

Learning Visual Quadrupedal Loco-Manipulation from Demonstrations

Zhengmao He^{1,2}, Kun Lei¹, Yanjie Ze¹, Koushil Sreenath³, Zhongyu Li³, Huazhe Xu^{1,4}
<https://zhengmaohe.github.io/leg-manip>

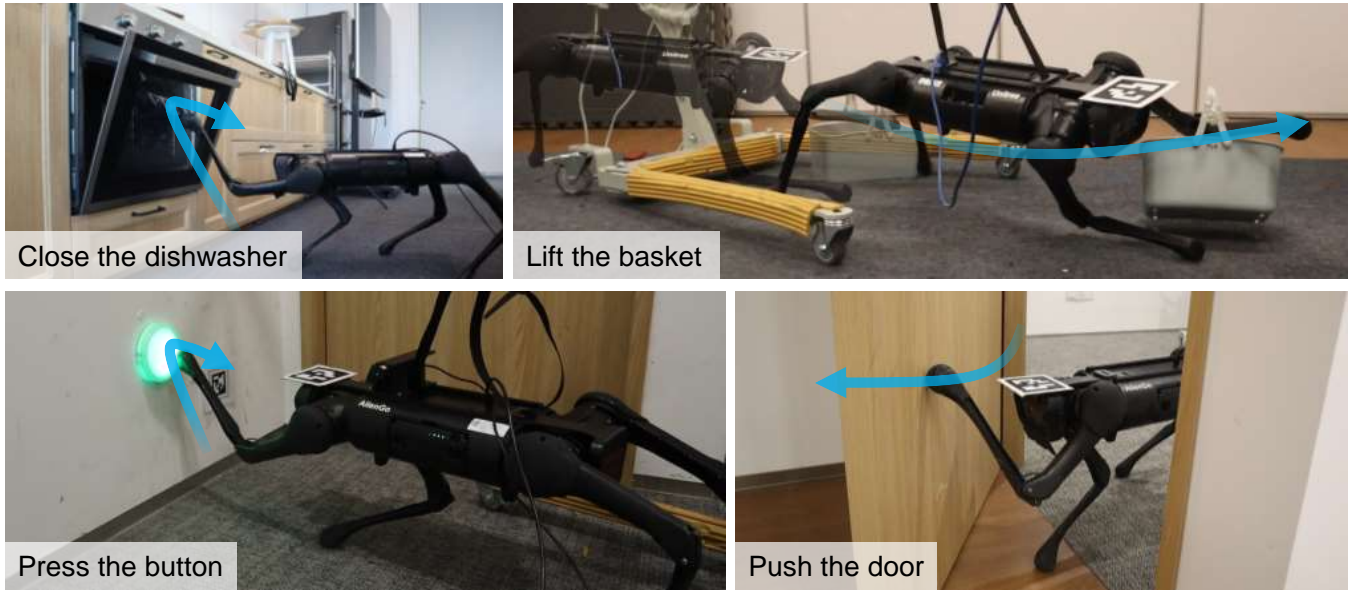


Fig. 1: We present a hierarchical learning framework to learn general loco-manipulation skills for quadruped robots. The framework enables a Unitree Aliengo robot to perform diverse skills in the real-world, including lifting baskets, pressing buttons, opening doors, and closing dishwashers, all while maintaining stable locomotion over a long distance. Videos are on the project website.

Abstract—Quadruped robots are progressively being integrated into human environments. Despite the growing locomotion capabilities of quadrupedal robots, their interaction with objects in realistic scenes is still limited. While additional robotic arms on quadrupedal robots enable manipulating objects, they are sometimes redundant given that a quadruped robot is essentially a mobile unit equipped with four limbs, each possessing 3 degrees of freedom (DoFs). Hence, we aim to empower a quadruped robot to execute real-world manipulation tasks using only its legs. We decompose the loco-manipulation process into a low-level reinforcement learning (RL)-based controller and a high-level Behavior Cloning (BC)-based planner. By parameterizing the manipulation trajectory, we synchronize the efforts of the upper and lower layers, thereby leveraging the advantages of both RL and BC. Our approach is validated through simulations and real-world experiments, demonstrating the robot’s ability to perform tasks that demand mobility and high precision, such as lifting a basket from the ground while moving, closing a dishwasher, pressing a button, and pushing a door.

I. INTRODUCTION

The four 3-DoF limbs of a quadrupedal robot, and its 6-DoF floating base torso can provide a wide workspace

and flexibility for manipulation tasks. This insight underscores the inherent versatility of quadruped robots as loco-manipulators, capable of integrating locomotion and manipulation without additional robotic arms.

Previous research on using quadruped robot legs for manipulation has enabled robots to execute tasks such as pressing buttons, dribbling balls, and ball shooting [1]–[4]. However, these studies still face certain limitations: (i) designs are often task-specific, leading to poor adaptability for different tasks [1]–[4]; (ii) manipulation is coarse, lacking precision [2], [3]; (iii) methods focus on static manipulation, not using the robots’ mobile abilities [1], [3]. Our approach addresses these issues with a general framework for versatile tasks, precise control, and mobile manipulation.

Utilizing the legs of quadrupedal robots for general manipulation tasks that require a large workspace and high precision is considerably more complex than merely combining locomotion with manipulation. This complexity introduces several unique challenges. First, as highly nonlinear systems, loco-manipulators lack the inherent stability of conventional wheeled mobile manipulators. This issue is further exacerbated when legs are used for manipulation and the system is highly underactuated. These characteristics render the problem challenging not only in terms of locomotion but also in manipulation. Second, the challenge becomes even more complex when incorporating vision for manipulation.

¹Shanghai Qi Zhi Institute.

²The Hong Kong University of Science and Technology (Guangzhou). zhe037@connect.hkust-gz.edu.cn

³University of California, Berkeley. {zhongyu_li, koushil}@berkeley.edu

⁴Tsinghua University, IIS. huazhe_xu@mail.tsinghua.edu.cn

Robot vision is crucial for versatile manipulation but is difficult to utilize effectively due to its high dimensionality and the significant gap between simulation and real-world application.

To tackle these challenges, we develop a hierarchical framework that merges Behavior Cloning (BC) with Reinforcement Learning (RL). Our framework enables the seamless integration of locomotion and manipulation skills, thereby extending the capabilities of legged robots beyond mere locomotion. The contributions of this work are multifaceted:

- We carefully design and implement a hierarchical learning framework that harnesses the strengths of both BC and RL, which capitalizes on the efficiency of BC in learning manipulation tasks from demonstrations, as well as the strength of RL in real-time control of high dimensional dynamic systems.
- Our high-level planner employs a diffusion-based BC policy to efficiently learn a variety of manipulation skills from demonstrations, marking a novel approach in whole-body loco-manipulation.
- We parameterize the manipulation trajectory of the end-effector for better integration of RL and BC. This method also enables easy data collection through parallel simulations, eliminating the need for teleoperation and the challenges of aligning human actions with legged robots.
- To evaluate the performance of our algorithm, we design multiple tasks. These tasks are grounded in practical requirements and are devised to comprehensively evaluate the multifaceted capabilities of loco-manipulators.

Collectively, these contributions represent a novel approach to bridging the gap between manipulation and locomotion.

II. RELATED WORK

A. Mobile Manipulation

Traditional mobile manipulator robots typically feature a high-DoF mechanical arm mounted on top of a wheeled chassis [5]–[11]. The low DoF of wheeled platforms lead to robust mobility performance, allowing end-to-end BC to be effective for them in acquiring complex mobile manipulation skills [7], [8]. However, this advantage is counterbalanced by their limitation to flat terrains, which significantly restricts their application scenarios. Legged robots, on the other hand, can readily overcome this limitation.

B. Legged Locomotion

In recent years, significant advancements have been made in the locomotion capabilities of legged robots. Model-based approaches allow for precise modeling of the robot and environment, enabling robots to achieve robust locomotion skills [12]–[14]. Model-free RL has empowered quadrupedal robots to navigate challenging terrains [15]–[20]. Through fine-tuning in the real world, robots can walk in some terrains they have not encountered before [21], [22]. By leveraging expert demonstrations to learn motion priors, robots are able

to learn various styles of locomotion [23]–[26]. However, alongside the development of advanced locomotion skills, it is also necessary to cultivate manipulation skills to facilitate their integration into human life.

C. Loco-Manipulation

Manipulation skills for legged robots have been greatly improved recently. Some researchers choose to augment robots with additional hardware on their backs [27]–[30], which significantly increases costs and the additional weight compromises the robot’s locomotion abilities. Others try to use the robot’s legs to perform manipulation tasks. However, these methods have several drawbacks and limitations. (i) The methods are designed for specific tasks, often restricted to predefined sequences of actions [1]–[4]. In contrast, we propose a general framework in which a single agent is trained to solve a series of tasks; (ii) The manipulation is very coarse and does not allow for fine-grained manipulation [2], [4]. By contrast, our method can train a low-level controller to achieve precise control for the legs; (iii) Performing static manipulation, failing to leverage the inherent dynamic capabilities of legged robots [3]. In contrast, our method enables robots to carry out manipulation tasks while preserving their locomotion abilities; (iv) It is limited to 3-DoF point tracking, which falls short for more complex manipulation tasks. Additionally, the authors only implemented a low-level controller, necessitating human teleoperation for the execution of all tasks [31]. Conversely, our method, utilizing a visual planner, empowers the robot to autonomously execute complex daily manipulation tasks, by tracking both trajectories and orientations.

III. HIERARCHICAL LEARNING FRAMEWORK

In this section, we introduce our hierarchical learning framework, utilizing BC at the high-level manipulation planning and RL at the low-level joint position control, to enable the robot to achieve versatile loco-manipulation skills.

A. Overview

As illustrated in Fig. 2, motivated by previous work [2], [3], we decompose the loco-manipulation of the robot into two parts: the high-level planner predicts the desired trajectory parameters for the end-effector, while the low-level controller enables the robot to achieve pose tracking with end-effector. The trajectory parameters define the desired pose of the end-effector and identify the legs functioning as manipulators with a manipulator flag f .

To develop the high-level planner π_p , we use the trained control policy π_c to collect expert demonstration data. Specifically, we design expert trajectory parameters for different tasks and make the robot track them in simulation while collecting robot states, point clouds, and trajectory parameters. Through large-scale parallel simulation, we can collect more than 100 expert demonstrations in 3 minutes. Utilizing the collected expert demonstrations, we develop a high-level planner with DP3 [33], which takes point clouds

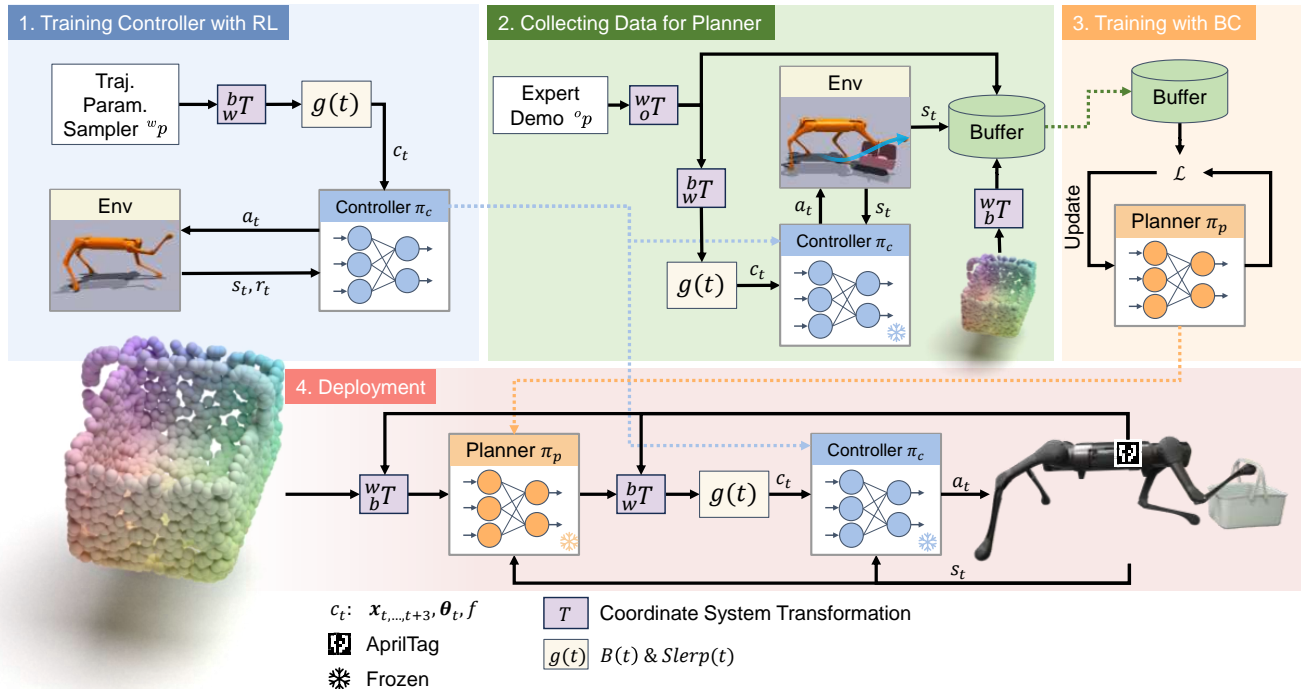


Fig. 2: (1) We train a control policy π_c that enables an end-effector to follow curves defined by Bézier control points and weight while maintaining stable locomotion with the other three legs. (2) We use the trained controller to collect expert data. We design manipulation trajectories for different tasks and collect demonstrations through parallel simulation. (3) We use the collected expert data and diffusion-based BC to train the planner. (4) In the deployment phase, we use the Realsense D435 to obtain the point cloud, and use an external camera to locate the pose of the robot based on AprilTag [32] for trajectory parameter and point cloud coordinate system transformation. These inputs are sequentially fed into the planner and controller, enabling the robot to perform whole-body loco-manipulation tasks.

and robot states as input and outputs trajectory parameters. The frequency of the high-level planner is 10 Hz.

We develop a low-level controller π_c , which allows the robot to track a 6-DoF target trajectory with any forelimb while maintaining stable walking. This is achieved by training a policy whose inputs include the manipulator flag, the current desired end-effector pose, the current target point of the end-effector calculated by the rational Bézier curve, and the next three target points. The output is desired robot joint positions $q_{tm}^d \in \mathbb{R}^{12}$. This control policy runs at 50 Hz, and the desired torque τ of the robot is obtained through a PD controller from the desired joint positions.

B. Trajectory Parameterization

To fully capitalize on the strengths of RL and BC and ensure their effective cooperation, we parameterize the manipulation trajectory of the end-effector. These trajectory parameters p serve as both the output of the high-level planner and are utilized to calculate the input for the low-level controller. Our trajectory parameters include the manipulation flag f , 6-order rational Bézier curve parameters, and the target orientation of the start and end points, which we further describe below.

a) Manipulator Flag: We use any one of the forelimbs as the manipulator, which is specified by a binary manipulation flag f , where 0 stands for the Front Left (FL) foot and 1 for the Front Right (FR) foot.

b) Desired Position Trajectory: In this paper, we use rational Bézier curves to represent the manipulation trajectory. Unlike traditional Bézier curves, which are defined solely by control points, rational Bézier curves introduce adjustable weights for these control points, enabling the seamless combination of curves and polylines while still maintaining smoothness by weighting the control points. This enables it to parametrically and flexibly represent different trajectories. We consider the foot toe as the end-effector, and the trajectory of the end-effector in 3D space is specified by rational Bézier curves:

$$B_n(t) = \frac{\sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} \mathbf{p}_i w_i}{\sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} w_i}. \quad (1)$$

Where $\mathbf{p}_i \in \mathbb{R}^{3 \times 1}$ represents the control points, $w_i \in \mathbb{R}$ denotes the weights of the control points, introducing an additional degree of control not present in traditional Bézier curves, and $n+1$ is the number of parameters, with $n=6$ chosen for this paper.

c) Desired Orientation: Our manipulation trajectory uniquely specifies the orientation, a capability not realized in previous work [1], [3], [31]. This advancement enables our robot to execute more complex manipulation tasks.

We employ spherical linear interpolation (SLERP) to calculate the desired orientation, which is a method for smooth interpolation between two orientation vectors:

$$Slerp(\mathbf{q}_0, \mathbf{q}_1, t) = \frac{\sin[(1-t)\theta] \cdot \mathbf{q}_0 + \sin(t\theta) \cdot \mathbf{q}_1}{\sin \theta} \quad (2)$$

Where \mathbf{q}_0 represents the target orientation of the starting point, and \mathbf{q}_1 represents the end orientation of the endpoint. θ is the angle subtended by the arc.

In Eq. (1) and (2), $t \in [0, 1]$ is the phase time that is scaled according to the time span of the trajectory.

d) The Choice of Reference Frame for Parameters:

Tracking points directly with a mobile robot in RL can face motion asymmetry issues [17], [34], [35], leading us to use the body frame for trajectory parameters ${}^b p$ in our low-level control policy. Our high-level planner operates at a lower frequency than the controller, which could cause tracking errors due to time lags between outputs. To prevent this, we use the world frame for both input point clouds ${}^w \mathbf{P}$ and output trajectory parameters ${}^w p$, avoiding continuous error corrections and swaying motions. Furthermore, when collecting expert data, we randomize robot and object poses, ensuring trajectory parameters ${}^o p$ focus on the object by representing them in the object frame and then converting to the world frame based on the pose of the object.

C. Learning Visual Manipulation Planning by BC

In this section, we will provide a detailed introduction to the high-level planner, which constitutes the upper layer of the framework. It processes input from point clouds and robot proprioception, and outputs the manipulation trajectory parameters for the end-effector.

1) Framework: Our planner π_p utilizes DP3 [33] as the backbone, which is a diffusion-based 3D visuomotor policy that can efficiently process 3D data and learn the manipulation trajectory of end-effector from expert demonstrations.

a) Input: We utilize proprioceptive data of the robot state s_t and visual point cloud data ${}^w \mathbf{P} \in \mathbb{R}^{n \times 3}$ as inputs. The visual data is captured by a depth camera mounted behind the robot and on its head, which is then transformed into point clouds in the world frame. During the manipulation process, we randomly sample $n = 768$ points to form the point clouds.

b) Output: Our planner generates the parameters ${}^w p$ for the manipulation trajectory of the end-effector, represented in the world frame. These parameters can be used to calculate the target point, target orientation, and the manipulator flag f at each moment of the manipulation process. This information is then fed into the subsequent low-level controller.

2) Expert Demonstration Collection: We represent the expert demonstration data as $\mathcal{D} = \{\xi_0, \xi_1 \dots \xi_n\}$, where each trajectory $\xi_i = \{({}^w \mathbf{P}_i, s_i, {}^w p_i)\}$ is a sequence of point cloud observations ${}^w \mathbf{P}$, the robot proprioceptive state s , and trajectory parameters ${}^w p$.

As illustrated in Fig. 3, we designed expert trajectory parameters ${}^o p$ for each task. To enable the robot to learn to manipulate objects placed in different poses, we randomized the positions and yaw axis angles of the objects. The expert trajectory parameters are represented in the object frame, allowing them to accurately follow the object. During the collection of expert demonstrations, the object frame is transformed to the world frame based on the object’s pose.

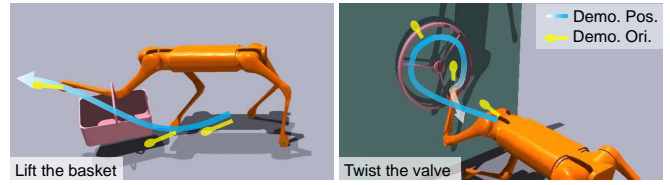


Fig. 3: Expert Demonstration. For different objects and tasks, we designed expert trajectory parameters and randomized the poses of the objects. The blue gradient curve represents the demonstration trajectory, and the yellow arrow indicate the demonstration orientation.

We gather expert data by having robots follow expertly designed trajectories, utilizing large-scale parallel simulations in IsaacGym [36]. This method enables the rapid collection of a significant volume of data. The collection time does not increase with the volume of data; it primarily depends on the size of the Video Random Access Memory (VRAM). In this paper, we collect 200 timesteps of expert demonstration data for each task, with 100 trajectories collected per task, taking approximately three minutes. The success rates of the data collected are shown in Tab. III, indicating that expert demonstrations collected by our method achieve relatively high success rates across most tasks.

Despite the collection of expert data in simulations, our method achieves seamless sim-to-real deployment in actual environments, thanks to the carefully designed hierarchical structure and the incorporation of point clouds as visual input.

D. Learning Joint Position Control by RL

In this section, we will introduce the training of the control policy π_c that enables the robot to track any spatial trajectory with its end-effector while maintaining stable movement with three legs, and the end-effector can handle unknown forces attached to it to ensure the smooth execution of subsequent control tasks. π_c is trained using the RL algorithm, Proximal Policy Optimization [37].

1) Environment: We train the robot to perform locomotion tasks with three legs, while one leg tracks a given trajectory. The target trajectories are randomly generated within an $4m \times 4m$ square area around the robot, ensuring that the robot learns omnidirectional movement and end-effector tracking tasks. The episode length is 20 seconds, during which the robot is encouraged to track the target trajectories while maintaining stable movement.

a) Policy Input: As illustrated in Fig. 2, the input of the control policy is \mathbf{o}_t including past 15-timesteps history of the proprioception states s_t and manipulation command \mathbf{c}_t . The first part is the proprioception states s_t , including the gravity vector in the body frame \mathbf{g}_t , robot joint position $q_t \in \mathbb{R}^{12}$, robot joint velocity $\dot{q}_t \in \mathbb{R}^{12}$, and action $a_{t-1} \in \mathbb{R}^{12}$. The second part is the manipulation command \mathbf{c}_t calculated by (1) and (2), including manipulate flag f , desired point $\mathbf{x}_t^d \in \mathbb{R}^3$, following desired points $\mathbf{x}_{t+1, t+2, t+3}^d$ of the next 3 timesteps and desired orientation θ_t^d . The inputs for both the actor and critic also encompass privileged information, which includes the robot’s velocity and the positions of the

TABLE I: Reward terms for trajectory tracking.

Term	Expression	Weight
pos tracking	$\exp\{- \mathbf{x}_{xy} - \mathbf{x}_{xy}^d ^2 / \sigma_{x,xy}\}$	0.8
pos tracking	$\exp\{- \mathbf{x}_z - \mathbf{x}_z^d ^2 / \sigma_{x,z}\}$	0.8
ori tracking	$\exp\{-(1 - (\theta \cdot \theta^d)) / \sigma_\theta\}$	0.3
end-effector accelerations	$ \ddot{\mathbf{x}}_{ee} ^2$	-5
body accelerations	$ \ddot{\mathbf{x}}_{base} ^2$	-5

TABLE II: Randomization range of trajectory parameters.

Parameter	Range	Unit
Bézier Parameters $\mathbf{p}_{x,y}$	$[-2.0, 2.0]$	m
Bézier Parameters \mathbf{p}_z	$[0.01, 1.2]$	m
Bézier Parameters w	$[1, 2000]$	1
Target Orientation $\mathbf{q}_{\phi,\psi}$	$[0, 2\pi]$	1
Target Orientation $\cos(\mathbf{q}_\theta)$	$[0.0, 1.0]$	1

end-effector as predicted by the state estimator [38]. This estimator takes the same 15-timesteps history observation as its input.

b) Action Space: The action \mathbf{a}_t of the control policy at time step t is the desired joint position $q_m^d \in \mathbb{R}^{12}$. These are passed through a low-pass filter followed by joint-level PD controller to obtain the motor torques $\tau \in \mathbb{R}^{12}$.

c) Reward: We use three types of rewards: a tracking reward for achieving end-effector tracking, a stability term to train the robot’s stability, and a smoothness term to ensure smoother movements of the robot.

As shown in Table I, to precisely follow the target trajectory, our tracking reward includes both position and orientation tracking. Position tracking is determined by comparing the current position of the end-effector with the target position, with the scale in the z -axis direction amplified fivefold to promote leg lifting. Meanwhile, the orientation tracking error is calculated based on the angle difference between the target orientation and the current orientation of the end-effector.

2) Domain and Command Randomization: To address the issue of varying loads on the end-effector during different tasks and the uncertainty of dynamics parameters when deployed in the real world, we randomized the dynamics parameters of the robot and the environment during the training process.

Furthermore, to enable the robot to accurately follow the target trajectory and orientation while moving omnidirectionally and stably, we randomized the manipulation trajectory parameters, as shown in Table II. Note that the Bézier control points are generated within the range of $x, y \in [-2.0, 2.0]$, but this does not limit the robot’s range of motion because the Bézier control points will be transformed to be represented in the body frame. This allows the robot to learn to move within an infinite range.

IV. DESIGN OF TASKS FOR LOCO-MANIPULATION

There are numerous robotic manipulation benchmarks available [39]–[42], but the majority focus on fixed-base robot manipulation. Currently, there is no suitable benchmark for assessing loco-manipulation performance. To evaluate the effectiveness of our algorithm and to provide a valuable

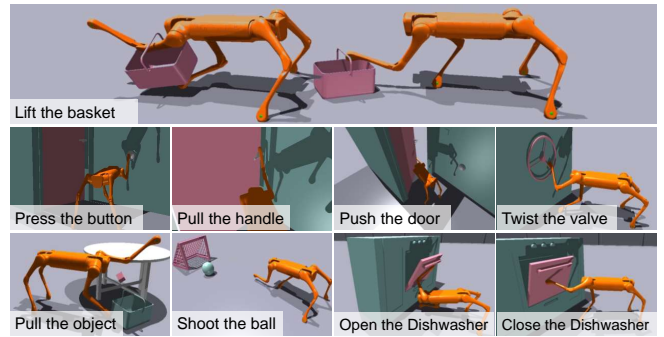


Fig. 4: Overview of the 9 loco-manipulation tasks we train our robot to accomplish. These tasks are designed to cover a large scope of the non-prehensile manipulations tasks that can be realized by the robot’s leg.

reference for the community, we have designed a set of tasks specifically for loco-manipulation tasks, as shown in Fig. 4.

A. Design Principles

Based on the application scenarios and working range of legged robots, we designed a set of tasks. Solving these tasks requires the robot to perform various skills, such as pushing, tapping, pulling down, pulling out, kicking, lifting, etc. This allows for a comprehensive evaluation of the performance of loco-manipulators across multiple dimensions.

B. Task Description

- **Press button.** The robot needs to press a hemispherical button with a diameter of 10 cm.
- **Pull handle.** The robot is required to pull down a door handle. Due to the height of the robot’s torso and the limitations in joint angles, this task is challenging for loco-manipulation using the feet, requiring the robot to maintain high precision in manipulation at the limits of its operating space.
- **Push door.** The robot needs to push open a door. This assesses the ability of the robot to walk over a large distance on just three legs while manipulating the door with the foot.
- **Lift the Basket.** The robot needs to lift a basket with its foot and walk a distance. This evaluates the robot’s precision control over the 6-DoF of its end-effector and its ability to move over a large distance.
- **Open dishwasher.** The robot needs to open the door of a dishwasher with its foot. This task is very easy for a robotic arm with a gripper but very difficult for a foot with point contact.
- **Close the Dishwasher.** The robot needs to close the door of dishwasher with its foot.
- **Pull objects.** The robot is required to pull objects off a table with its foot.
- **Twist the Valve.** The robot has to twist a valve with a diameter of about 40 cm, where the axis is 60 cm off the ground. This tests the robot’s ability to manipulate over a large range at the limits of its workspace.
- **Shoot ball.** The robot needs to run a distance and shoot a soccer ball into a goal with its foot.

TABLE III: Performance of our method against baseline Hierarchical Reinforcement Learning (HRL) across 9 tasks. Our method achieves significantly better success rates on all tasks. The success rate for each task is calculated as the average across three seeds.

Success rate (%)	Expert	HRL	Ours
Press Button	93.67±1.53	21.67±3.51	92.33±8.96
Pull Handle	86.33±3.21	0.00±0.00	82.33±3.21
Push Door	88.00±7.76	15.67±3.51	85.33±5.03
Lift Basket	67.00±9.54	11.00±4.36	59.33±12.01
Open Dishwasher	20.67±4.62	1.33±1.15	5.67±5.51
Close Dishwasher	92.33±5.51	4.67±1.53	50.33±8.14
Pull Objects	42.00±1.73	8.00±2.65	12.67±10.79
Twist Valve	70.33±2.31	10.00±4.58	52.33±10.21
Shoot Football	37.67±18.18	3.67±2.08	26.00±2.00

V. EXPERIMENTS

In this section, we design experiments to test the effectiveness of the proposed method and compare the performance of task completion against different baselines. Subsequently, we validate the learned loco-manipulation skills on an Unitree Aliengo robot and demonstrate sim-to-real transfer capabilities. Finally, we analyze the performance of the proposed method across both the planner and the control policy.

A. Performance Comparison

Our approach utilized 3 billion timesteps of robot state data to train the low-level controller and 20k timesteps of visual data to train the high-level planner. The final success rates for 9 tasks are presented in Table III.

- **End-to-End BC (BC).** We employ DP3 [33] to train an end-to-end BC policy for the same tasks. During the collection of expert demonstration data, we also gather the outputs of the controller, which means that trajectory $\xi_i = \{(^w P_i, s_i, ^w p_i, a_i)\}$, where a_i is the robot desired joint position.
- **Visual RL as Planner (HRL).** We utilize one of the best visual Reinforcement Learning (RL) algorithms, DrQ-v2 [43], as the planner within our framework. The low-level controller employed is identical to the one integrated within our framework.
- **End-to-End Visual RL (VRL).** We utilize DrQ-v2 [43] to train an end-to-end policy for solving the tasks.

Considering the difficulties that most baselines encounter in accomplishing our challenging tasks, we chose to closely examine the button pressing and dishwasher closing tasks. In these tasks, our method achieves the highest success rate and approximately a 50% success rate, respectively. Concurrently, we evaluate the performance of both our method and the HRL baseline across all tasks, which is the best performing baseline on the button pressing and dishwasher closing tasks.

Success rate. In Table. III, we compared the performance of our method against HRL across multiple tasks. Our method surpasses HRL in all tasks. HRL requires meticulous adjustment of rewards for each task; without this, it struggles to learn how to tackle these challenging tasks.

Data efficiency. As shown in Table. IV, we compare the performance of our method against BC, VRL and HRL for the button pressing and dishwasher closing tasks, testing the

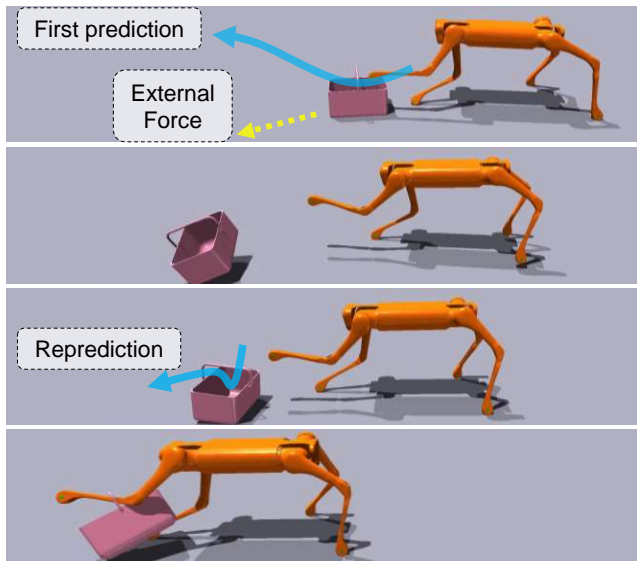


Fig. 5: The performance of our proposed method in the task of lifting a basket when encountering unexpected situations. At the start, the robot estimated the trajectory based on initial pose of basket. When the basket was pushed by external force, the robot quickly updated its trajectory prediction to account for the basket’s new pose, enabling it to lift the displaced basket successfully.

performance of each method with an increasing number of visual data. Our method only requires 20k timesteps of visual data to successfully complete the task, while BC and HRL are also difficult to complete the task with a much larger amount of visual data, VRL is completely unable to achieve our task. This data efficiency is achieved by our carefully designed framework, which trains low-level controllers with easily accessible robot state data.

B. Robust Manipulation in Unexpected Situations

Our expert trajectories are generated in simulation with fixed trajectory parameters, a method that is notably efficient and rapid. However, unlike human video data and teleoperation, this approach does not allow for the collection of data in unexpected situations. Surprisingly, even under these conditions, our method still demonstrated robustness to unforeseen circumstances.

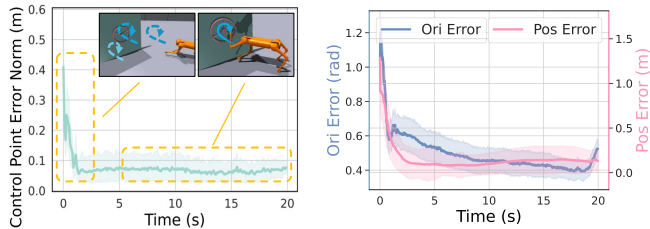
Taking the task of lifting a basket as an example, we analyze the robot’s manipulation process in the face of unexpected events. As shown in Fig. 5, at the beginning of the episode, the robot predicted the trajectory parameters according to the basket initial pose. During the approach to the basket, we applied random force to the basket, causing it to roll approximately 1.5 meters away. However, the robot quickly repredicted the trajectory parameters based on the new point clouds of the object, allowing the robot to successfully lift the displaced basket.

C. Sim2real Transfer

We directly implement the trained controller and planner on a real-world Aliengo quadruped robot without the need for further fine-tuning, demonstrating the robustness of our

TABLE IV: Comparison of performance on the button pressing and dishwasher closing task. Our method versus end-to-end Behavior Cloning (BC), Hierarchical Reinforcement Learning (HRL) and end-to-end Visual Reinforcement Learning (VRL), each trained with varying amounts of visual data, with the amounts of data labeled after each baseline. M denotes million; K denotes one thousand. Our approach not only achieved a significantly higher success rate but also required considerably less expensive visual data. The success rate is calculated as the average across three seeds.

Success rate (%)	Ours-2K	Ours-20K	BC-20K	BC-500K	HRL-1M	VRL-5M
Press Button	42.33±17.16	92.33±8.96	0.00±0.00	0.67±0.58	21.67±3.51	0.00±0.00
Close Dishwasher	33.67±7.77	50.33±8.14	0.00±0.00	3.33±2.08	4.67±1.53	0.00±0.00



(a) Valve Twisting Task Performance. The green line displays the differences between the predicted parameters and the expert parameters during the manipulation process, as calculated from data collected during 100 tests. (b) End-effector Trajectory Tracking Performance of the Control Policy. Data on the position and orientation of the end-effector were collected and analyzed during the process of collecting expert demonstrations 10 times for each of the 9 tasks.

Fig. 6: The predictive and tracking performance of planners and control policy in simulation.

approach. As shown in Fig. 2, the pose of the robot is determined using AprilTag for the coordinate system transformation of trajectory parameters and point clouds. Following this, we applied preprocessing to the point cloud to ensure alignment with the simulation. In Fig. 1, we illustrate the robot performing tasks like lifting a basket with its forelimbs and pushing the door during locomotion. For demonstrations of all the skills developed through our method, please refer to the accompanying video.

D. Planner Performance

The trained planner is capable of predicting the trajectory parameters based on the point clouds and transmitting them to the controller, enabling the robot to complete challenging loco-manipulation tasks, as shown in Fig. 4.

We conduct an in-depth test of the planner’s performance, using the valve twisting task as an example for analysis. As shown in Fig. 6a, at the beginning of the episode, the predicted parameters are constantly changing, leading to a rather chaotic predicted trajectory. As the manipulation progresses, the predicted trajectory tends to converge.

E. Control Policy Performance

Robust tracking against disturbances. We test the tracking performance of the control policy. In the tests, the robot is required to follow the expert trajectories for 9 different tasks and record the position and orientation errors of the end-effector in comparison to the desired value.

As shown in Fig. 6b, both position and orientation errors peaked when the manipulation began, then position error converged to the lowest value after approximately 3 seconds. In the latter half of the episode, due to the contact with objects, position errors slightly increased but still remained

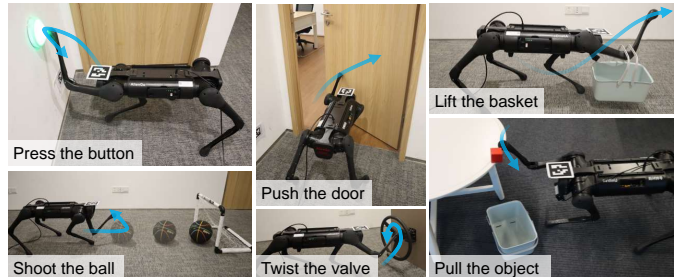


Fig. 7: Teleoperation in real world with control policy. We accomplish the above tasks in the real world by the low-level controller with specific trajectory parameters via a joystick.

at a low level. Note that precise tracking of both the position and orientation of the EE is typically unsolvable, given the system is highly underactuated. Our system opts to prioritize position tracking, resulting in the reduction of the minimum orientation error to 0.4 radians.

Teleoperation with control policy. Besides autonomous manipulation, we can *also* collect data via teleoperation. As shown in Fig. 7, using the trained low-level controller, we executed a series of real-world experiments on the robot, guided by specific trajectory parameters. These experiments show that our low-level controller can effectively perform a variety of daily tasks in real-world settings, tasks that previously necessitated a robotic arm.

VI. CONCLUSION AND FUTURE WORKS

In this work, we decompose the loco-manipulation process of legged robots into a low-level controller based on RL and an high-level planner based on BC.

a) limitation: While we have demonstrated the effectiveness of our method in both simulation and real-world scenarios, there are areas for improvement. (i) The inference speed limitations of diffusion-based BC hinder task performance, making it challenging for robots to handle dynamic environments. (ii) While our approach to gathering expert demonstrations is significantly efficient, it does not robustly handle unexpected situations. Additionally, the diversity of the expert demonstrations collected is limited.

b) Future works: This study represents a novel effort to master whole-body loco-manipulation, possessing boundless potential. (i) It showcases remarkable scalability, enabling rapid collection of expert data for either scale up or data-mixture with real-world data. (ii) To further bridge the simulation-to-reality gap, exploring the use of teleoperation for real-world data collection is an option. Given our method’s high data efficiency, this method could substantially lessen the required effort.

REFERENCES

- [1] X. Cheng, A. Kumar, and D. Pathak, "Legs as manipulator: Pushing quadrupedal agility beyond locomotion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [2] X. Huang, Z. Li, Y. Xiang, Y. Ni, Y. Chi, Y. Li, L. Yang, X. B. Peng, and K. Sreenath, "Creating a dynamic quadrupedal robotic goalkeeper with reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2715–2722.
- [3] Y. Ji, Z. Li, Y. Sun, X. B. Peng, S. Levine, G. Berseth, and K. Sreenath, "Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 1479–1486.
- [4] Y. Ji, G. B. Margolis, and P. Agrawal, "Dribblebot: Dynamic legged manipulation in the wild," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5155–5162.
- [5] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, 2023.
- [6] H. Xiong, R. Mendonca, K. Shaw, and D. Pathak, "Adaptive mobile manipulation for articulated objects in the open world," *arXiv preprint arXiv:2401.14403*, 2024.
- [7] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *arXiv*, 2024.
- [8] N. M. M. Shafiqullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, "On bringing robots home," *arXiv preprint arXiv:2311.16098*, 2023.
- [9] B. Wu, R. Martín-Martín, and L. Fei-Fei, "M-ember: Tackling long-horizon mobile manipulation via factorized domain transfer," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 690–11 697.
- [10] N. Yokoyama, A. Clegg, J. Truong, E. Undersander, T.-Y. Yang, S. Arnaud, S. Ha, D. Batra, and A. Rai, "Asc: Adaptive skill coordination for robotic mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 779–786, 2024.
- [11] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu *et al.*, "Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments," in *Conference on Robot Learning*. PMLR, 2022, pp. 477–490.
- [12] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model predictive control," 2022.
- [13] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "Dtc: Deep tracking control," *Science Robotics*, vol. 9, no. 86, p. eadh5401, 2024.
- [14] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1154–1171, 2021.
- [15] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition @ CoRL2023*, 2023.
- [16] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," in *Conference on Robot Learning (CoRL)*, 2023.
- [17] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," 2023.
- [18] T. Miki, J. Lee, L. Wellhausen, and M. Hutter, "Learning to walk in confined spaces using 3d representation," 2024.
- [19] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Conference on Robot Learning*. PMLR, 2023, pp. 403–415.
- [20] 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.
- [21] K. LEI, Z. He, C. Lu, K. Hu, Y. Gao, and H. Xu, "Uni-o4: Unifying online and offline deep reinforcement learning with multi-step on-policy optimization," in *The Twelfth International Conference on Learning Representations*, 2024.
- [22] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Legged robots that keep on learning: Fine-tuning locomotion policies in the real world," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1593–1599.
- [23] R. Yang, Z. Chen, J. Ma, C. Zheng, Y. Chen, Q. Nguyen, and X. Wang, "Generalized animal imitator: Agile locomotion with versatile motion prior," in *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition @ CoRL2023*, 2023.
- [24] J. Wu, G. Xin, C. Qi, and Y. Xue, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robotics and Automation Letters*, 2023.
- [25] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter, "Advanced skills through multiple adversarial motion priors in reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5120–5126.
- [26] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics (ToG)*, vol. 40, no. 4, pp. 1–20, 2021.
- [27] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: Learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning (CoRL)*, 2022.
- [28] 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.
- [29] S. Zimmermann, R. Poranne, and S. Coros, "Go fetch! - dynamic grasps using boston dynamics spot with external robotic arm," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4488–4494.
- [30] B. Forrai, T. Miki, D. Gehrig, M. Hutter, and D. Scaramuzza, "Event-based agile object catching with a quadrupedal robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 177–12 183.
- [31] P. Arm, M. Mittal, H. Kolvenbach, and M. Hutter, "Pedipulate: Enabling manipulation skills using a quadruped robot's leg," 2024.
- [32] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Proc. Int. Conf. Robot. Automat.*, 2011.
- [33] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy," *arXiv preprint arXiv:2403.03954*, 2024.
- [34] F. Abdolhosseini, H. Y. Ling, Z. Xie, X. B. Peng, and M. van de Panne, "On learning symmetric locomotion," in *Motion, Interaction and Games*, ser. MIG '19. New York, NY, USA: Association for Computing Machinery, 2019.
- [35] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2497–2503.
- [36] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance GPU based physics simulation for robot learning," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- [37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [38] 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.
- [39] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2019.
- [40] Y. Chen, T. Wu, S. Wang, X. Feng, J. Jiang, Z. Lu, S. McAleer, H. Dong, S.-C. Zhu, and Y. Yang, "Towards human-level bimanual dexterous manipulation with reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5150–5163, 2022.
- [41] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín, "robosuite: A modular simulation framework and benchmark for robot learning," *CoRR*, vol. abs/2009.12293, 2020.
- [42] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.
- [43] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, "Mastering visual continuous control: Improved data-augmented reinforcement learning," in *International Conference on Learning Representations*, 2021.