Learning a Unified Policy for Position and Force Control in Legged Loco-Manipulation

Peiyuan Zhi[1,2,*], **Peiyang Li**[1,3,*], **Jianqin Yin**[3], **Baoxiong Jia**[1,2,†], **Siyuan Huang**[1,2,†]
[1] State Key Laboratory of General Artificial Intelligence, BIGAI
[2] Joint Laboratory of Embodied AI and Humanoid Robots, BIGAI & UniTree Robotics
[3] Beijing University of Posts and Telecommunications
https://unified-force.github.io/
[*] Equal contribution. [†] Corresponding authors.

Figure 1: **We present a unified force-position policy for legged robots** that enables diverse loco-manipulation behaviors, including position tracking, force application, and compliant interactions (top). When used for imitation learning data collection, the policy's learned internal force estimator provides force-aware demonstrations, improving model performance in contact-rich tasks without external force sensors (middle). Results on quadruped and humanoid robots demonstrate the policy's versatility and robustness (bottom).

Abstract: Robotic loco-manipulation often involves contact-rich interactions with the environment, requiring the joint modeling of contact force and robot position. However, recent visuomotor policies often focus solely on learning position or force control, overlooking their co-learning. We propose the first unified policy for legged robots that jointly models force and position control learned without relying on force sensors. By simulating diverse combinations of position and force commands alongside external disturbance forces, we use reinforcement learning to learn a policy that estimates forces from historical robot states and compensates for them through position and velocity adjustments. This policy enables a wide range of manipulation behaviors under varying force and position inputs, including position tracking, force application, force tracking, and compliant interactions. Moreover, we demonstrate that the learned policy enhances trajectory-based imitation learning pipelines by incorporating essential contact information through its force estimation module, achieving approximately $\sim$39.5% higher success rates in four challenging contact-rich manipulation tasks over position-control policies. Experiments on both a quadrupedal manipulator and a humanoid robot validate the versatility and robustness of the proposed policy in diverse scenarios.

Keywords: Unified Force and Position Control, Force-aware Imitation Learning

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

1 Introduction

Legged robots have recently advanced in locomotion and manipulation [1, 2, 3, 4], enabling them to traverse complex terrains (*e.g.*, stairs) and extend their workspace through adaptive body posture, revitalizing interest in loco-manipulation [5, 6, 7, 8]. However, controlling legged manipulators is challenging due to their complex kinematic structures. This difficulty is further exacerbated in contact-rich manipulation tasks, where accurate modeling of contact forces is essential for desired control behaviors (*e.g.*, compliance), yet is hindered by the absence of force-sensing hardware. These challenges underscore the need for robust, adaptable policies to support effective robot-environment and human-robot interactions.

To tackle the control challenge of legged manipulators, reinforcement learning (RL) algorithms have emerged as effective alternatives to traditional control methods, offering robust and generalizable policies trained through domain randomization [2, 6, 7, 8, 9, 10]. These policies integrate locomotion and manipulation in complex tasks but primarily depend on precise position control, limiting their applicability in contact-rich scenarios. This reliance has also driven the rise of position-based robot imitation learning [11, 12, 13, 14, 15], with large datasets [13, 16, 17, 18] focused solely on robot trajectories, omitting crucial contact information due to the lack of force sensing. As shown in Section 4.2, such trajectory-only data is insufficient for training effective policies, even for basic contact-rich tasks (*e.g.*, wiping a blackboard). This underscores the limitation of position control and emphasizes the necessity of integrating force sensing and modeling into learning-based policies for more effective task execution.

In light of the aforementioned challenges and observations, we propose **the first unified policy for legged robots that seamlessly integrates force and position control without the need for force sensors**. Unlike previous methods [9] that handle force and position control independently, we train a single control policy using RL in Isaac Gym [19] by simulating diverse combinations of position and force commands alongside external disturbance forces. The policy leverages a force estimator to predict external forces based on the robot's historical states and offsets to target positions, enabling adaptive adjustments to the robot's position and velocity. The learned policy supports versatile manipulation behaviors, including position tracking, force application, force tracking, and compliant responses to varied force and position inputs. We also verify the generalizability of this learning framework to different robot embodiments through an extensive spectrum of 7 experiments on both the Unitree B2-Z1 quadrupedal manipulator platform and the Unitree G1 humanoid robot.

Additionally, we highlight the capabilities of the learned policy in facilitating imitation learning with contact force information. Specifically, we develop a force-aware data collection pipeline that utilizes our learned policy as the base teleoperation policy, simultaneously passing position and force commands to the robot while collecting contact-rich manipulation data via the embedded contact force estimator. We validate the effectiveness of this data by integrating the estimated force into a position-based imitation learning policy, leading to **a significant improvement ($\sim$39.5%) in success rates over the vanilla position-based methods** across three challenging contact-rich tasks. These experimental results underscore the potential of our learned policy as a general framework for curating contact-rich robot interaction data, particularly in the absence of explicit force sensors.

Overall, our contributions can be summarized as follows:

1. We propose the first model for learning unified force and position control in legged loco-manipulation, enabling diverse control behaviors such as position tracking, force control, and compliance with a single policy.
2. Through 7 experiments on a quadrupedal manipulator and a humanoid robot, we demonstrate the effectiveness and robustness of our learned policy across diverse and challenging task scenarios.
3. We develop a force-aware robot imitation learning data collection pipeline using our learned force estimator, improving position-based imitation learning baselines by $\sim$39.5% on three challenging contact-rich manipulation tasks, highlighting our policy's promise as a general and efficient framework for contact-rich task demonstration curation.

2 Related Works

Whole-body Control Whole body control (WBC) has been widely adopted to enhance robotic capabilities in mobile manipulation, particularly within classical control frameworks [5, 20, 21]. More recently, RL with parallel simulators [19, 22] has become the mainstream approach for addressing complex control challenges in legged robots. Several learning-based methods [6, 7, 23, 24, 25] have improved the robustness of WBC, while others have extended its application to force-intensive tasks [26, 27, 28, 29, 30]. For instance, [27] coordinates joint movements to apply sufficient force during pushing, and ALMA[29] combines WBC with force control to achieve precise end-effector actuation. [30] integrates Cartesian impedance control into a QP formulation, enabling compliant loco-manipulation through a double mass-damper-spring model. These works collectively highlight WBC's effectiveness in unifying force and position control.

Hybrid Force and Position Control In contact-rich manipulation tasks, relying solely on end-effector trajectory control is often insufficient due to the inherent coupling between force and position. Early work [31, 32, 33, 34], including the introduction of impedance control [34], laid the foundation for hybrid force-position strategies. Recent studies [1, 9, 35, 36, 37] have advanced compliance control, with some leveraging force sensors and others estimating force indirectly via internal signals or reinforcement learning. Inspired by these trends, our work eliminates the need for force sensors by using reinforcement learning to train a quadruped robot to simultaneously control force and position. This enables flexible switching between force following, impedance control, and hybrid modes through different command configurations.

Imitation Learning for Mobile Manipulation Imitation learning [38, 39, 40, 41, 42] has recently become a prominent approach for training robots to perform various tasks. Behavior cloning (BC) [43, 44] is a straightforward method of imitation learning that learns policies by supervising observation-action pairs from expert demonstrations. Studies leveraging image data and proprioceptive sensing [8, 45, 46, 47] to generate robot control commands have shown remarkable success in mobile manipulation tasks. Furthermore, recent research [48, 49] has begun incorporating tactile sensing to enhance the sensory capabilities of robots. Similarly, our work utilizes force inputs without relying on force sensors, demonstrating that force information is critical in enabling robots to complete challenging tasks effectively.

3 Method

3.1 A Unified Formulation for Force and Position Control

We begin by introducing the general problem formulation of our approach. As shown in the upper part in Fig. 2(c), given the position command relative to the robot body frame and force command, $\mathbf{x}^{\text{cmd}}$ and $\boldsymbol{F}^{\text{cmd}}$, our goal is to learn a RL policy that ensures the robot's behavior adheres to these commands under net force $\boldsymbol{F}$. To achieve this goal, we adopt the impedance control formulation:

$$\boldsymbol{F} = K(\mathbf{x} - \mathbf{x}^{\text{des}}) + D(\dot{\mathbf{x}} - \dot{\mathbf{x}}^{\text{des}}) + M(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}^{\text{des}}), \tag{1}$$

where $\mathbf{x}$ denotes the actual position of the robot. $\mathbf{x}^{\text{des}}$, $\dot{\mathbf{x}}^{\text{des}}$, and $\ddot{\mathbf{x}}^{\text{des}}$ denotes the desired goal position, velocity, and acceleration of the robot. The parameters K, D, and M correspond to the stiffness and damping coefficients, and equivalent mass (inertia), respectively.

End-effector Modeling As the end-effector typically moves slowly during manipulation tasks, we can make the following simplification over Eq. (1): $\boldsymbol{F} = K(\mathbf{x} - \mathbf{x}^{\text{des}})$. The net force $\boldsymbol{F}$ primarily consists of three components: the active force $\boldsymbol{F}^{\text{cmd}}$, the passive reaction force $\boldsymbol{F}^{\text{react}}$ which arises from applying $\boldsymbol{F}^{\text{cmd}}$ to the environment, and additional external disturbances $\boldsymbol{F}^{\text{ext}}$. Therefore, the desired target position $\mathbf{x}^{\text{target}}$ of the end-effector is given by:

$$\mathbf{x}^{\text{target}} = \mathbf{x}^{\text{cmd}} + \frac{\boldsymbol{F}^{\text{ext}} + (\boldsymbol{F}^{\text{cmd}} - \boldsymbol{F}^{\text{react}})}{K}, \tag{2}$$

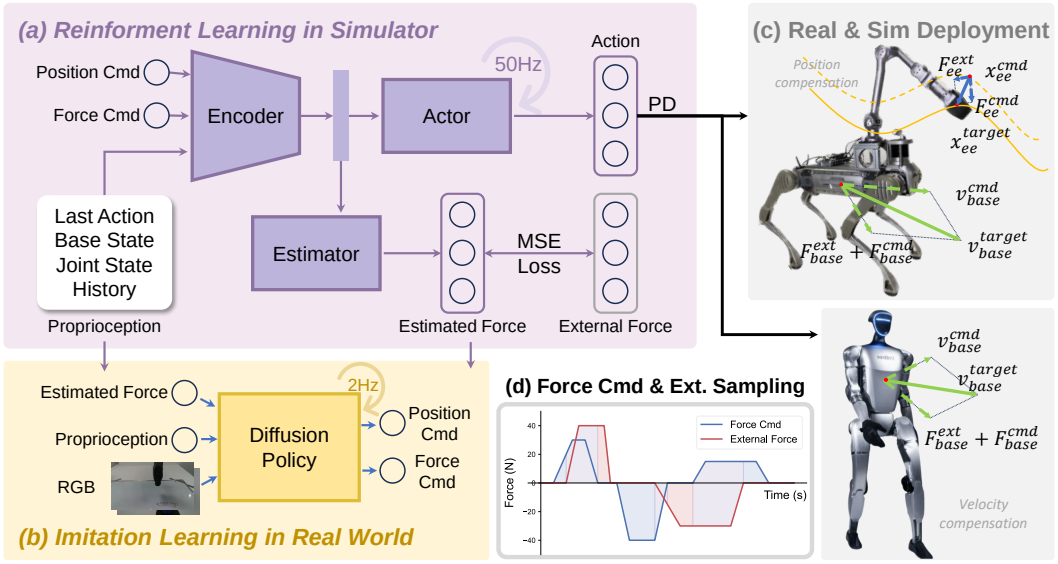

Figure 2: **Method Overview**. (a) Architecture of the unified position-force policy trained via reinforcement learning to track position and force commands under external disturbances. (b) Force-aware imitation learning enabled by demonstrations collected using our learned policy, without requiring force sensors. (c) Illustration of position and velocity compensation for force interactions modeled at both the end-effector and the robot base. (d) Visualization of sampled force commands and disturbances used to simulate diverse contact scenarios during policy training.

where the environment reaction force prevents the end-effector from reaching the commanded position $\mathbf{x}^{\text{cmd}}$. Under the formulation of Eq. (2), several manipulation behaviors can be derived by appropriately specifying the position command $\mathbf{x}^{\text{cmd}}$ and force command $\boldsymbol{F}^{\text{cmd}}$, including *Position Control*, *Force Control*, *Impedance Control* and *Hybrid Position and Force Control*. We present a detailed formulation of these control behaviors in Section A. In complex scenarios involving simultaneous position commands, force commands, and external disturbances, the system adheres to Eq. (2), integrating these basic control modes.

Multi-contact Modeling For other robot body parts beyond the end-effector, the formulation in Eq. (2) can be extended accordingly. Taking the robot base as an example, we typically care not about its joint state, but rather its velocity or global position. In such cases, Eq. (2) can be simplified by assuming the robot is controlled through base velocity commands. Specifically, given the velocity and force commands $\mathbf{v}_{\text{base}}^{\text{cmd}} = \dot{\mathbf{x}}_{\text{base}}^{\text{cmd}}$ and $\boldsymbol{F}_{\text{base}}^{\text{cmd}}$, along with external disturbances $\boldsymbol{F}_{\text{base}}^{\text{ext}}$, we can derive from Eq. (1):

$$\boldsymbol{F}_{\text{base}} = D(\dot{\mathbf{x}}_{\text{base}} - \dot{\mathbf{x}}_{\text{base}}^{\text{des}}) = D(\mathbf{v}_{\text{base}} - \mathbf{v}_{\text{base}}^{\text{des}}), \tag{3}$$

where we omit the position term because global position of the base is not available. Here, $\boldsymbol{F}_{\text{base}} = \boldsymbol{F}_{\text{base}}^{\text{ext}} + (\boldsymbol{F}_{\text{base}}^{\text{cmd}} - \boldsymbol{F}_{\text{base}}^{\text{react}})$ is the net force and transform Eq. (2) into:

$$\mathbf{v}_{\text{base}}^{\text{target}} = \mathbf{v}_{\text{base}}^{\text{cmd}} + \frac{\boldsymbol{F}_{\text{base}}^{\text{ext}} + (\boldsymbol{F}_{\text{base}}^{\text{cmd}} - \boldsymbol{F}_{\text{base}}^{\text{react}})}{D}. \tag{4}$$

After this transformation, we can implement similar basic control modes using derivations from Eq. (4). Furthermore, this formulation can be extended to scenarios where external disturbances and force commands on body parts are transformed to end-effectors by converting the net force on the robot base $\boldsymbol{F}_{\text{base}}$ into an external force to the end-effector $\boldsymbol{F}_{\text{base2ee}}$, and *vice versa*. However, due to the learning complexity of such methods, this work focuses on treating the end-effector and robot base independently, leaving the integrated derivation as important future work.

As a summary, we build our policy learning with reward provided following Eqs. (2) and (4) which models the behavior of the end-effector and the base of the legged robot considering both active and passive forces. We provide the detailed training settings and model in Section 3.2.

3.2 Learning a Unified Force-Position Control Policy

We detail the learning of the proposed unified force-position control policy by first defining the space of observations, commands, and actions. Specifically, we define the robot's observation $\mathbf{o}_t$ with the robot's base orientation $\mathbf{g}_t^{\text{base}}$, angular velocity ω_t^{base}, joint position $\mathbf{q}_t$, joint velocities $\dot{\mathbf{q}}_t$, previous action $\mathbf{a}_{t-1}$, command $\mathbf{c}_t^{\text{cmd}}$, and the feet clock timings θ_t^{feet}:

$$\mathbf{o}_t = \left[\mathbf{g}_t^{\text{base}}, \omega_t^{\text{base}}, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}, \mathbf{c}_t^{\text{cmd}}, \theta_t^{\text{feet}}\right] \tag{5}$$

where the input command $\mathbf{c}_t^{\text{cmd}} = [\mathbf{v}_{\text{base}}^{\text{cmd}}, \mathbf{x}_{\text{ee}}^{\text{cmd}}, \boldsymbol{F}_{\text{ee}}^{\text{cmd}}, \boldsymbol{F}_{\text{base}}^{\text{cmd}}]$ covers the base velocity, end-effector position, end-effector force, and base force commands. For quadrupedal robots, we consider all four command types. For humanoid robots, only the locomotion command $\mathbf{v}_{\text{base}}^{\text{cmd}}$ and base force command $\boldsymbol{F}_{\text{base}}^{\text{cmd}}$ are considered as there is no gripper available on the robot for manipulation tasks. The output action $\mathbf{a}_t$ is a residual added to a predefined default pose and $\mathbf{q}_t^{\text{target}}$ are the joint position targets for the PD controller, calculated as $\mathbf{q}_t^{\text{target}} = \sigma_a \mathbf{a}_t + \mathbf{q}^{\text{default}}$, where σ_a scales the policy output and $\mathbf{q}^{\text{default}}$ represents a standard pose.

Policy Design We provide an overview of our policy model in Fig. 2(a). Our policy model comprises three modules: the observation encoder, the state estimator, and the actor. The encoder processes the observation history $\mathbf{o}_{[t-H,\ldots,t-1,t]}$ $(H = 32)$ and outputs a latent feature, which is then sent to the state estimator and the actor. The state estimator then predicts the robot's state, including the external force $\boldsymbol{F} = \boldsymbol{F}^{\text{ext}} + \boldsymbol{F}^{\text{react}}$, the end-effector position, and the base velocity. This estimated force could then be translated to command signals in certain desired control behaviors.

Force Simulation To simulate diverse scenarios for learning the unified force-position policy, we randomly sample position, velocity, and force commands, along with external net forces as required in Eqs. (2) and (4). The ranges of input commands and external forces are detailed in Section C.1. Notably, the reaction force $\boldsymbol{F}^{\text{react}}$ is not modeled explicitly but is incorporated into the net external force $\boldsymbol{F} = \boldsymbol{F}^{\text{ext}} + \boldsymbol{F}^{\text{react}}$. The sampling range of $\mathbf{x}^{\text{cmd}}$ slightly exceeds the arm's original workspace without whole-body movement, while ensuring that the resulting $\mathbf{x}_{\text{ee}}^{\text{target}}$ remains within the operational limits when whole-body motion is allowed. During training, as illustrated in Fig. 2(d), sampled forces are linearly ramped up to target values, held constant for a fixed interval, and then reduced back to zero according to a pre-defined schedule. After a brief zero-force period, new forces are and the cycle repeats. This sampling strategy exposes the policy to a variety of control conditions, echoing the different desired control behavior discussed in Section 3.1 and enabling a single policy to adapt to varying control task demands.

Policy Learning We adopt a two-stage training procedure: first focusing on whole-body reaching and locomotion, then introducing random force commands and external disturbances. This staged approach empirically yields more stable training than a single-stage setup, as further analyzed in Section C. Policy learning is supervised by rewarding accurate tracking of the target end-effector position $\mathbf{x}_{\text{ee}}^{\text{target}}$ and base velocity $\mathbf{v}_{\text{base}}^{\text{target}}$ under varying input and disturbance combinations. Additionally, an MSE loss is used to improve the accuracy of the state estimator for both robot state and external force. Full reward specifications are provided in Table A.1.

3.3 Force-aware Imitation Learning

Recognizing the importance of force information in real-world manipulation tasks and its absence in most existing datasets, we leverage our learned force-position policy to collect force-aware data for imitation learning. Concretely, we teleoperate the robot to record joint states, base states, control commands, estimated end-effector contact forces, and RGB images from cameras mounted on both the end-effector and the robot base. This data is used to train a diffusion-based force-aware imitation learning policy that takes as input the robot states, estimated forces, and image observations, and predicts both force and end-effector position commands as inputs to our low-level force-position policy. Unlike prior works relying solely on visual inputs, our force estimator supplements the policy with contact information, enabling more accurate object interaction and force application. We

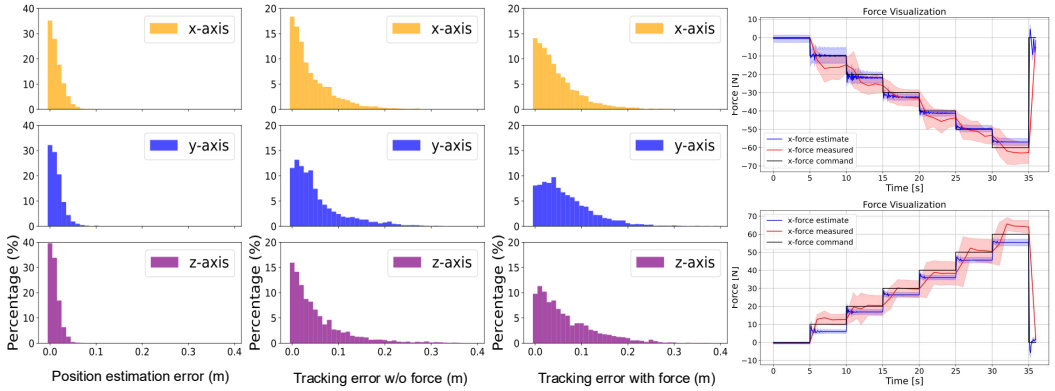

Figure 3: **Force and position control evaluation.** (a)–(c) Evaluation of force and position control tracking errors in simulation environments. (d) Real-world evaluation of force control, shaded areas indicate variance measured across 5 different end-effector positions.

validate the impact of the collected data and the effectiveness of our approach in Section 4.2. Details of the teleoperation pipeline and training procedures are provided in Section B and Section D.

4 Experiment

4.1 Force and Position Command Tracking

Position Tracking To evaluate performance in simulation, we conduct 6000-step rollouts with randomly generated end-effector trajectories with position commands only, covering the entire training workspace. We report the average position tracking and estimation errors across these trials. As shown in Fig. 3(b), the end-effector tracking error remains mostly within 0.1m when in the absence of external forces and force commands. Slightly higher errors are observed along the Y-axis, likely due to fewer available degrees of freedom in that direction, which limits precision. We also assess the accuracy of the state estimator by comparing the estimated end-effector positions with ground-truth simulation values. Across all axes, the estimation error remains within 0.05m, as shown in Fig. 3(a).

Force Control We evaluate the ability of our proposed policy to estimate and act with forces under two settings. First, we assess position tracking performance when the policy receives force commands that match the applied external forces, serving as an indirect evaluation for evaluating unified force-position control. As shown in Fig. 3(c), compared to the experiment without external forces, tracking error increases slightly compared to the no-force setting but remains mostly within 0.1m, demonstrating effective force-aware behavior. Second, we conduct a direct force control evaluation on real robots by applying force commands ranging from 0 N to 60 N and measuring end-effector forces using a dynamometer. Measurements at five different end-effector positions yield average errors within 10 N, as shown in Fig. 3(d). Force estimation across six discrete levels shows errors between 5–10 N. Due to hardware limitations, evaluations along the Y- and Z-axes are capped at 40 N. Despite minor sim-to-real discrepancies, particularly along the Y-axis, the estimator remains sufficiently accurate for the targeted manipulation tasks. More analyses are provided in Section E.

4.2 Force-aware Imitation Learning

Task Settings We evaluate our method on four real-world tasks that require hybrid force-position control and force sensing: `wipe-blackboard`, `open-cabinet`, `close-cabinet`, and `open-drawer-occlusion`. In the `wipe-blackboard` task, the robot must maintain continuous contact with the surface while moving laterally to erase ink marks. We collect 50 trajectories and trained the force-aware diffusion policy for $30k$ steps. In `open-cabinet` and `close-cabinet`, the robot interacts with a push-to-open cabinet, while in `open-drawer-occlusion`, it opens a drawer that gradually becomes occluded in visual observations (see the setup in Fig. A.8). For each of

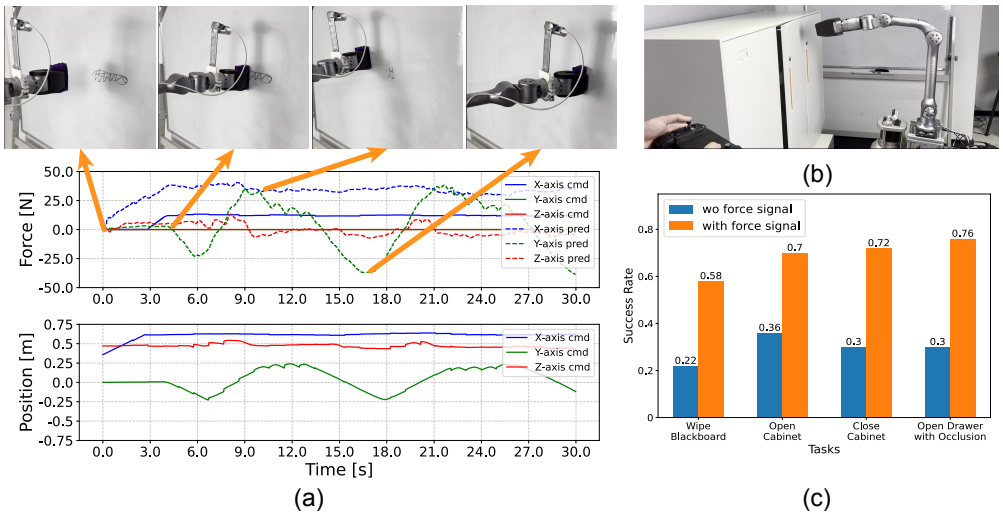

Figure 4: **Force-aware imitation learning.** (a) Time-series outputs of position and force commands to the trained force-aware imitation policy in the `wipe-blackboard` task. *cmd* denotes the output of the imitation learning policy, while *pred* indicates the external force estimated by the low-level policy. (b) A visualization of the data collection process. (c) The performance comparison between our policy and a baseline vision-only policy over 50 trials across four tasks.

the three open/close tasks, we collect 30 episodes per task and train the policy for $20k$ steps. As a baseline, we deploy the trained low-level policy but exclude the force estimator and force command signals during teleoperation data collection. Each task is performed 50 times with each trial constrained to a maximum of 1000 steps ($\sim$20 seconds) for successful completion. For completeness, we provide additional details about the task definitions and experimental settings in Section D.

Results and Analyses In Fig. 4 , we compare our method to the baseline in the four real-world tasks. Our approach achieves $\sim$39.5% higher success rates than the baseline. In `wipe-blackboard`, the position-only policy fails to maintain stable contact, often resulting in insufficient wiping or excessive force that risks surface damage. In contrast, our force-aware policy ensures consistent contact pressure, while the low-level policy improves compliance and reduces mechanical stress. For `open-` and `close-cabinet`, the primary challenge lies in the push-to-open mechanism's narrow 3mm stroke, which is difficult to detect using vision alone. Our force estimator accurately senses the required contact force, enabling reliable activation. In `open-drawer-occlusion`, the baseline policy, relying solely on visual cues, suffers a sharp success rate drop to 0.3 due to unobservable contact. Our method leverages force sensing to detect contact under occlusion, increasing the success rate to 0.76 and underscoring the importance of force estimation in vision-compromised scenarios. We provide all quantitative comparisons and additional details in Section D.

4.3 Basic Manipulation Policies

Force Control Force control directly applies a commanded force by moving the end-effector in the force direction until the applied and commanded forces match. Our unified strategy requires no additional training; following Eq. (A.7), where the sum of the estimated external force and the force command determines displacement compensation until equilibrium is achieved. As demonstrated in Fig. 5(a) and the supplementary video, when a $2.5kg$ dumbbell is attached to the end-effector, the robot achieves balance with a $25N$ upward force command. Without this command, however, the end-effector drops due to the dumbbell's gravity.

Force tracking Force tracking is a special case of force control where the end-effector tracks a zero-force command. When external forces are applied, the end-effector moves in the force direction. Once the force is removed, it stays in the displaced position instead of returning to the original target. This behavior is implemented using our unified policy, following Eq. (A.9). As shown in

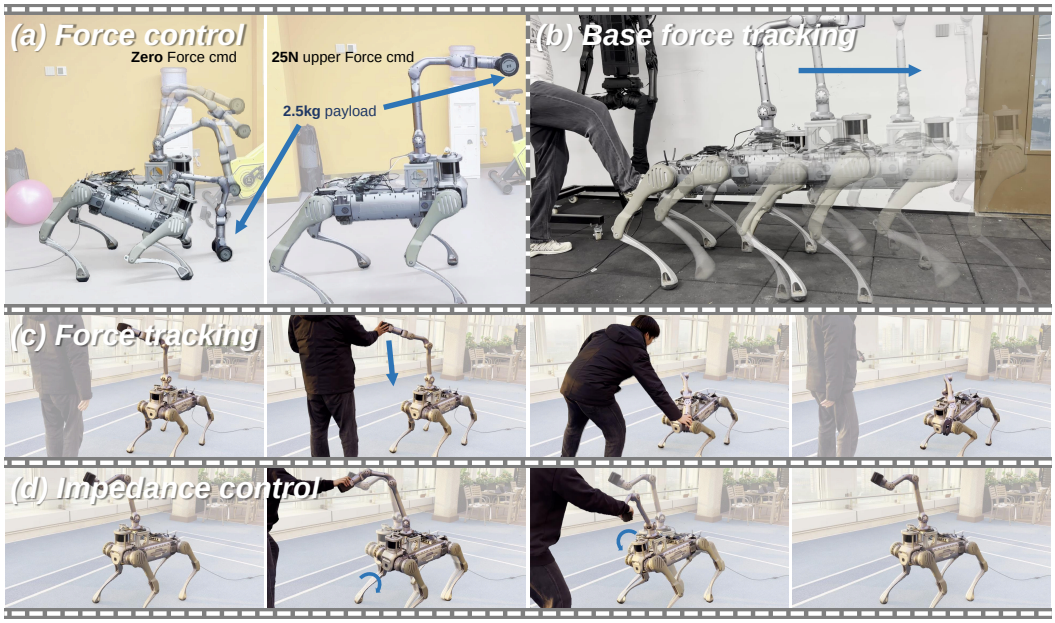

Figure 5: **Diverse skills facilitated by our policy.** (a) Force control: The robot counteracts gravity to support a payload when given a $25N$ force command. (d) Base force tracking: The robot responds compliantly to pushes on its base, enabling intuitive human guidance. (c) Force tracking: The robot tracks a zero-force command by minimizing external force interactions. (d) Impedance control: The robot adjusts its whole-body posture to counteract and comply with external disturbances.

Fig. 5(c) and the supplementary video, this capability is demonstrated by setting the force command to zero. In this case, the end-effector follows the external force and remains in the displaced position after the force is removed, achieving effective force tracking.

Impedance Control A key application of force control is impedance control, where the end-effector tracks a target position while responding compliantly to external forces, following the dynamics of a spring-mass-damper system. We implement impedance control using the unified policy, following Eq. (A.8). As shown in Fig. 5(d) and the supplementary video, we demonstrate this capability in human-robot tug-of-war and arm-wrestling scenarios. In these tasks, the further the end-effector deviates from the target position, the greater the resistive force exerted by the robot, showcasing impedance behavior.

4.4 Cross Embodiment Performance

To validate the cross-embodiment capability of our unified policy,we tested it on the Unitree G1 humanoid robot and Unitree B2-Z1 quadrupedal manipulator. For locomotion, unlike manipulation tasks where force compensation applies directly to the end-effector, we adjust the robot base velocity to compensate for external forces, following Eq. (4). As shown in the third row of Fig. 1, when the compensated velocity equals and opposes the velocity command, the humanoid robot halts and leans its body to maintain balance. Similarly, as shown in Fig. 5(b), the quadrupedal robot begins walking forward when kicked, even with zero force and velocity commands.

5 Conclusion

We propose a unified force-position control policy for legged robots, enabling contact-rich loco-manipulation tasks without explicit force sensors. Using reinforcement learning, our policy estimates external forces from historical states and compensates for them through position and velocity adjustments. This approach supports diverse behaviors like position tracking, force application, and compliance. Additionally, integrating force estimation into imitation learning improves task success in contact-rich environments. Experiments on quadrupedal and humanoid robots validate the policy's adaptability and robustness in real-world scenarios.

6 Limitations and Future Work

First, while the policy successfully estimates external forces without direct force sensing, its accuracy tends to degrade in high-frequency interactions and at the edges of the robot's workspace. Future work could focus on improving force estimation in these corner cases. One possible direction is to incorporate velocity and acceleration terms from Eq. (2) to enhance force estimation, allowing the model to better capture dynamic interactions.

Second, while our policy generalizes well from simulation to real-world deployment, discrepancies remain due to the sim-to-real gap, particularly in force accuracy along different coordinate axes. These differences likely stem from mismatches in actuator dynamics and contact modeling between simulation and real hardware. Future work could explore techniques such as domain randomization and real-to-sim corrections to improve robustness across varying real-world conditions.

Additionally, our current framework primarily focuses on estimating force at a single interaction point. Future work could explore multi-point force estimation and whole-body force interaction tasks. For example, in scenarios such as a quadrupedal robot opening a heavy door, the robot could use its body to brace against the door while simultaneously using its manipulator to press down on the handle. Developing policies that coordinate multiple contact forces across different body parts could enable more complex and effective real-world interactions.

References

[1] 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*, 4(26):eaau5872, 2019. 2, 3

[2] Z. Zhuang, S. Yao, and H. Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024. 2

[3] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi. Agile but safe: Learning collision-free high-speed legged locomotion. *arXiv preprint arXiv:2401.17583*, 2024. 2

[4] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024. 2

[5] J.-P. Sleiman, F. Farshidian, and M. Hutter. Versatile multicontact planning and control for legged loco-manipulation. *Science Robotics*, 8(81):eadg5014, 2023. 2, 3

[6] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023. 2, 3

[7] M. Liu, Z. Chen, X. Cheng, Y. Ji, R. Qiu, R. Yang, and X. Wang. Visual whole-body control for legged loco-manipulation. *The 8th Conference on Robot Learning*, 2024. 2, 3

[8] R.-Z. Qiu, Y. Song, X. Peng, S. A. Suryadevara, G. Yang, M. Liu, M. Ji, C. Jia, R. Yang, X. Zou, et al. Wildlma: Long horizon loco-manipulation in the wild. *arXiv preprint arXiv:2411.15131*, 2024. 2, 3

[9] T. Portela, G. B. Margolis, Y. Ji, and P. Agrawal. Learning force control for legged manipulation. *arXiv preprint arXiv:2405.01402*, 2024. 2, 3

[10] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023. 2

[11] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2023. 2

[12] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 2

[13] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 2

[14] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 2

[15] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 2

[16] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023. 2

[17] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 2

[18] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024. 2

[19] 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. 2, 3, 14

[20] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters*, 6 (3):4688–4695, 2021. 3

[21] M. P. Polverini, A. Laurenzi, E. M. Hoffman, F. Ruscelli, and N. G. Tsagarakis. Multi-contact heavy object pushing with a centaur-type humanoid robot: Planning and control for a real demonstrator. *IEEE Robotics and Automation Letters*, 5(2):859–866, 2020. 3

[22] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022. 3

[23] G. Pan, Q. Ben, Z. Yuan, G. Jiang, Y. Ji, J. Pang, H. Liu, and H. Xu. Roboduet: A framework affording mobile-manipulation and cross-embodiment. *arXiv preprint arXiv:2403.17367*, 2024. 3

[24] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter. Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators. *IEEE Robotics and Automation Letters*, 7(2):2377–2384, 2022. 3

[25] J. Wang, J. Rajabov, C. Xu, Y. Zheng, and H. Wang. Quadwbg: Generalizable quadrupedal whole-body grasping. *arXiv preprint arXiv:2411.06782*, 2024. 3

[26] M. P. Murphy, B. Stephens, Y. Abe, and A. A. Rizzi. High degree-of-freedom dynamic manipulation. In *Unmanned Systems Technology XIV*, volume 8387, pages 339–348. SPIE, 2012. 3

[27] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5682–5689. IEEE, 2015. 3

[28] B. U. Rehman, M. Focchi, J. Lee, H. Dallali, D. G. Caldwell, and C. Semini. Towards a multi-legged mobile manipulator. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3618–3624. IEEE, 2016. 3

[29] C. D. Bellicoso, K. Krämer, M. Stäuble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter. Alma-articulated locomotion and manipulation for a torque-controllable robot. In *2019 International conference on robotics and automation (ICRA)*, pages 8477–8483. IEEE, 2019. 3

[30] M. Risiglione, V. Barasuol, D. G. Caldwell, and C. Semini. A whole-body controller based on a simplified template for rendering impedances in quadruped manipulators. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9620–9627. IEEE, 2022. 3

[31] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. 1981. 3, 13

[32] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981. 3

[33] T. Yoshikawa. Dynamic hybrid position/force control of robot manipulators–description of hand constraints and calculation of joint driving force. *IEEE Journal on Robotics and Automation*, 3(5):386–392, 1987. 3

[34] N. Hogan. Impedance control: An approach to manipulation. In *1984 American control conference*, pages 304–313. IEEE, 1984. 3

[35] X. Zhang, L. Sun, Z. Kuang, and M. Tomizuka. Learning variable impedance control via inverse reinforcement learning for force-related tasks. *IEEE Robotics and Automation Letters*, 6(2):2225–2232, 2021. 3

[36] Y. Hou, Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, S. Feng, B. Burchfiel, and S. Song. Adaptive compliance policy: Learning approximate compliance for diffusion guided control. *arXiv preprint arXiv:2410.09309*, 2024. 3

[37] J. de Wolde, L. Knoedler, G. Garofalo, and J. Alonso-Mora. Current-based impedance control for interacting with mobile manipulators. *arXiv preprint arXiv:2403.13079*, 2024. 3

[38] K. Bousmalis, G. Vezzani, D. Rao, C. M. Devin, A. X. Lee, M. B. Villalonga, T. Davchev, Y. Zhou, A. Gupta, A. Raju, et al. Robocat: A self-improving generalist agent for robotic manipulation. *Transactions on Machine Learning Research*, 2023. 3

[39] O. Mees, D. Ghosh, K. Pertsch, K. Black, H. R. Walke, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, et al. Octo: An open-source generalist robot policy. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024. 3

[40] Q. Vuong, S. Levine, H. R. Walke, K. Pertsch, A. Singh, R. Doshi, C. Xu, J. Luo, L. Tan, D. Shah, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023. 3

[41] J. Yang, D. Sadigh, and C. Finn. Polybot: Training one policy across robots while embracing variability. *arXiv preprint arXiv:2307.03719*, 2023. 3

[42] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 653–660. IEEE, 2024. 3

[43] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 3

[44] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars, 2016. URL https://arxiv.org/abs/1604.07316. 3

[45] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning (CoRL)*, 2024. 3

[46] 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. 3

[47] Z. He, K. Lei, Y. Ze, K. Sreenath, Z. Li, and H. Xu. Learning visual quadrupedal loco-manipulation from demonstrations. *arXiv preprint arXiv:2403.20328*, 2024. 3

[48] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning visuotactile skills with two multifingered hands. *arXiv preprint arXiv:2404.16823*, 2024. 3

[49] W. Yang, A. Angleraud, R. S. Pieters, J. Pajarinen, and J.-K. Kämäräinen. Seq2seq imitation learning for tactile feedback-based manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5829–5836. IEEE, 2023. 3

[50] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 14

A Problem Formulation

Our policy enables a wide range of manipulation behaviors under varying force and position inputs, including position control, force control, impedance control, and hybrid position and force control. Under the formulation of Eq. (2), we can easily derive the following basic manipulation policies:

a) Position Control: When there is no external disturbance or active force command applied, Eq. (2) becomes

$$\mathbf{x}^{\text{target}} = \mathbf{x}^{\text{cmd}}, \tag{A.6}$$

where the target end-effector position $\mathbf{x}^{\text{target}}$ should reach the commanded position $\mathbf{x}^{\text{cmd}}$.

b) Force Control: When in contact with the environment and applying a force $\boldsymbol{F}^{\text{cmd}}$ without external disturbances, the desired goal position of the end-effector is defined following Eq. (2) as:

$$\mathbf{x}^{\text{target}} = \mathbf{x}^{\text{cmd}} + \frac{(\boldsymbol{F}^{\text{cmd}} - \boldsymbol{F}^{\text{react}})}{K}. \tag{A.7}$$

During the system execution, the reaction force $\boldsymbol{F}^{\text{react}}$ gradually increases to match the force command $\boldsymbol{F}^{\text{cmd}}$, leaving the final target pose $\mathbf{x}^{\text{target}}_{\text{final}} = \mathbf{x}^{\text{cmd}}$.

c) Impedance Control: When the end-effector is subjected to an external disturbance force but does not apply force to the environment, Eq. (2) simplify to:

$$\mathbf{x}^{\text{target}} = \mathbf{x}^{\text{cmd}} + \frac{\boldsymbol{F}^{\text{ext}}}{K}, \tag{A.8}$$

where the end-effector, when subjected to external disturbances, adjusts its position to exhibit compliance in response to the external force $\boldsymbol{F}^{\text{ext}}$. Notably, we can also implement force tracking and gravity compensation using Eq. (A.8) by dynamically adjusting the position command $\mathbf{x}^{\text{cmd}}$ following:

$$\Delta\mathbf{x}^{\text{cmd}} = \frac{\boldsymbol{F}^{\text{ext}}}{K}, \tag{A.9}$$

where $\boldsymbol{F}^{\text{ext}}$ can be the gravity term or an external force.

d) Hybrid Position and Force Control: As defined in [31], hybrid position and force control refers to controlling the end-effector's movement using $\mathbf{x}^{\text{cmd}}$ while applying a force command $\boldsymbol{F}^{\text{cmd}} = \boldsymbol{F}^{\text{cmd}}_{\perp}$ perpendicular to the tangential direction of the movement. In this scenario, the system follows Eq. (A.6) along the tangential direction without force command and satisfies Eq. (A.7) in the perpendicular direction where the force command is active.

Simplified impedance model. In our formulation, the damping and inertia terms are omitted. This simplification is suitable for relatively static or quasi-static tasks (e.g., wiping or tug-of-war), where motion is slow and normal force ensures contact. In these cases, the static model is sufficient. For dynamic or agile skills, higher control frequencies and explicit velocity/acceleration terms will be necessary, which we plan to explore in future work.

Assumption of rigid contact. Our formulation $\boldsymbol{F} = K(\mathbf{x} - \mathbf{x}^{\text{des}})$ naturally handles both rigid and compliant objects, as the end effector position depends only on the net force $\boldsymbol{F}$ and the desired goal position $\mathbf{x}^{\text{des}}$, regardless of the stiffness of the object. For rigid contacts, small displacements generate the required force, whereas for soft objects, larger deformations occur until the force equilibrium is reached.

B Hardware Settings and Teleoperation System

Robot System Setup The humanoid robot system (Fig. 2) is the 29-DOF Unitree G1 robot. The quadruped robot system (Fig. A.6a) consists of a 12-DOF Unitree B2 robot and a 6-DOF Unitree Z1 robot arm, both powered by the battery of B2. We customize two RealSense cameras mounted on

(a) **B2-Z1 robot hardware.** A Unitree B2 robot with a Z1 arm is teleoperated via a wireless controller, with two RealSense cameras for visual input.

(b) **Iphone teleoperation.** Using the MuJoCo AR app, an iPhone is employed to remotely control the robot for performing manipulation tasks.

Figure A.6: **Hardware setting and teleoperation system.**

the arm and head of the quadruped robot. Our whole-body controller and diffusion policy inference are executed on a dedicated desktop with an RTX 3090 GPU, interfaced with the B2-Z1 robot via an internet connection.

Teleoperation System As shown in Fig. A.6b, for manipulation tasks that require only position control, we utilize the MuJoCo AR application, which allows flexible teleoperation of the robot using an iPhone. However, as illustrated in Fig. A.6a, this approach is no longer suitable for manipulation tasks involving hybrid force and position control. To address this, we developed a dedicated teleoperation system based on the B2 robot's built-in wireless controller. In this system, two joysticks are used to control the robot's base movement, while eight buttons below are mapped to control the end-effector's position and the opening/closing of the gripper. Additionally, by holding the "L1" button, the system switches from issuing position commands to force commands for the end-effector. In practice, data collection with this setup was relatively slow, e.g., approximately 20 seconds per wiping trial and 10 seconds per cabinet trial, mainly due to the limited bandwidth of teleoperation. We believe that exoskeleton-based teleoperation with force feedback will offer a more natural and efficient way to collect demonstrations for forceful manipulation, and we plan to explore this direction in future work.

C Details on Policy Learning with Reinforcement Learning

We utilize Proximal Policy Optimization (PPO) [50] to train the actor policy and implement the state estimator with a multi-layer perception (MLP) network for state prediction outputs. We train our RL policy in Isaac Gym [19] with 4096 parallel environments.

C.1 Input Commands and Disturbance Forces

We sample input commands and disturbance forces within the ranges below during training:

1. End-effector position command in spherical coordinates within the body frame $\mathbf{x}_{\text{ee}}^{\text{cmd}} = (r^{\text{cmd}}, \theta^{\text{cmd}}, \phi^{\text{cmd}})$, where $r^{\text{cmd}} \in [0.35, 0.85\,\text{m}]$, $\theta^{\text{cmd}} \in [-0.4\pi, 0.4\pi\,\text{rad}]$, $\phi^{\text{cmd}} \in [-0.6\pi, 0.6\pi\,\text{rad}]$.
2. End-effector force command in cartesian coordinates within the body frame $\boldsymbol{F}_{\text{ee}}^{\text{cmd}} \in \mathbb{R}^3$: $[-60N, 60N]$.
3. Base velocity command $\mathbf{v}_{\text{base}}^{\text{cmd}} = (v_x^{\text{cmd}}, v_y^{\text{cmd}}, \omega_z^{\text{cmd}})$, where $v_x \in [-0.8, 0.8\,\text{m/s}]$, $v_y \in [-0.6, 0.6\,\text{m/s}]$, $\omega_z \in [-0.8, 0.8\,\text{rad/s}]$.
4. Base force command within the body frame $\boldsymbol{F}_{\text{base}}^{\text{cmd}} \in \mathbb{R}^3$: $[-60N, 60N]$.

Table A.1: **Reward terms** for learning the whole-body policy.

Term	*Equation*	*Weight*
end-effector Unified Position and Force Control		
gripper position	$\exp\{-\lvert\mathbf{x}_{\text{ee}} - (\mathbf{x}^{\text{cmd}} + (\boldsymbol{F}^{\text{ext}} + \boldsymbol{F}^{\text{cmd}} - \boldsymbol{F}^{\text{react}})/B\rvert/0.5\}$	2.0
Base Unified Position and Force Control ($\mathbf{r}_v^b$)		
base velocity	$\exp\{-\lvert\mathbf{v}_{\text{base}} - (\mathbf{v}_{\text{base}}^{\text{cmd}} + F_{\text{base}}/D)\rvert/0.25\}$	2.0
Safety and Smoothness		
collision penalty	$\mathbb{1}_{\text{collision}}$	-5.0
joint limit	$\mathbb{1}_{q>0.8*q^{max}\|\|q<0.8*q^{min}}$	-10.0
torques	$\lvert\tau\rvert^2$	-5×10^{-6}
joint velocities	$\lvert\dot{q}\rvert^2$	-8×10^{-4}
joint acceleration	$\lvert\ddot{q}\rvert^2$	-2×10^{-7}
action rate	$\lvert a_{t-1} - a_t\rvert$	-0.02
torque limit	$\mathbb{1}_{\tau>0.9*\lvert\tau^{max}\rvert}$	-0.005
Gait		
contact number	$\sum_{\text{foot}} \mathbb{1}_{\tau_{contact}>5.} * stance\_mask$	2.0
reference motion	$\lvert q - q^{ref}\rvert^2$	1.0

Table A.2: **Domain radomization** for learning the whole-body policy.

Term	*Unit*	*Range*
Friction	-	$[0.3,\ 2.0]$
Body Mass	kg	$[0.0,\ 15.0]$
base com (x,y,z axis)	m	$[-0.15,\ 0.15]$
Motor Strength	$\%$	$[85,\ 115]$
Gripper Payload	kg	$[0.0,\ 0.5]$
Push robot	m/s	$[0.0,\ 0.8]$, interval = 8s

5. External net force from the environment at the end-effector $\boldsymbol{F}_{\text{ee}} \in \mathbb{R}^3$: $[-60N, 60N]$ and at robot base $\boldsymbol{F}_{\text{base}} \in \mathbb{R}^3$: $[-60N, 60N]$.

C.2 Reward and Domain Radomization

Table A.1 provides a detailed overview of the reward structure employed in this study, and Table A.2 outlines the adopted domain randomization scheme.

C.3 World-Aligned End-Effector Position Estimation

Our estimator predicts not only the external force, but also the end-effector position and base linear velocity. Although forward kinematics provides the end-effector position relative to the arm base, it decouples arm control from base posture. In contrast, we estimate the end-effector position in a world-aligned base frame (with fixed height and orientation relative to the base projection). This representation allows, for example, a downward end-effector command to naturally induce robot base leaning near workspace limits, enabling coordinated whole-body behavior without explicit base

control. As the end-effector position in this frame cannot be directly measured, we estimate it instead.

C.4 Additional Analyses on Policy Learning

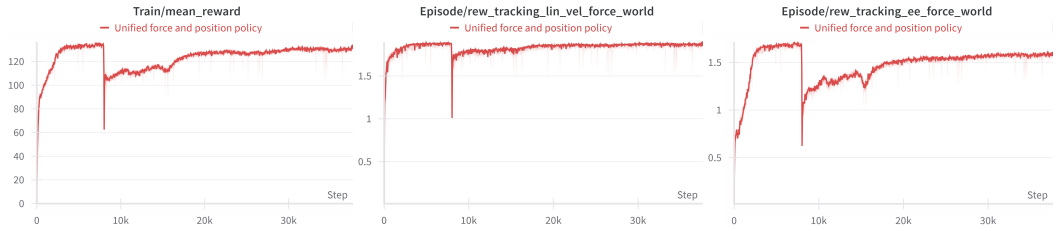

Figure A.7: Training reward curve (view with zoom-in).

Introducing external disturbances early makes training challenging, as initial policies struggle with body balance and end-effector stability. To address this, we used a two-stage curriculum: first, training whole-body reaching and locomotion, then adding random force commands and disturbances. The training reward curve (Fig. A.7) shows an initial drop in stage two, with locomotion recovering and whole-body reaching stabilizing, despite a slight reaching reward decrease due to increased sampling diversity of force commands and disturbances.

C.5 Impact of Motor Gains

Compared to pure position-based whole-body control, we use smaller Kp/Kd gains. We observed that lower gains, which allow greater overshoot, improve force estimation. However, excessively low values degrade position tracking accuracy.

C.6 Sim-to-real Gap and Force Estimation Robustness

Our force estimator experiences sim2real gaps due to mismatches in motor dynamics and contact modeling. While domain randomization (e.g., varying Kp/Kd gains) helps reduce these gaps, sim2real transfer remains challenging for RL-based policies. Inspired by how humans rely more on tactile feedback than precise force sensing in contact-rich tasks, our IL policy combines estimated forces with visual input to achieve robust performance without requiring high force accuracy. To further reduce the sim2real gap, we plan to fine-tune the estimator with real-world data and apply system identification methods.

D Details on Force-aware Imitation Learning Policy

Table A.3: Imitation learning results (**50 trials per task**)

Task	`wipe-blackboard`	`open-cabinet`	`close-cabinet`	`open-drawer-occlusion`
w/o Force	0.22	0.36	0.30	0.30
w/ Force	0.58	0.70	0.72	0.76

Task Settings We select four tasks that require a combination of hybrid force and position control and force sensing capabilities in the real world.

- For the `wipe-blackboard` task, the robot must press against the blackboard while moving laterally to effectively remove ink marks. If the robot loses contact or fails to apply sufficient force during movement, the ink will not be wiped away. A purely position-controlled approach struggles with maintaining consistent contact force, often resulting in intermittent contact or excessive

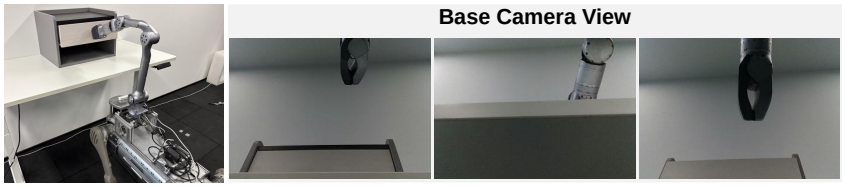

Figure A.8: Open drawer with occlusion, the gripper becomes **occluded** during manipulation.

force application. In contrast, our force estimator enables the robot to sustain stable contact pressure, ensuring effective wiping while minimizing the risk of excessive force that could damage the robot.

- For the two tasks of `open-` and `close-cabinet` with a push-to-open cabinet, the robot must apply sufficient force to press the door, triggering the built-in mechanism that causes it to spring open or close. Unlike the previous task, this scenario requires the robot to exert enough force to overcome the mechanism's resistance, despite the minimal displacement during activation.
- For the task of `open-drawer-occlusion` with a rebound mechanism, the robot must apply sufficient force to press on the drawer front under visual occlusion, triggering the built-in mechanism that makes the drawer spring open, as shown in Fig. A.8.

Evaluation We provide the quantitative comparison between our method and the baseline in Table A.3. And specifically:

- For the task of `wipe-blackboard`, success is defined as the robot erasing 90% of the ink marks. Failure occurs if the robot exceeds the time limit without achieving this goal.
- For the two tasks of `open-` and `close-cabinet` with a rebound mechanism, success is defined as the robot pressing the cabinet door to open (or close) the cabinet and then releasing the surface. Failure occurs if the robot is unable to activate the mechanism or does not release the cabinet door after pressing it.
- For the task of `open-drawer-occlusion` with a rebound mechanism, success is defined as the robot pressing the drawer front under visual occlusion and then releasing the surface. Failure occurs if the robot is unable to activate the mechanism or does not release the drawer front after pressing it.

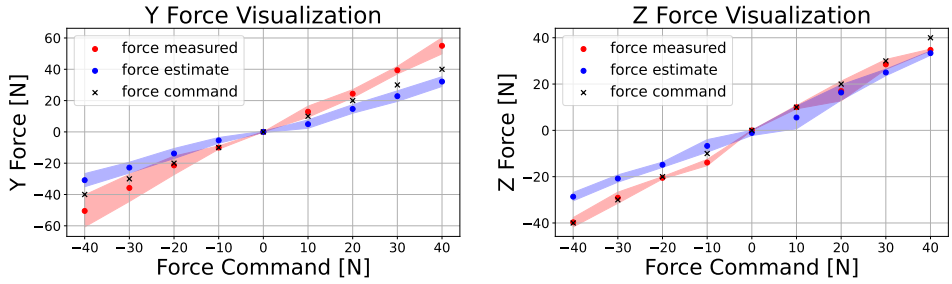

Figure A.9: Real-world force control evaluation.

E Assessing Force Estimation Accuracy Along X and Y Axes

We evaluate the force estimator by randomly selecting five positions and applying forces ranging from -60N to 60N along the X-axis (as in Fig. 4 of the main paper). We additionally evaluate on the Y- and Z-axis within 40N (due to hardware constraints of Unitree-Z1) in Fig. A.9. While sim-to-real

discrepancies introduce inaccurate estimations, especially along Y-axis, we argue that the current estimator suffices for the coarse-grained manipulation tasks discussed in this study. Reducing this sim-to-real gap for finer control will be one focus Additional of our future work.