweight robot platforms, but effective control methods for large-size, high-inertia robots still remain insufficient. To address these challenges, we propose a learning-based training framework called Decoupled Loco-Manipulation (DeLM), which decouples the whole body control tasks into upper-body manipulation tasks and lower-body locomotion tasks. DeLM is specifically designed to enhance lower-body balance on high-inertia mobile platforms while maintaining extensibility for upper-body manipulation tasks. In particular, we introduce an Arm Randomization Curriculum (ARC) method within the framework to improve the robot's dynamic stability by diversifying arm poses. This approach effectively improves the robustness of the lower-body balance during training. Finally, we introduce a parameter calibration method to reduce the sim-to-real gap and we successfully apply our method on a 65.7 Kg high-inertia wheel-legged bipedal robot, demonstrating stable grasping in tasks such as bottle grasping and waste collection, as shown in Figure 1. To the best of our knowledge, this is the first successful implementation of a learning-based approach for stable grasping tasks on such high-inertia wheel-legged bipedal robot platforms with a well-defined application scenario.**

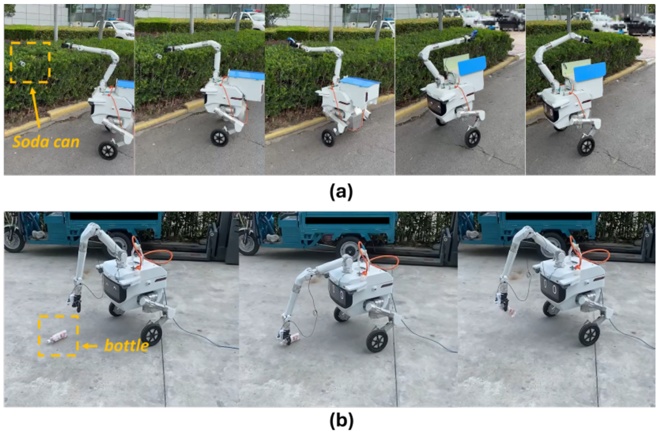

Fig. 1: Real world test. (a) shows the robot collecting a soda can from bushes and placing it into the onboard bins via teleoperation. (b) shows the robot using its onboard fisheye camera for visual inference to autonomously pick up bottles from the ground.

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, significantly enhancing their practical utility and productivity.

With the impressive progress of Reinforcement Learning (RL) in robotics, an increasing number of studies have adopted learning-based methods for robot control. Prior research on loco-manipulation primarily focuses on low-cost, lightweight mobile platforms [3]–[7], achieving promising results in tasks such as locomotion [7], loco-manipulation [8] and navigation [9]. However, some robots have to be designed for specialized applications, often resulting in larger size and higher mass due to their structural configurations. The big-size or high-inertial properties make them harder to control compared to the lightweight robots [10]. Therefore, it still lacks a simple and effective control method to solve the problem, especially for wheel-legged bipedal robots that meet the conditions.

In loco-manipulation tasks, the ideal goal for the robots is to maintain absolute lower-body stability, regardless of the upper body's manipulation tasks. Some works propose the method for decoupling upper body control from locomotion in the control of humanoid robot, using inverse kinematics (IK) and motion retargeting for precise manipulation, while RL focuses on robust lower-body locomotion [11], [12]. We adopt a similar decoupling strategy, but apply it to an arm-equipped wheel-legged robot. Unlike humanoid robots with flat-foot support, the inherent instability of wheels poses greater challenges for maintaining balance during loco-manipulation tasks.

To address the challenges, we propose a learning-based training framework called Decoupled Loco-Manipulation (DeLM). DeLM decouples the whole-body control tasks into two tasks: upper-body manipulation task and lower-body locomotion task. To tackle the stability issues of the high-inertia floating base, DeLM specifically focuses on the lower-body balance tasks. It also preserves extensibility for many arm manipulation tasks on the mobile robot platform, such as some Vision-Language-Action (VLA) [13] models or tele-operations. Furthermore, we introduce Arm Randomization Curriculum (ARC) within the framework to enhance the robustness of lower-body stability. ARC simulates various arm poses as real disturbances on lower-body training to improve the balance robustness of the robot. Ultimately, we validated the effectiveness of our approach in both simulation and real world. We successfully apply our method on a 65.7 Kg wheel-legged bipedal robot to complete loco-manipulation tasks stably, as shown in Figure 1. This method

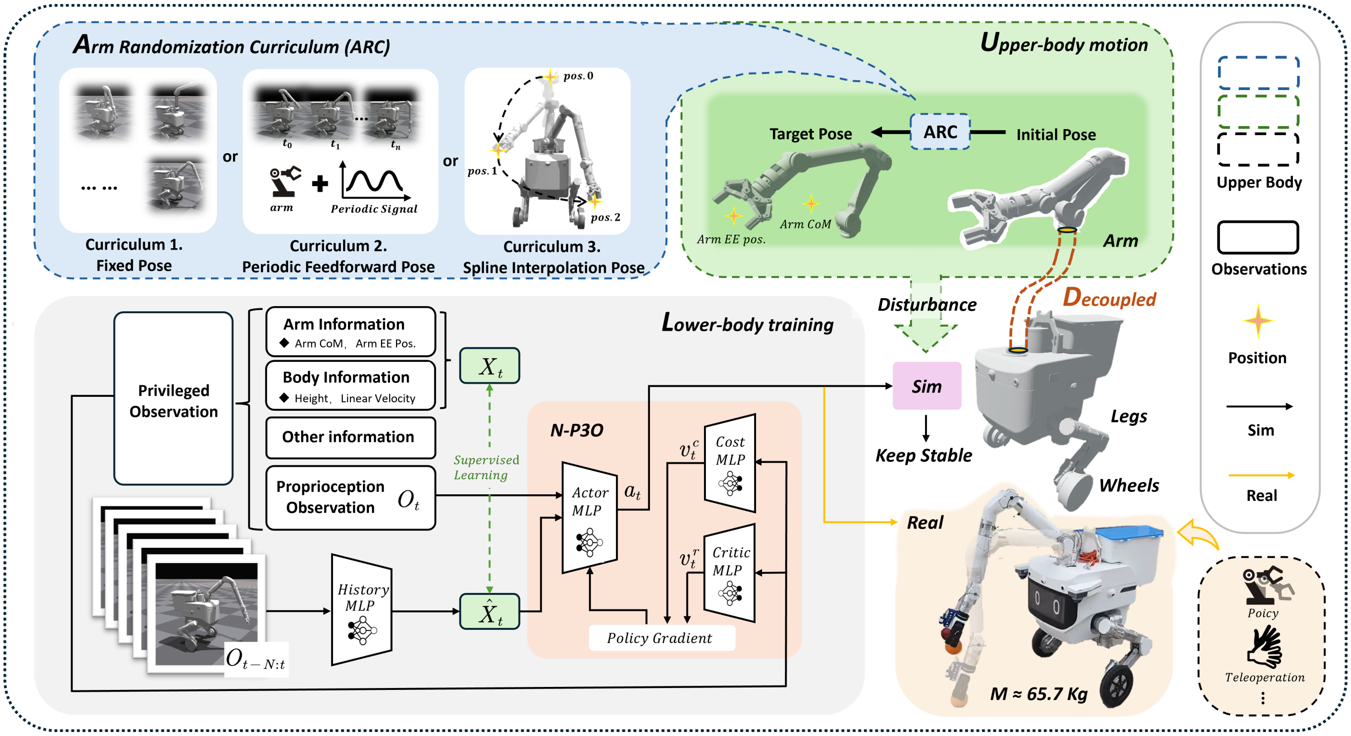

Fig. 2: Overview of the DeLM framework. We define the upper body for robotic arm and the lower body for robot body, legs and wheels. In the ARC section (blue region), we propose three ways in ARC to generate various arm motion poses. In the upper-body motion section (green region), the robotic arm pose changes from the initial pose to the target pose by ARC, which is introduced as disturbances into the lower-body training section (gray region) to make robot learn to keep stable in the simulation. In real robot (yellow region), we use arm manipulation policies and direct teleoperation to control the robotic arm.

has potential for real-world applications in our robot, such as road tidying, sweeping and garbage collection.

In summary, our works are concluded as follows:

- **Decoupled Loco-Manipulation Training Framework**: We propose a learning-based training framework for arm-equipped high-inertia wheel-legged bipedal robots, specifically addressing the challenge of maintaining lower-body platform stability during loco-manipulation tasks.
- **Arm Randomization Curriculum**: We introduce an ARC method through simulating upper-body disturbances on the lower body to further enhance the robustness of the robot stability. Meanwhile, ARC solves the issue about the lack of arm motions data when training.
- **Real-world Deployment**: We propose a parameter calibration methodology aimed at minimizing the sim-to-real gap, thereby enabling successful deployment on our 65.7 Kg wheel-legged robot.

II. Related Work

A. Loco-manipulation Tasks for Legged Robots

Quadrupedal robots have demonstrated impressive capabilities in both locomotion [3]–[7] and loco-manipulation tasks [8], [14]–[18], establishing themselves as robust platforms for mobile manipulation. Several learning-based methods have achieved notable progress in loco-manipulation. For example, ROA [15] developed a Regularized Online Adaptation method to train whole-body controllers for loco-manipulation. UMI-L [18] combined real and simulated data to train arm-equipped quadrupeds. Other works address dynamic, coordinated tasks like badminton [8] and object throwing [16] using arm-equipped robots. For humanoid robots, several studies decouple upper-body control from lower-body locomotion, enabling more stable and coordinated loco-manipulation [11], [12]. Our approach follows a similar decoupling ideology but is implemented on a wheel-legged robotic platform, which presents greater challenges due to its inherent instability.

B. Locomotion Tasks for Wheel-Legged Robots

Compared to locomotion control in quadrupedal robots, achieving stable motion in wheel-legged bipedal robots remains a significant challenge. Previous research has demonstrated promising results using model-based control methods in various tasks, including balance control [19] and jumping motion planning [20], [21]. More recently, learning-based approaches have shown notable success in lightweight robots, such as blind stair climbing [22] and loco-manipulation tasks [23]. Despite these advances, learning-based approaches for loco-manipulation in wheel-legged bipedal robots remain insufficient, lacking a simple and effective method that can be widely applied across diverse scenarios.

C. Locomotion Tasks for High-Inertial Robots

The impact of heavy limbs and payloads on legged robot locomotion and stability has been well studied. Specifically, large masses increase inertial challenges, making whole-body control harder and reducing walking performance [10], [24]. Additionally, dynamic effects such as horizontal wobbling can further destabilize bipedal walking [24]. To mitigate these issues, designing robotic arms with geometric and load constraints is crucial for minimizing negative effects on motion [25]. Moreover, integrating heavy manipulators on quadrupeds requires careful co-optimization of design and control to maintain loco-manipulation stability [26]. In summary, these findings highlight the essential importance of high-inertial properties effects to maintain stable robotic movement.

III. Methodology

A. 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:

$$J_r(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1})\right], \quad (1)$$

where $s_t, s_{t+1} \in \mathcal{S}$, $r \in \mathcal{R}$ and $a_t \in \mathcal{A}$. 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) [27]. 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 [28], [29], which is formulated as follows:

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

where

$$J_{c_i}(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c_i(s_t, a_t, s_{t+1})\right]. \quad (3)$$

B. Decoupled Loco-Manipulation

Our framework DeLM is illustrated in Figure 2. We decouple the whole-body control task into two separate components: upper-body manipulation tasks for the robotic arm and lower-body locomotion tasks for the wheeled legs to maintain balance.

1) ***Manipulation tasks for arm:*** As illustrated in the ARC section of Figure 2, we categorize the ARC into three types: Randomized Fixed Pose (RFP), Randomized Periodic Feedforward Pose (RPFP) and Randomized Spline Interpolation Pose (RSIP). By adjusting the relative weights of these curricula, our method can be adapted to different task requirements. In the upper-body motion section of Figure 2, the robotic arm pose changes from the initial pose to the target pose through ARC. These upper-body motions are applied as disturbances during the lower-body balance training. Notably, we retain all the physical parameters of the robotic arm except for collision constraints. This omission is intentional. By ignoring collision constraints, the arm can reach a wider range of motions. This promotes greater exploration and variability during training. Moreover, although Domain Randomization (DR) includes randomized external forces during training, it cannot fully capture the real impact of arm motions on lower-body balance. This is because DR typically simulates only single external forces, such as pushes or pulls, which are insufficient to represent the complex disturbances caused by arm movement.

2) ***Locomotion tasks for legs:*** As illustrated in the lower-body training section of Figure 2, we adopt a one-stage training approach to accomplish this task. In this method, arm and body information serve as supervised signals to train the historical state estimator [30]. The arm information includes the end effector position $P_{ee}^{arm} \in \mathbb{R}^3$ and the center of mass (CoM) of the robotic arm $P_{com}^{arm} \in \mathbb{R}^3$ in the base frame. The body information includes the robot base linear velocity $v \in \mathbb{R}^3$ and height $h \in \mathbb{R}^1$. Our method employs Normalized-P3O (N-P3O) [28] as the primary network to enforce the physical constraints of the real robot. Within our framework, the historical state estimator takes the last five proprioceptive observations as input and outputs the estimated arm and body information. The N-P3O network then receives this estimated information along with the current proprioceptive observation as input, and outputs the lower-body joint angles as control actions. The lower-body policy is trained under arm-induced disturbances, requiring the robot to learn stable balancing behaviors. Finally, we deploy our policy on a real robot and stably complete loco-manipulation tasks under both arm manipulation policy and teleoperation.

C. Arm Randomization Curriculum

We simplify the arm model by focusing on the first three arm joint angles as the target arm joint angles. $q^{arm,t}$ denotes the target arm pose, $q^{arm,d}$ denotes the default arm pose, $i \in [1, 2, 3]$ denotes the index for the first three joints. All joint angles in this section are expressed in radians. The ARC method includes the following three curricula.

1) ***Randomized Fixed Pose (RFP):*** RFP curriculum is designed to simulate the scenario where the robot starts with the arm in a total random initial pose q^{RFP} in the real world. In this curriculum, the arm is initialized in a fixed pose until it is reset. It is formulated as follows:

$$q_i^{arm,t} = q_i^{RFP} + q_i^{arm,d}, \quad (4)$$

where q^{RFP} follows a uniform distribution. Each joint distribution range is differently designed based on its joint limit. They are formulated as follows: $q_1^{RFP} \sim \mathcal{U}(-0.3, 0.3)$, $q_2^{RFP} \sim \mathcal{U}(0, 2.4)$ and $q_3^{RFP} \sim \mathcal{U}(-1.3, 0)$.

*2) **Randomized Periodic Feedforward Pose (RPFP)**:* The RPFP curriculum is designed to simulate periodic feedforward motions for the robotic arm. This primarily aims to enhance the robot's stability during specific arm motions. In some cases, the curriculum is more important than others in ARC. To better characterize its effects, we categorize RPFP into three distinct types.

(a) Feedforward Trajectory. This is the basic trajectory type used in the RPFP curriculum. To simulate dynamic picking poses and other movements, the target arm pose for each joint follows a sinusoidal signal with randomly sampled frequency f, amplitude a and phase ϕ, where f, a and ϕ follows uniform distribution. It is formulated as follows:

$$\begin{bmatrix} q_1^{RPFP1} \\ q_2^{RPFP1} \\ q_3^{RPFP1} \end{bmatrix} = \begin{bmatrix} a_1 \sin\left(2\pi f_1 t + \phi_1\right) \\ a_2 \cos\left(2\pi f_2 t + \phi_2\right) + 1 \\ -a_3 \cos\left(2\pi f_3 t + \phi_3\right) - 1 \end{bmatrix}, \tag{5}$$

$$q_i^{arm,t} = q_i^{RPFP1} + q_i^{arm,d}, \tag{6}$$

where q_i^{RPFP1} represents the i-th arm joint angle generated by feedforward trajectory method. $f_i \sim \mathcal{U}(0, 0.5)$, $\phi_i \sim \mathcal{U}(0, \pi/2)$ and $a_i \sim \mathcal{U}(q_i^{arm,min}, q_i^{arm,max})$. $q_i^{arm,min}$ and $q_i^{arm,max}$ represent the minimum and the maximum limits of i-th arm joint respectively

(b) Fourier Series Trajectory. This method synthesizes multiple sinusoidal signals with randomized frequencies to approximate arbitrary waveforms, thereby overcoming the constraints imposed by strictly periodic signals on the robot's response. It is mathematically formulated as follows:

$$q_i^{RPFP2} = \sum_{n=1}^{N} a_{i,n} \sin(2\pi f_{i,n} t + \phi_{i,n}), \tag{7}$$

$$q_i^{arm,t} = q_i^{RPFP2} + q_i^{arm,d}, \tag{8}$$

where q_i^{RPFP2} represents the i-th arm joint angle generated by this method. N denotes the total number of variants summed in the Fourier series.

(c) Randomized Pause Time. Interruptions during the robotic arm's motion frequently occur in practical applications. To model this case, we employ time truncation to simulate random pauses in arm movement. Specifically, when the elapsed time t exceeds a random cutoff time t_c, the arm remains fixed in its final pose until the trajectory is reset. It is formulated as:

$$q_i^{RPFP3} = a_i \sin\left(2\pi f_i t' + \phi_i\right), \quad t' = min(t, t_c), \tag{9}$$

$$q_i^{arm,t} = q_i^{RPFP3} + q_i^{arm,d}, \tag{10}$$

where q_i^{RPFP3} denotes the i-th arm joint angle generated by this method. The variable k is uniformly sampled from the distribution $\mathcal{U}(0, 1)$, and the cutoff time t_c is given by $t_c = kt$, where t represents the total duration of the motion.

*3) **Randomized Spline Interpolation Pose (RSIP)**:* RSIP curriculum is designed to enhance the randomness of the arm motions. Considering the entire arm length is approximately 15 cm, we define a hemisphere with a radius of 10 cm which is centered at the robot's base. A target point for the arm's end effector is randomly sampled from this hemisphere. By using Inverse Kinematics (IK), we compute each corresponding arm joint angle, and then apply spline interpolation to generate a continuous arm motion trajectory from the init pose to the target pose. This process can be formally expressed as:

$$q_i^{arm,t} = q_i^{RSIP} + q_i^{arm,d}, \tag{11}$$

where q_i^{RSIP} represents the i-th arm joint angle generated by this method.

TABLE I
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: randomization variants related to physical properties.
Blue: randomization variants related to system delays.
Red: randomization variants related to parameter calibration.

D. Parameter Calibration Method

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 I. Through empirical analysis, we identified three motor parameters that critically impact real-world performance: joint motor friction, damping, and armature.

Some previous works collect real-world data to train actuator network models for the legs [5] in order to reduce the sim-to-real gap. However, this approach increases data collection costs and may amplify errors due to the neural network's sensitivity to training parameter tuning. In contrast, we propose a parameter calibration method tailored to our robot. In simulation, the robot is fixed in the mid-air while a

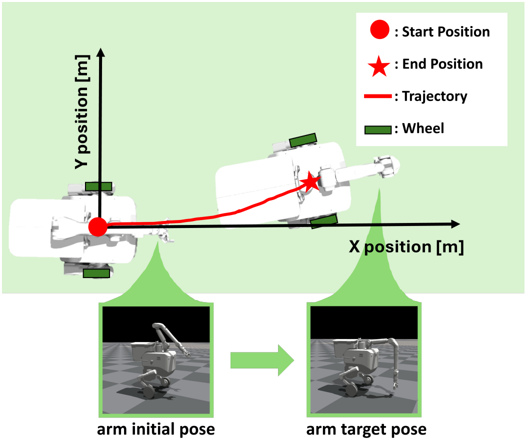

Fig. 3: Top-view trajectories of the robot on a 2D plane. As the robotic arm moves from the initial pose to the target pose, the robot autonomously adjusts its body from the start position to the end position, resulting in a positional deviation. This deviation trajectory is illustrated by red lines.

sinusoidal signal is applied to each joint. A PD controller regulates the signal to drive the legs in a periodic up-and-down motion, during which we record motor feedback data including joint torques, velocities, and angles. In the real world, the same sinusoidal signals are applied to the robot's joints to collect the corresponding motor feedback. By matching the simulation and real-world data, we can efficiently calibrate the initial parameters of each joint motor. These initial parameters and their corresponding DR ranges are highlighted in red in Table I. Despite inherent uncertainties in real-world applications, our approach demonstrates both effectiveness and efficiency.

Our policy is trained in simulation at 200 Hz and deployed on the real robot at 50 Hz. We measured the delay times on the real robot and accordingly designed delay ranges within DR, including torque delay, observation delay, action delay, joint delay and IMU delay.

IV. Experiments and Results

A. Robot Hardware

Our robot is a wheel-legged bipedal robot platform equipped with a 6-DoF (Degrees of Freedom) robotic arm. It has a total mass of approximately 65.7 Kg, a base height of roughly 33 cm, and each wheel has a diameter of about 23 cm. The robot's mass is significantly higher than that of comparable wheel-legged robots, primarily due to its specialized functional and structural design. Each leg is 5-DoF. A fisheye camera is mounted at the front of the robotic arm's grasper.

B. Experimental Setup

1) Observation Space: The body proprioception observations are represented as $O_t \in \mathbb{R}^{25}$, with detailed components listed in Table IV. Regarding privileged observations, in addition to proprioceptive observations as well as arm and body information, the observations also include leg joint accelerations $\ddot{q}^{leg}$, leg joint torques τ^{leg}, base mass m, base CoM, default leg joint positions $q^{leg,d}$ and the joint stiffness and damping coefficients K_p and K_d.

TABLE II
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)	$\exp(-160\|\|R_y\|\|^2)$	0.8
Ang. vel. (y)	$\exp(-50\|\|w_y\|\|^2)$	0.8
Feet distance	$\exp(-100d^{feet})$	0.2
Regularization		
Lin. vel. (z)	$\|\|v_z\|\|^2$	-1e-4
Ang. vel. (xy)	$\|\|w_{xy}\|\|^2$	-0.05
Joint vel.	$\|\|\dot{q}^{leg}\|\|^2$	-5e-4
Joint acc.	$\|\|\ddot{q}^{leg}\|\|^2$	-5e-7
Joint power	$\|\tau^{leg}\dot{q}^{leg}\|$	-1e-8
Action rate	$\|\|a_{t-1} - a_t\|\|^2$	-0.2
Action smoothness	$\|\|a_{t-2} + a_t - 2a_{t-1}\|\|^2$	-0.5
Penalty		
Collision	$\mathbb{I}_{\|\|F_c\|\|>0.1}$	-20.0
Stand still penalty	$\|\|v\|\|^2 \times \mathbb{I}_{\|\|v_{xy}^{cmd}\|\|<0.1}$	-50.0
Orientation mismatch	$\|\|g_{xy}\|\|^2$	-10.0

TABLE III
Cost function design.

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

2) Reward Function Design: In Table II, we classify our reward functions into three categories based on their functionality: task rewards, regularization rewards and penalty rewards. For the feet distance reward, we define the feet distance d^{feet} as follows:

$$d^{feet} = ||P_{xy}^{left} - P_{xy}^{right}||, \quad (12)$$

where P_{xy}^{left} and P_{xy}^{right} denote the xy positions of the left and right feet respectively. This reward encourages the robot

TABLE IV
Proprioception observations.

Term	Description	Obs Scale
v_x^{cmd}	Command linear velocity in x-axis	1.0
w_z^{cmd}	Command angular velocity in z-axis	1.0
h_z^{cmd}	Command height in z-axis	1.0
q^{leg}	Joint angles of the legs	0.02
$\dot{q}^{leg}$	Joint velocities of the legs	1.0
$\dot{q}^{wheel}$	Joint velocities of the wheels	1.0
a_{t-1}	Previous action	1.0
w	Angular velocities of the body	1.0
R	Euler angles of the body	1.0

TABLE V
ABLATION STUDY RESULTS ON LOCO-MANIPULATION TASKS METRICS.

Terms	MBDD (m)↓	MWDD$_l$ (m)↓	MWDD$_r$ (m)↓	MBED$_y$ (deg)↓	MBED$_z$ (deg)↓	SR (%)↑
Evaluated under the RFP						
w/o both	0.271 ± 0.004	0.246 ± 0.005	0.252 ± 0.004	0.489 ± 0.122	1.280 ± 0.593	87.1%
w/o AEI	**0.095 ± 0.004**	0.130 ± 0.006	**0.059 ± 0.004**	0.572 ± 0.113	6.742 ± 2.846	94.1%
w/o ARC	0.241 ± 0.006	0.178 ± 0.006	0.264 ± 0.007	0.566 ± 0.200	2.827 ± 1.170	89.2%
ours	0.123 ± 0.005	**0.099 ± 0.005**	0.105 ± 0.005	**0.307 ± 0.116**	**0.815 ± 0.207**	**94.6%**
Evaluated under the RPFP						
w/o both	0.564 ± 0.010	0.524 ± 0.010	0.583 ± 0.010	0.476 ± 0.204	3.637 ± 2.045	66.5%
w/o AEI	0.317 ± 0.004	0.398 ± 0.004	0.251 ± 0.004	0.553 ± 0.190	9.012 ± 5.425	81.6%
w/o ARC	0.549 ± 0.010	0.487 ± 0.010	0.594 ± 0.010	0.339 ± 0.186	3.532 ± 1.369	67.4%
ours	**0.231 ± 0.006**	**0.203 ± 0.006**	**0.246 ± 0.006**	**0.325 ± 0.213**	**2.799 ± 0.992**	**86.4%**
Evaluated under the RSIP						
w/o both	0.191 ± 0.018	0.153 ± 0.018	0.175 ± 0.018	0.674 ± 0.198	1.882 ± 0.906	61.6%
w/o AEI	0.115 ± 0.020	0.138 ± 0.020	0.047 ± 0.020	0.547 ± 0.153	4.838 ± 2.442	76.8%
w/o ARC	0.166 ± 0.014	0.101 ± 0.014	0.182 ± 0.014	0.703 ± 0.198	2.773 ± 0.831	66.5%
ours	**0.073 ± 0.005**	**0.041 ± 0.005**	**0.052 ± 0.005**	**0.232 ± 0.216**	**1.015 ± 0.352**	**85.3%**

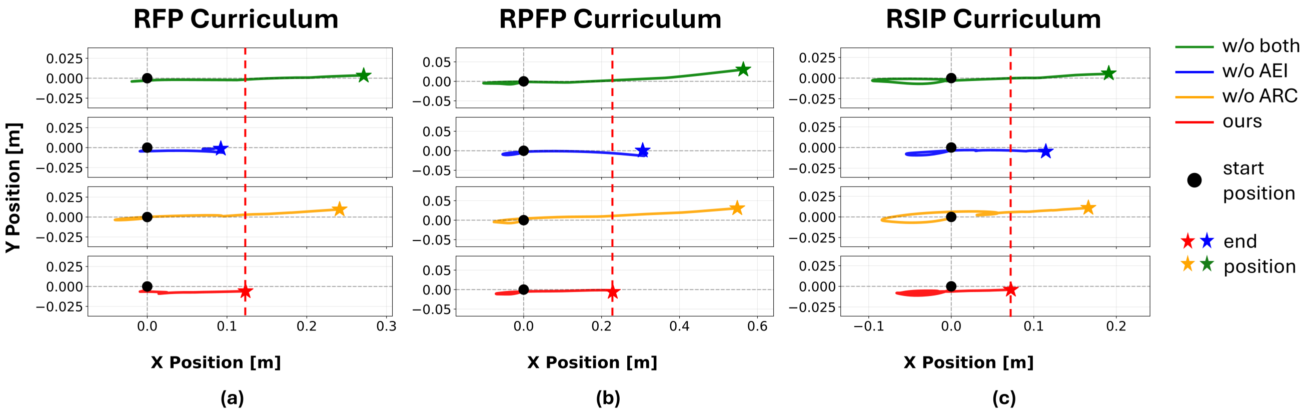

Fig. 4: Ablation study results about robot's top-view trajectories on a 2D plane. The red dashed line represents the end position achieved by our method in comparison to others.

to minimize the use of its fore and hind legs to maintain stable support. Regarding the collision reward, we define the indicator function $\mathbb{I}_{\{.\}}$, which returns 1 if the specified condition is met, and 0 otherwise. Here, F_c denotes the contact force, and g_{xy} represents the projection of gravity along the x and y axes.

3) Cost Function Design: In Table III, we define four constraint limits: joint position limits q_{lim}^{leg}, joint velocity limits $\dot{q}_{lim}^{leg}$, joint torque limits τ_{lim}^{leg} and acceleration limits $\ddot{q}_{lim}^{leg}$. 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 in N-P3O.

C. Ablation Study

The ablation study evaluates two critical components of our method: Arm Estimated Information (AEI) and the ARC. AEI includes P_{ee}^{arm} and P_{com}^{arm}, which provide essential information about the arm's pose and CoM respectively. As the robotic arm moves, its CoM will deviate accordingly. These changes cause involuntary movement of the robot's lower body in the direction of the arm's motion. Figure 3 illustrates the corresponding top-view trajectory. To quantify the positional deviation, we define the Mean Base Distance Deviation (MBDD) as the average Euclidean distance between the robot's start and end positions on the 2D plane:

$$\text{MBDD} = ||P_{xy}^{start} - P_{xy}^{end}||, \tag{13}$$

where P_{xy}^{start} and P_{xy}^{end} denote the initial and final positions of the base on the 2D plane, respectively. Similarly, the Mean Wheel Distance Deviation (MWDD) is defined as the corresponding average distance deviation for each wheel, denoted by the subscripts $(.)_l$ and $(.)_r$, respectively.

To account for potential rotations of the robot, we introduce the Mean Base Euler Deviation (MBED), which quantifies the maximum deviation in Euler angles from zero. Finally, the Success Rate (SR) is defined as the proportion of environments in which the MBDD remains below a predefined deviation threshold of 25 cm. This threshold is chosen based on the physical characteristics of the robot, particularly considering the diameter of the wheels.

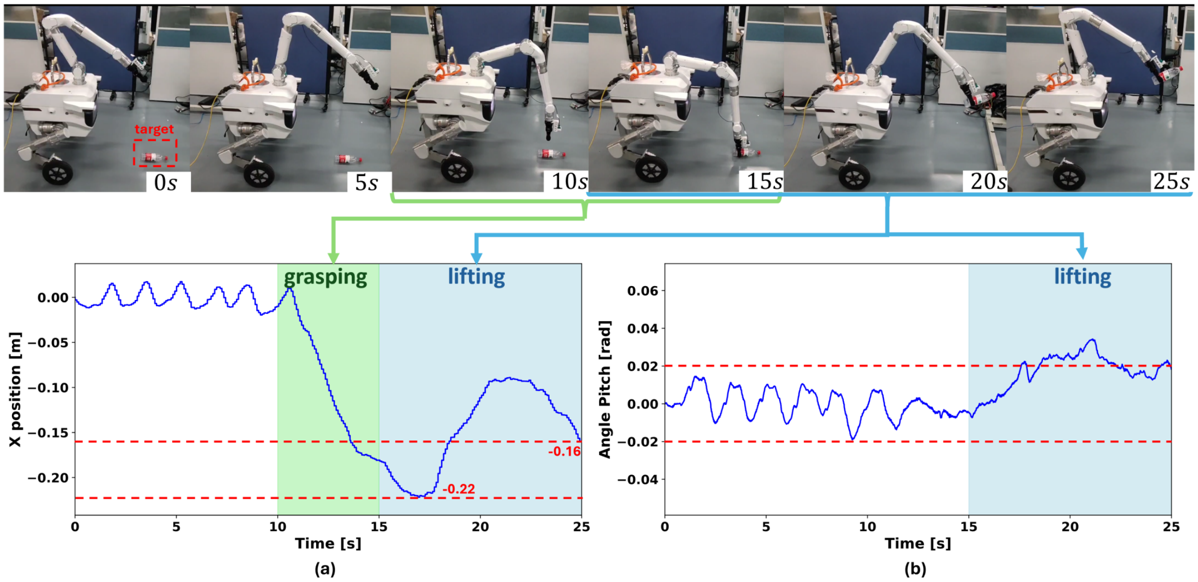

Fig. 5: Real world experimental results for MBDD$_x$ in (a) and MBED$_y$ in (b). The green and blue regions indicate the durations of the robotic arm's downward grasping and upward lifting motions, respectively. The red dashed lines represent the tolerable range in (b).

We use Isaac Gym [31] as our simulator to train 4,096 parallel environments simultaneously, each one was trained for 11,000 iterations on an NVIDIA RTX 4070 GPU. The ablation variants are defined as follows:

- w/o ARC: Policy trained without ARC.
- w/o AEI: Policy trained without AEI.
- w/o both: Policy trained without both ARC and AEI.
- ours: Policy trained using both ARC and AEI as proposed.

We evaluate our approach across 2,000 environments using random seeds over five rounds. For experimental conditions, we evaluate our model by using three types of methods in ARC to randomly generate robotic arm motions. Specifically, the proportions of types (a), (b) and (c) in the RPFP curriculum are set to 0.3, 0.3 and 0.4 respectively.

Our ablation results are summarized in Table V. In the RFP evaluation, the robot is required to maintain stability under randomly fixed poses of the robotic arm. According to the MBDD metric, our method yields a value of 0.123 m, which is slightly higher than that of the variant without AEI (0.095 m), indicating a marginal increase in positional deviation under this specific measure. However, the variant without AEI exhibits a notable orientation deviation, with a 7 cm discrepancy between MBDD$_l$ and MBDD$_r$, as well as a substantial yaw deviation of $6.742°$ in MBED$_z$. In contrast, our method achieves a much smaller left-right deviation of only 0.6 cm and a significantly lower yaw deviation of $0.815°$. Furthermore, our final SR reaches 94.6%, outperforming all other variants.

In the RPFP and RSIP evaluations, the robot must dynamically respond to continuous variations in the robotic arm's pose. In these evaluations, our method achieves SRs of 86.4% and 85.3% and represents improvements of 19.5% and 23.7% respectively, over the variant without ARC and AEI. The MBDD scores, 0.231 m and 0.073 m, are the best among all evaluated methods (highlighted in bold in the table). Although the variant without ARC achieves the SR of 89.2% in the static arm task, its performance drops to 67.4% and 66.5% in the dynamic arm tasks, indicating that incorporating ARC improves the stability of loco-manipulation tasks. While the variant without AEI performs reasonably well under all three evaluation conditions, it consistently exhibits the largest yaw deviation in the MBED$_z$ metric due to the lack of accurate estimation of the arm's pose. This suggests that the robot tends to compensate by rotating its body to maintain stability. Such behavior is generally considered undesirable.

To provide an intuitive understanding of the data in Table V, we use the top-view trajectories in Figure 4 to show the robot's positional deviations along the x and y axes. Initialized 0.36 m above the ground, the robot first shifts backward, then deviates further due to the robotic arm's movement, and finally stabilizes at the end position. The red dashed line marks our method's end position on the x-axis compared to other variants. In (a), only the variant without AEI shows smaller deviation, but in (b) and (c), our method outperforms all others.

D. Real Robot Experiment

We select a representative experiment from multiple real-world trials for detailed analysis. The robot is evaluated by picking up a water bottle from the ground via teleoperation. As shown in Figure 5 (a), the deviation remains small and oscillates near the initial position during the first 10 seconds. As the gripper descends deeper, the robot performs a gradual backward movement of approximately 0.22 m (green region) to adapt to changes in the arm's CoM. After grasping the object at the 15-second mark, the arm lifts, causing a deviation in its CoM. To compensate for this, the robot moves forward (blue region). Ultimately, the robot's final position deviates by only 0.16 m from the initial position. Figure 5 (b) shows the pitch angle deviation from zero. The pitch angle fluctuates mildly within ±0.02 radians, as indicated by the red lines, with only a slight deviation beyond this range in the final stage (blue region).

V. Conclusion

In this work, we propose a decoupled training framework, DeLM, designed to enhance motion stability during manipulation tasks for high-inertia wheel-legged robots. We introduce the ARC training method to address the challenge of insufficient lower-body stability during manipulation, which significantly improves the robustness of lower-body balance control. Additionally, we develop a parameter calibration method to bridge the sim-to-real gap, providing valuable insights for future deployments. However, our method still faces challenges in highly unstructured terrains. In future work, we plan to extend our framework to handle more complex terrains while achieving coordinated control of both locomotion and manipulation tasks.

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] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, "Learning quadrupedal locomotion on deformable terrain," *Science Robotics*, vol. 8, no. 74, p. eade2256, 2023.

[4] 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*. PMLR, 2022, pp. 91–100.

[5] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[6] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024.

[7] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 443–11 450.

[8] 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.

[9] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter, "Learning robust autonomous navigation and locomotion for wheeled-legged robots," *Science Robotics*, vol. 9, no. 89, p. eadi9641, 2024.

[10] T. Zhang, L. Yue, H. Zhang, L. Zhang, X. Zeng, Z. Song, and Y.-H. Liu, "Whole-body control framework for humanoid robots with heavy limbs: A model-based approach," *arXiv preprint arXiv:2506.14278*, 2025.

[11] 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.

[12] 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.

[13] 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.

[14] I. Dadiotis, M. Mittal, N. Tsagarakis, and M. Hutter, "Dynamic object goal pushing with mobile manipulators through model-free constrained reinforcement learning," *arXiv preprint arXiv:2502.01546*, 2025.

[15] 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.

[16] 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.

[17] J.-P. Sleiman, F. Farshidian, and M. Hutter, "Versatile multicontact planning and control for legged loco-manipulation," *Science Robotics*, vol. 8, no. 81, p. eadg5014, 2023.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] T. Kamimura and A. Sano, "Effect of the dynamics of a horizontally wobbling mass on biped walking performance," *arXiv preprint arXiv:2209.14515*, 2022.

[25] Z. Zhu, Z. Luo, Y. Zhu, T. Jiang, M. Xia, S. Chen, and B. Jin, "Bio-inspired design and inverse kinematics solution of an omnidirectional humanoid robotic arm with geometric and load capacity constraints," *Journal of Bionic Engineering*, vol. 21, no. 2, pp. 778–802, 2024.

[26] A. Rigo, M. Hu, J. Ma, S. K. Gupta, and Q. Nguyen, "Design and control co-optimization for dynamic loco-manipulation with a robotic arm on a quadruped robot," *Journal of Mechanisms and Robotics*, vol. 17, no. 5, p. 051006, 2025.

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

[28] 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.

[29] 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.

[30] G. Ji, J. Mun, H. Kim, and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4630–4637, 2022.

[31] 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.