FACET: Force-Adaptive Control via Impedance Reference Tracking for Legged Robots

Botian Xu[*1] **Haoyang Weng**[*1] **Qingzhou Lu**[*1 *] **Yang Gao**[123] **Huazhe Xu**[123]
[1]Tsinghua University, [2]Shanghai Qizhi Institute, [3]Shanghai AI Lab
Project Website: https://facet.pages.dev/

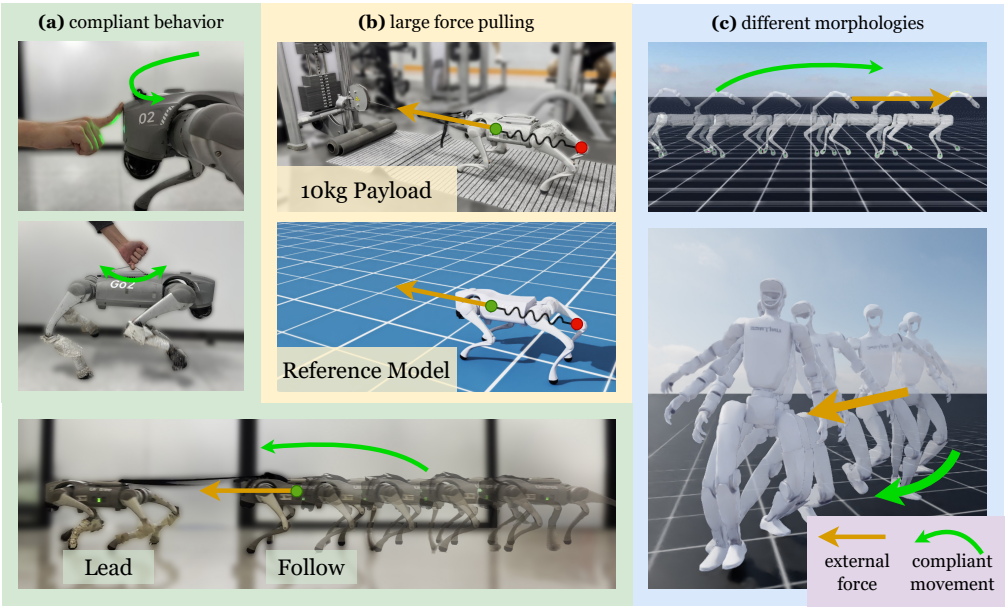

Figure 1: Inspired by impedance control, **FACET** enables task-space variable compliance and force-adaptive control on legged robots by imitating a reference spring-mass-damper model (b) using reinforcement learning. A high compliance allows the robot to be stopped or kinesthetically guided with ease (a), while a high stiffness allows the robot to exert large forces when pushing/pulling a payload (b). The framework applies to different morphologies and more complex configurations (c).

Abstract: Reinforcement learning (RL) has made significant strides in legged robot control, enabling locomotion across diverse terrains and complex loco-manipulation capabilities. However, the commonly used position or velocity tracking-based objectives are agnostic to forces experienced by the robot, leading to stiff and potentially dangerous behaviors and poor control during forceful interactions. To address this limitation, we present *Force-Adaptive Control via Impedance Reference Tracking* (FACET). Inspired by impedance control, we use RL to train a control policy to imitate a virtual mass-spring-damper system, allowing fine-grained control under external forces by manipulating the virtual spring. In simulation, we demonstrate that our quadruped robot achieves improved robustness to large impulses (up to 200 Ns) and exhibits controllable compliance, achieving an 80% reduction in collision impulse. The policy is deployed to a physical robot to showcase both compliance and the ability to engage with large forces by kinesthetic control and pulling payloads up to $2/3$ of its weight. Further extension to a legged loco-manipulator and a humanoid shows the applicability of our method to more complex settings to enable whole-body compliance control.

Keywords: Reinforcement Learning, Legged Robot Control, Impedance Control

[*]equal contribution.

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

1 Introduction

Reinforcement learning (RL) has powered many recent advancements in legged robot control, showcasing remarkable agility, robustness, and expressiveness. RL-based locomotion policies enable robots to navigate a diverse range of terrains [1, 2] and tackle complex parkour challenges [3, 4, 5]. Additionally, RL-based loco-manipulation has also received increasing attention for its potential to achieve whole-body coordination to fully leverage the degrees of freedom offered by the robot. Successful applications include pick and place [6, 7, 8], door-opening [9, 10], and ball shooting [11].

Despite the impressive progress, existing RL-based controllers often overlook or greatly simplify interactions involving external forces. Specifically, most locomotion controllers are trained to track commanded velocities regardless of the external force experienced, resulting in stiff and jerky motions [12]. Similarly, manipulation controllers are trained to track commanded end-effector poses, often assuming that the forces involved are negligible or can be easily counteracted, e.g., the objects to interact with have negligible mass. The limited consideration of forceful interaction poses issues when the robot operates in complex, human-populated environments. The robot's response when intervened by humans or upon collision is either stiff or unspecified, leading to potential damage to both the robot and its surroundings.

To address these limitations and provide more fine-grained control for legged robots under forceful interactions, we propose a reinforcement learning framework named *Force-Adaptive Control via Impedance Reference Tracking* (FACET). Impedance control [13] has been widely used in robotic manipulation to control the end-effector with indirect force specification. Inspired by its formulation, we model the robot's behavior with a virtual mass-spring-damper system and utilize RL to train the robot to follow its dynamics. To achieve efficient training, we design a reference model, i.e., a virtual mass-spring-damper system defined on the center of mass (CoM) of the robot, to generate smooth tracking targets and use a teacher-student training recipe for sim-to-real transfer.

Experiments on multiple robots demonstrate the advantage of our force-adaptive controllers in terms of improved robustness and controllable compliance: a quadruped robot can survive impact forces exceeding twice its weight and significantly reduce the force required to either stop or move it. We also showcase an extension to a legged loco-manipulation system to underscore its potential for more complex forceful interaction scenarios.

The contributions of this work are summarized as follows:

- We propose FACET to enable compliant and force-adaptive behaviors on legged robots by learning to imitate the dynamics of a virtual mass-spring-damper model. The key technical novelty of our method is to leverage a reference model to generate position–velocity tracking targets under external perturbations, while exposing the virtual impedance parameters $(\mathbf{x}_{\text{des}}, K_p, K_d)$ as part of the control interface. This implicit force command interface enables intuitive simultaneous regulation of motion and interaction forces through simple manipulation of the spring–damper gains.
- We validate FACET across three robot platforms (quadruped Go2, humanoid G1, and quadruped with mounted arm B1+Z1) in simulation and on real quadruped robot. In simulation, we show that FACET: (i) survives much larger external impulses by following rather than resisting them; (ii) produces significantly smaller collision impulses against a wall by tuning stiffness; and (iii) outperforms baselines and ablations in tracking error and success rate. On hardware, we demonstrate intuitive compliant following under manual guidance and reliable pulling of payloads up to 10 kg, confirming that FACET delivers robust, soft, and high-force interactions.

2 Related Work

Reinforcement learning for legged robot control. Reinforcement learning has become a promising tool for training locomotion controllers for quadrupedal and bipedal legged robots in various scenarios [1, 2], especially favored for agility [4, 14]. For the robot to operate reliably in complex

environments, many prior works have aimed to improve robustness and adaptability under external forces by applying perturbations during training [12, 15], enhancing state estimation [16, 17], or learning some special behaviors to avoid damage [18, 19]. However, as discussed in Deep Compliant Control (DMC) [12], the commonly used velocity tracking reward function produces highly stiff control policies. This issue arises because the velocity tracking objective does not specify appropriate responses to external forces or always seeks to reject. Although DMC identified this issue and designed a recovery stage after the perturbation is applied, it does not offer a principled way to control the level of compliance. The same issue also happens in RL-based loco-manipulators, which use position tracking as the objective for end-effector control [6, 7, 20].

Force and impedance control with reinforcement learning. Hybrid position-and-force control is desirable for many robotic tasks requiring forceful interactions with the environment. Direct force control and impedance control [13] have been well studied in robotic manipulation in both model-based and learning-based settings [21, 22, 23, 24, 25], but remain undeveloped for legged robots. The complex and contact-rich movement of the legs makes it difficult to accurately measure or track the required ground reaction force, underscoring the potential of RL-based controllers. Some prior works explored torque-based action spaces for legged robots [26, 27, 28] to improve motion smoothness, robustness, and sim-to-real transfer. While introducing compliance to the joint space, they do not directly lead to compliance motion in Cartesian space. Research in physics-based character control [29] introduced compliance to motion tracking controllers by altering the reference motion. Closest to our work, [8] proposed a force control objective for quadrupedal loco-manipulation systems [6, 30, 7, 31]. Their method allows users to command the force at the end-effector, greatly enhancing the capability for forceful interaction. However, it requires explicitly switching between a position-tracking mode and a force mode. Meanwhile, it assumes the system is near static, such that the force at the end-effector can be approximated by the external force applied. A comparison of different objectives used in RL-based control policies is shown in Tab. 1.

Table 1: Comparison of the control capabilities achieved by different training objectives.

Method	Position	Velocity	Compliance	Force	Inertia*	Dynamic
Position tracking	✓	✗	✗	✗	✗	-
Velocity tracking, e.g. [1, 2]	✓	✓	✗	✗	✗	-
Deep Compliant Control [12]	✓	✓	✓*	✗	✗	-
Learning Force Control [8]	✓	✓	✓	✓	✗	✓†
FACET	✓	✓	✓	✓	✓	✓

* inertia shaping allows for manipulating a virtual mass of the system to control its acceleration behavior
* the compliance does not have a principled form and can not be controlled after training
† a near-static assumption is needed for the force output to be approximated by external forces applied

3 Background

Impedance control for robotic manipulation. Compliance and force control are crucial in many manipulation tasks requiring complex forceful interaction. Impedance control [13] can achieve task space compliance through an indirect force control interface. The controller takes as input the desired position $\mathbf{x}_{\text{des}}$ and velocity $\dot{\mathbf{x}}_{\text{des}}$ of a *setpoint*, along with stiffness K_p and damping K_d that define a virtual spring-damper system. The goal is to output joint-space torques that produce an end-effector force/torque in task space according to:

$$f_{\text{spring}} = K_p(\mathbf{x}_{\text{des}} - \mathbf{x}) + K_d(\dot{\mathbf{x}}_{\text{des}} - \dot{\mathbf{x}}), \tag{1}$$

where $\mathbf{x}$ and $\dot{\mathbf{x}}$ denote the translational and rotational position and velocity of the body.

For fixed-base systems such as robotic arms, impedance control can be implemented by mapping the task-space force f_{spring} into joint-space torques. Rather than commanding a sequence of positions directly, the user controls a virtual spring by appropriately adjusting the setpoints and the impedance gains K_p, K_d, then the controller will generate desired interaction forces [32] according to Eq. (1) to manipulate an object.

4 Method

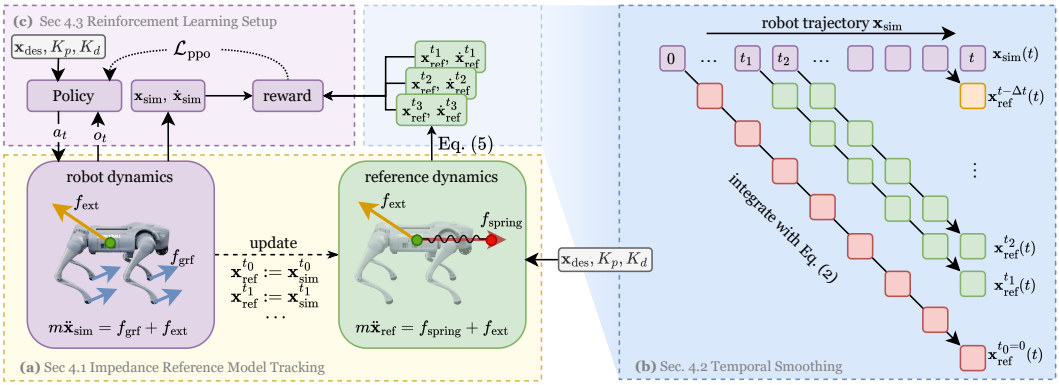

Figure 2: Method overview. **(a)**: The policy is trained to imitate the dynamics of an impedance reference model, i.e., a virtual mass-spring-damper system defined on the CoM of the robot (Sec. 4.1). **(b)**: The reference dynamics is integrated from different starts along the robot trajectory to generate the reference targets, which provides tracking rewards for the policy. (Sec. 4.2). **(c)** The policy receives the same set of impedance parameters $(\mathbf{x}_{\text{des}}, K_p, K_d)$ and is optimized to produce the same trajectory as that of the reference model (Sec. 4.3).

4.1 Impedance Reference Model Tracking

Impedance control has proven effective in achieving compliant and force-adaptive behaviors for robotic manipulators. Inspired by its success, we extend impedance control principles to legged robots, aiming to regulate the center of mass (CoM) dynamics through a virtual mass-spring-damper model, so that the CoM responds as if tethered to its target by a virtual spring.

To realize this, we define a reference model whose dynamics satisfy

$$m\ddot{\mathbf{x}}_{\text{ref}} = f_{\text{spring}} + f_{\text{ext}} = K_p(\mathbf{x}_{\text{des}} - \mathbf{x}_{\text{ref}}) + K_d(\dot{\mathbf{x}}_{\text{des}} - \dot{\mathbf{x}}_{\text{ref}}) + f_{\text{ext}} \tag{2}$$

where m, K_p, and K_d are the virtual mass, stiffness, and damping; $(\mathbf{x}_{\text{des}}, \dot{\mathbf{x}}_{\text{des}})$ the commanded setpoint; f_{ext} the external forces and torques applied to the system, e.g., a push or kick at the trunk, while the robot is modeled with a simplified dynamics:

$$m\ddot{\mathbf{x}}_{\text{sim}} = f_{\text{grf}} + f_{\text{ext}}. \tag{3}$$

Our policy thus takes as input the impedance parameters $(\mathbf{x}_{\text{des}}, K_p, K_d)$, in contrast to velocity-tracking controllers or direct force-control approaches, which use explicit velocity or force targets as their input interface. This indirect force commanding interface enables a variety of intuitive, force-centric tasks. For example, by setting $K_p = 0$, a human can easily guide the robot with minimal effort with a string attached. Additionally, kinesthetic teaching for loco-manipulators becomes feasible for wall-swiping tasks by moving the setpoint along the surface while maintains a compliant end-effector contact force.

As illustrated in Fig. 2 (lower left), both the robot's CoM and the reference model experience the same external perturbation f_{ext}. The reference model additionally experiences the spring force f_{spring} while the robot experiences ground reaction forces f_{grf} through contact with the ground. In principle, realizing the dynamics in Eq. (2) requires controlling the robot to generate f_{grf} through feet contact to match the virtual spring force f_{spring}. However, this is challenging due to the complex contact dynamics and various constraints associated with the robot. Reliable sensing of f_{ext} is also infeasible due to the absence of force sensors in most robots.

To circumvent these issues, our method resorts to RL to track the reference trajectory generated by the virtual mass-spring-damper system. Specifically, we integrate the reference dynamics Eq. (2)

over time to produce smooth position and velocity trajectories $(\mathbf{x}_{\text{ref}}, \dot{\mathbf{x}}_{\text{ref}})$, yielding the objective:

$$\mathcal{L}_{\text{track}} = \|\mathbf{x}_{\text{sim}} - \mathbf{x}_{\text{ref}}\|_2^2 \tag{4}$$

$$+ \|\dot{\mathbf{x}}_{\text{sim}} - \dot{\mathbf{x}}_{\text{ref}}\|_2^2 \tag{5}$$

This approach bypasses the need for direct force or acceleration tracking and enables stable and robust training despite the complex contact dynamics inherent to legged locomotion.

4.2 Temporal Smoothing of Tracking Targets

As illustrated in Fig. 2 (b), we obtain the reference state at time t by integrating Eq. (2):

$$\dot{\mathbf{x}}_{\text{ref}}^{t_0}(t) = \dot{\mathbf{x}}_{\text{sim}}(t_0) + \int_{t_0}^{t} \ddot{\mathbf{x}}_{\text{ref}}(t)dt, \; \mathbf{x}_{\text{ref}}^{t_0}(t) = \mathbf{x}_{\text{sim}}(t_0) + \int_{t_0}^{t} \dot{\mathbf{x}}_{\text{ref}}^{t_0}(t)dt. \tag{6}$$

where $\mathbf{x}_{\text{sim}}(t_0)$ and $\dot{\mathbf{x}}_{\text{sim}}(t_0)$ represent the robot's actual state at timestep t_0.

Integrating from the beginning of each episode gives us the open-loop tracking target $(\mathbf{x}_{\text{ref}}^{t_0=0}(t), \dot{\mathbf{x}}_{\text{ref}}^{t_0=0}(t))$. It faithfully follows Eq. (2) but does not consider the gap between the robot dynamics and the reference model caused by physical constraints, e.g., the acceleration a robot can produce depends on its motor capabilities and contact state with the ground. In contrast, integrating from the most recent states leads to closed-loop tracking target $(\mathbf{x}_{\text{ref}}^{t-\Delta t}(t), \dot{\mathbf{x}}_{\text{ref}}^{t-\Delta t}(t))$, which is adaptive to the robot's real state but can be noisy and short-sighted to provide sufficient reward signal for RL. Here, $\Delta t = 0.02s$ is the control interval of the policy.

To address these issues, we introduce *temporal smoothing*: instead of a single target, we use a mixture of targets obtained by integrating from different time steps $t' \in \{t_1, t_2, \cdots, t_M\}$ along the robot's trajectory. The reward function is then defined to have the form:

$$r_t = \frac{1}{M} \sum_{t'} \exp(-||\mathbf{x}_{\text{sim}}(t) - \mathbf{x}_{\text{ref}}^{t'}(t)||_2^2) + \exp(-||\dot{\mathbf{x}}_{\text{sim}}(t) - \dot{\mathbf{x}}_{\text{ref}}^{t'}(t)||_2^2). \tag{7}$$

In our implementation, we use a smoothed tracking targets with $t' \in \{t - 8\Delta t, t - 16\Delta t, t - 32\Delta t\}$. Note that choosing $t' \in \{0\}$ or $t' \in \{t - \Delta t\}$ recovers the open-loop or close-loop case, respectively.

4.3 Reinforcement Learning for Impedance Tracking

Observation and action spaces Following Eq. (2), the command input to the robot is $\mathbf{u} = (\mathbf{x}_{\text{des}}, K_p, K_d, m)$. For a legged robot, we are concerned with the planar translation and yaw-rotation $\mathbf{x}_{\text{des}} = (x, y, \theta_z)$. The policy is trained using a teacher-student framework: the student observes the command, proprioceptive information, and previous actions $o_t = (\mathbf{u}, \mathbf{q}, \dot{\mathbf{q}}, \mathbf{g}, a_{t-3:t-1})$, while the teacher additionally observes $o_t^{\text{priv}} = (\dot{\mathbf{x}}, \ddot{\mathbf{x}}, \boldsymbol{\tau}, f_{\text{ext}}, \mathbf{c})$. At each time step, the policy outputs action $\mathbf{a}_t$, which are translated to joint position targets $\mathbf{q}_{\text{des}}$ and then tracked by a PD controller. During training, we sample various command and external force distributions to expose the policy to a variety of profiles for forceful interactions. The details of sampling distributions can be found in Sec. 10.1.4.

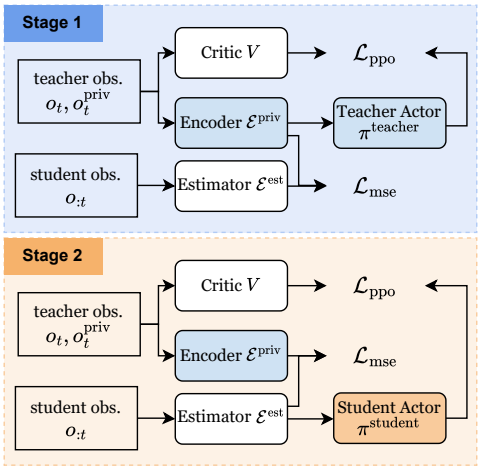

Figure 3: We train a **state estimator** $\mathcal{E}^{\text{est}}$ to predict the feature extracted by the **privileged encoder** $\mathcal{E}^{\text{priv}}$. In the second stage, the student actor π^{student} is initialized with the teacher's parameters and continues to be finetuned.

Sim-to-real transfer via teacher-student training In the real-world, direct measurements of body velocity and external forces are unavailable due to the

lack of such sensors. The reference model is therefore also unavailable. During training, a control policy must learn to infer such *privileged information* to be deployed in the real world.

We adopt a two-stage teacher-student framework similar to RMA [33]. When training the teacher policy π^{teacher} in the first stage, a state estimator **state estimator** $\mathcal{E}^{\text{est}}(o_{:t})$ is trained at the same time to predict the privileged feature extracted by the **privileged encoder** $\mathcal{E}^{\text{priv}}(o_t, o_t^{\text{priv}})$. Different from [33], we initialize the student actor π^{student} with the teacher's parameters and continue to train it using PPO in the second stage. This way, the student is driven to find the optimal policy with imperfect estimation instead of faithfully imitating the teacher.

Following the asymmetric actor-critic paradigm, the critic $V(o_t, o_t^{\text{priv}})$ always receives both observable and privileged information to ensure accurate value estimation. The policy is trained using PPO [34] in IsaacLab [35]. More implementation and training details can be found in Sec. 10.1.

5 Simulation Experiments

In this section, we conduct comprehensive experiments in simulation to validate the efficacy of our method and design. The domain randomizations used during training, e.g., random perturbations to joint parameters and body mass, are also applied during evaluation.

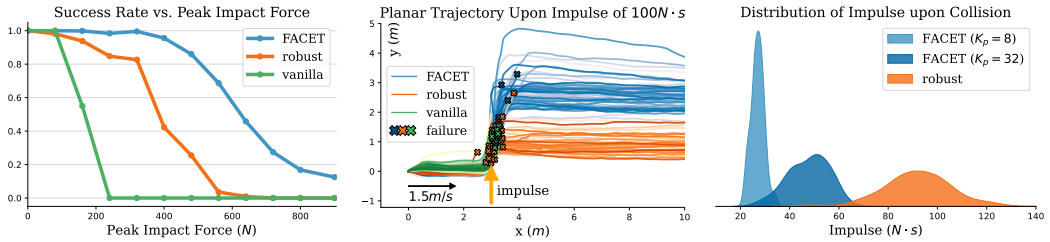

Figure 4: **Left**: The success rate under varying levels of lateral impulses during locomotion at 1.5 m/s in the x-direction. FACET demonstrates superior robustness to large impulses compared to baseline policies trained with velocity tracking (vanilla) and with random impulse perturbations (robust). **Middle**: Planar (xy) trajectories of robot CoM under $400N$ peak force. 64 trajectories are shown for each policy. FACET can compliantly follow the impulse, adapting its velocity to keep balance, unlike the stiffer responses or failures of the baselines. **Right**: Distribution of collision impulse when the robot walks into a virtual wall. FACET achieves a significantly lower collision impulse, and this can be modulated by adjusting the impedance parameter K_p, indicating enhanced safety during physical interaction.

Robustness to large perturbations. Our method introduces controllable compliance into the policy, allowing it to follow instead of counteract external forces when the forces are too large. To show that, we command the robot to walk forward along the x-axis at $1.5m/s$ and add a horizontal ramp impulse of varying magnitudes. We compare our method (FACET) with two baselines: (1) vanilla: a locomotion policy trained with velocity tracking reward, and (2) robust trained additionally with random impulse domain randomizations. The distribution of perturbations applied during training for FACET and robust are identical (random impulse lasting $0.4 \sim 0.6s$, peaking $80 \sim 200N$). Failure is defined as bodies other than feet colliding with the ground.

As shown in Fig. 4 (Left), vanilla quickly fails as the impulse gets larger. While robust policy shows that random perturbation during training enhances robustness, it is only effective within a medium range. In contrast, FACET can survive even larger impulses. This is depicted in Fig. 4 (Middle), which shows the trajectory at the peak impact force of $400N$. While vanilla and robust either fail or resist stiffly, FACET adjusts its velocity to move along the y-axis to restore balance.

Soft collision. Safety is a critical aspect when the robot operates in cluttered environments, where intense collisions can damage both the subject and the robot itself. Our approach enables more compliant behavior upon collision by using small K_p and K_d. We validate this property in simulation

by letting the robot walk into a wall (implemented using a soft collision model) at $1.5m/s$ and measure the impulse (change of momentum) within $0.5s$ after collision. As shown in Fig. 4 (Right), the impulse induced by FACET is significantly smaller than robust and can be controlled by tuning K_p.

Extension to multiple bodies: legged loco-manipulator. On more complex robots, we may be interested in more than the CoM of the trunk body. Our formulation can readily extend to such cases. For example, for a legged loco-manipulator (Unitree B1 quadruped + Z1 Arm), we can define reference models for both the base link and the end-effector:

$$m^{\text{base}}\ddot{\mathbf{x}}_{\text{ref}}^{\text{base}} = f_{\text{spring}}^{\text{base}} + f_{\text{ext}}^{\text{base}} - a \cdot f_{\text{spring}}^{\text{eef}}, \tag{8}$$

$$m^{\text{eef}}\ddot{\mathbf{x}}_{\text{ref}}^{\text{eef}} = f_{\text{spring}}^{\text{eef}} + f_{\text{ext}}^{\text{eef}}, \tag{9}$$

where $a \in [0, 1]$ controls whether the force applied on the end effector would transmit to the base. The reward functions are defined analogously to Eq. (7) for end-effector and base, respectively. Setting $a = 1$ and $\mathbf{x}_{\text{ref}}^{\text{base}} = \mathbf{x}^{\text{base}}$ gives the fully compliant mode. This convenient property allows us to perform kinesthetic teaching by pulling the end-effector to move the robot as shown in Fig. 1 (c). On the other hand, we can whole-body large force pulling by specifying high values for both K_p^{base} and K_p^{eef}. This is illustrated in Fig. 5, where we vary the value of K_p^{base} to gradually increase and then decrease the pulling force. The pulling force produced roughly matches the force specified in $f_{\text{spring}}^{\text{eef}}$ and $f_{\text{spring}}^{\text{base}}$, showing the desirable property of fine-grained force control.

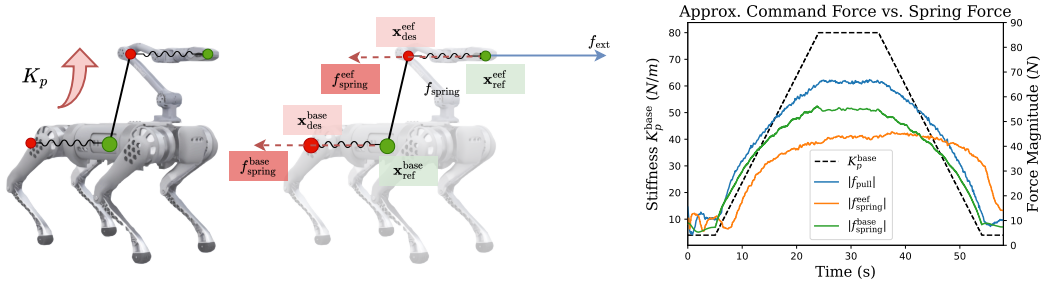

Figure 5: Extension to legged loco-manipulator with two bodies of interest. With a fixed $K_p^{\text{eef}} = 90$, we increase and then decrease K_p^{base} to examine the relationship between the virtual spring forces and the actual pulling forces produced f_{pull}. Similar to [8], f_{pull} is approximated by a force applied at the end-effector, which counteracts the robot so that it has near-zero velocity.

6 Ablation Study

We evaluate two key questions: **Q1:** Must we use integrated position and velocity trajectories rather than raw acceleration as tracking targets? **Q2:** Does temporal smoothing of the reference trajectories yield more stable performance compared to open-loop or closed-loop targets? To answer these, we compare four *teachers*—`smoothed`, `open-loop`, `closed-loop`, and `acceleration` targets—and three *student* policies derived from the `smoothed` and `open-loop` teachers, plus a `concurrent` baseline. Performance is measured by tracking errors and success rate. See Sec. 10.3 for detailed definition.

For the *teacher* policies, the large position and velocity errors of `Open-loop Teacher` and `Acceleration Teacher` indicate they fail to make meaningful progress throughout, while `Closed-loop Teacher` achieves a similar result as `Smoothed Teacher`, indicating that integrated position/velocity trajectories are essential for stable reference tracking. The `Open-loop Teacher` achieves even slightly better performance than the `Smoothed Teacher`, potentially because the open-loop reference trajectory is smoother and less noisy. However, it does not compensate for the gap between the robot dynamics and the reference model and is therefore too perfect for the student to learn from. This is evidenced by the higher errors by `Open-loop Student` compared to `Smoothed Student` after the second stage.

Table 2: Success Rate and Normalized Tracking Errors

Method	Normalized Errors ↓			Success ↑
	pos	vel	acc	
Smoothed Teacher	1.00	1.00	1.00	0.98
Open-loop Teacher	0.72	0.92	0.97	0.97
Closed-loop Teacher	17.71	2.04	1.14	0.96
Acceleration Teacher	70.51	6.64	1.65	0.97
Smoothed Student	5.18	1.85	1.05	0.97
Open-loop Student	10.31	2.07	1.04	0.98
Concurrent	26.23	3.53	1.19	0.92

Acceleration errors remain comparatively large and similar across all variants, confirming that raw `Acceleration` is not suited to be used as a reward signal. The smoothed targets allow more imperfection and lead to a smaller imitation gap while still adhering to the reference model.

Meanwhile, the one-stage explicit estimation baseline `Concurrent` [36], which directly estimates the external forces and reference targets, also leads to much worse performance. Estimating such information in the input space is challenging, and inaccurate estimation would in turn harm training.

7 Real-World Results

In this section, we deploy the student policy to a real Unitree Go2 quadruped for further validation of the force-adaptive capabilities offered by our approach.

Compliant following. To demonstrate our policy is compliant and can adapt to very small forces, we set impedance gains $K_p = 0, K_d = 4$, and apply small external forces to guide the robot's motion. As shown in Fig. 1 (a), the robot follows effortlessly under gentle pulling with a string attached to its body, and can start or stop moving with the push of a fingertip. Despite the absence of explicit velocity commands, the robot smoothly follows the direction of the applied force. This behavior is difficult to achieve with velocity-tracking policies or the built-in controller, which resist deviations from velocity command.

Large force pulling. To evaluate the force-generating capability and robustness of our controller, we set the impedance gains K_p to a large value. In this setup, the quadruped robot is tethered to a payload via a pulley in gym and tasked with pulling forward/backward. We compare FACET with two baselines: (1) a velocity-tracking policy (robust), (2) the built-in controller (builtin). A video demonstration of the pulling task is provided in the supplementary material.

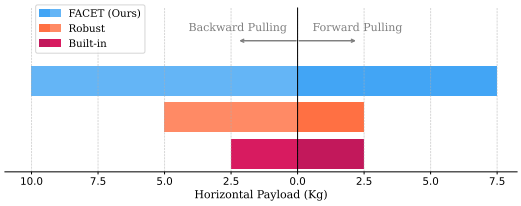

Figure 6: **Large Force Pulling**. Our controller achieves a significantly wider pulling range than the baselines and demonstrates more robust whole-body coordination behaviors.

The built-in controller successfully pulls up to 2.5kg, while robust fails beyond 5 kg, frequently collapsing or spinning in place due to instability. In contrast, our FACET policy reliably pulls up to 10kg without failure, maintaining stable contact and body orientation.

8 Conclusion

In this paper, we introduced FACET, a framework for training force-aware controllers for legged robots, and validated its efficacy through experiments. It addresses the issue found in previous learning-based controllers, where responses to forces are often unspecified or cannot be modulated, which we believe to be essential for robots to safely operate in complex environments with human presence.

9 Limitations

Our method has two main limitations: (1) the gap between the robot dynamics and the reference model is not fully addressed, especially when applied in the multi-body setting; and (2) the sim-to-real gap caused by the lack of reliable force sensing and acceleration measurement. Regarding (1), the mechanical constraints between multiple bodies are currently not considered. For example, the reference state targets of the end-effector of a loco-manipulator's end-effector can be infeasible to reach. Regarding (2), better state estimation methods could be incorporated to make full use of the sensors available on a robot, e.g., accelerometer and foot force sensors. Meanwhile, we used oscillators with predefined frequencies for gait control. While providing a clear reward signal for learning stable and nice-looking gaits, it may limit the robot's ability to adapt its foothold based on the target acceleration and velocity. We will move on to resolve these issues in future work.

Acknowledgments

We sincerely thank Chaoyi Pan, Chenhao Lu, Guanqi He, Sicheng He and Zhengmao He (alphabetical order) for their valuable insights and suggestions that made this work more complete.
This work is supported by the National Natural Science Foundation of China (62176135), the National Key R&D Program of China (2022ZD0161700), Shanghai Qi Zhi Institute Innovation Program SQZ202306 and the Tsinghua University Dushi Program, the grant of National Natural Science Foundation of China (NSFC) 12201341.

References

[1] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47), Oct. 2020. ISSN 2470-9476. doi:10.1126/scirobotics.abc5986. URL http://dx.doi.org/10.1126/scirobotics.abc5986.

[2] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023.

[3] Z. Zhuang, Z. Fu, J. Wang, C. G. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. In *Conference on Robot Learning CoRL*, 2023.

[4] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450. IEEE, 2024.

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

[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.

[7] 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.

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

[9] 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.

[10] Z. Wang, Y. Jia, L. Shi, H. Wang, H. Zhao, X. Li, J. Zhou, J. Ma, and G. Zhou. Arm-constrained curriculum learning for loco-manipulation of the wheel-legged robot. *arXiv preprint arXiv:2403.16535*, 2024.

[11] N. Fey, G. B. Margolis, M. Peticco, and P. Agrawal. Bridging the sim-to-real gap for athletic loco-manipulation. *arXiv preprint arXiv:2502.10894*, 2025.

[12] A. Hartmann, D. Kang, F. Zargarbashi, M. Zamora, and S. Coros. Deep compliant control for legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11421–11427. IEEE, 2024.

[13] N. Hogan. Impedance control: An approach to manipulation: Part i—theory. *Journal of Dynamic Systems Measurement and Control-transactions of The Asme*, 107:1–7, 1985.

[14] 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.

[15] F. Shi, C. Zhang, T. Miki, J. Lee, M. Hutter, and S. Coros. Rethinking robustness assessment: Adversarial attacks on learning-based quadrupedal locomotion controllers. *ArXiv*, abs/2405.12424, 2024. URL https://api.semanticscholar.org/CorpusID:269929916.

[16] Z. Xiao, X. Zhang, X. Zhou, and Q. Zhang. Pa-loco: Learning perturbation-adaptive locomotion for quadruped robots. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9110–9115, 2024. URL https://api.semanticscholar.org/CorpusID:271039823.

[17] W. Xiao, S. Lyu, Z. Gong, R. Wang, and D. Wang. Learning robotic policy with imagined transition: Mitigating the trade-off between robustness and optimality. *arXiv preprint arXiv:2503.10484*, 2025.

[18] Y. Wang, M. Xu, G. Shi, and D. Zhao. Guardians as you fall: Active mode transition for safe falling. In *2024 IEEE International Automated Vehicle Validation Conference (IAVVC)*, pages 1–8. IEEE, 2024.

[19] J. Shi, C. Bai, H. He, L. Han, D. Wang, B. Zhao, M. Zhao, X. Li, and X. Li. Robust quadrupedal locomotion via risk-averse policy learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11459–11466. IEEE, 2024.

[20] T. Portela, A. Cramariuc, M. Mittal, and M. Hutter. Whole-body end-effector pose tracking, 2025. URL https://arxiv.org/abs/2409.16048.

[21] 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.

[22] R. Martín-Martín, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1010–1017. IEEE, 2019.

[23] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, T. Nishi, S. Kikuchi, T. Matsubara, and K. Harada. Learning force control for contact-rich manipulation tasks with rigid position-controlled robots. *IEEE Robotics and Automation Letters*, 5(4):5709–5716, 2020.

[24] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal. Learning force control policies for compliant manipulation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4639–4644. IEEE, 2011.

[25] C. Chen, Z. Yu, H. Choi, M. Cutkosky, and J. Bohg. Dexforce: Extracting force-informed actions from kinesthetic demonstrations for dexterous manipulation. *arXiv preprint arXiv:2501.10356*, 2025.

[26] S. Chen, B. Zhang, M. W. Mueller, A. Rai, and K. Sreenath. Learning torque control for quadrupedal locomotion. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.

[27] S. Sood, G. Sun, P. Li, and G. Sartoretti. Decap: Decaying action priors for accelerated imitation learning of torque-based legged locomotion policies. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2809–2815. IEEE, 2024.

[28] D. Kim, G. Berseth, M. Schwartz, and J. Park. Torque-based deep reinforcement learning for task-and-robot agnostic learning on bipedal robots using sim-to-real transfer. *IEEE Robotics and Automation Letters*, 8(10):6251–6258, 2023.

[29] S. Lee, P. S. Chang, and J. Lee. Deep compliant control. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–9, 2022.

[30] M. Liu, Z. Chen, X. Cheng, Y. Ji, R. Yang, and X. Wang. Visual whole-body control for legged loco-manipulation. *arXiv preprint arXiv:2403.16967*, 2024.

[31] 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.

[32] R. Tedrake. *Robotic Manipulation*. 2024. URL http://manipulation.mit.edu.

[33] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021.

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

[35] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi:10.1109/LRA.2023.3270034.

[36] 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*, 7(2):4630–4637, 2022.

[37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

10 Appendix

10.1 Implementation and Training Details

10.1.1 Training Setup

The training environment and the task setup are implemented based on Isaac Lab [35]. We use PPO [34] to train the policies with 4096 parallel environments. Following the setup shown in Fig. 3, the critic V and encoder $\mathcal{E}$ are Multi-Layer Perceptron (MLP) with $(512, 256, 256)$ and 256 hidden units, respectively. The estimator $\hat{\mathcal{E}}$ is a GRU [37] with hidden dimension 256, while the policy π is parameterized by an MLP with $(256, 256, 256)$ hidden units outputting the mean of a diagonal Gaussian distribution with state-independent variance. Empirical normalization with running statistics is applied to the network inputs to improve optimization. The two-stage training runs for 200 million environment steps (first stage) and 50 million steps (second stage), requiring approximately 1 hour on a gaming laptop with Intel Core i9-13980HX CPU and NVIDIA RTX 4080 GPU. More details, such as hyper-parameters, can be found in the codebase.

10.1.2 Observation Space Details

We adopt a teacher-student observation structure where the student has access to a limited set of observations suitable for real-world deployment, while the teacher is granted access to privileged information during simulation training. All Cartesian-space observations, including position, and velocity, and wrench, are converted to the base's frame.

Table 3: List of Observations

Observation	Notation	**Available To**	**Description**
Command	$\mathbf{u}$	Student & Teacher	Includes $\mathbf{x}_{des}$, K_p, K_d, and virtual mass $\mathbf{m}$
Joint Positions	$\mathbf{q}$	Student & Teacher	Raw joint angles
Joint Velocities	$\dot{\mathbf{q}}$	Student & Teacher	Angular velocities of joints
Base Angular Velocity	ω	Student & Teacher	Angular velocity of the base (from IMU)
Projected Gravity Vector	g	Student & Teacher	Gravity vector projected into base frame
Base Linear Velocity	$\mathbf{x}$	Teacher Only	Linear velocity of the robot's base
Feet Contact States	$\mathbf{c}$	Teacher Only	Binary indicators of foot-ground contact
External Forces/Torques	f_{ext}	Teacher Only	Forces and torques applied to the base
Joint torques	$\boldsymbol{\tau}$	Teacher Only	Torques Commanded/applied at each joint
Reference Targets	$\{\mathbf{x}_{\text{ref}}, \dot{\mathbf{x}}_{\text{ref}}\}$	Teacher Only	Reference targets described in Sec. 4.2

10.1.3 Rewards Functions

The main task reward consists of position and velocity tracking of xy translation and yaw rotation with the form and tracking targets described in Eq. (7). The loco-manipulation experiment additionally includes position and tracking rewards defined for the end-effector. In practice, we found tracking rewards of the form $\exp(-||\dot{\mathbf{x}} - \dot{\mathbf{x}}_{\text{ref}}||_2^2)$ may saturate when the error is large during the early stage. A quadratic cost term $-||\dot{\mathbf{x}} - \dot{\mathbf{x}}_{\text{ref}}||_2^2$ is added to resolve this issue. A series of regularization rewards is used to ensure natural behaviors. A full list of rewards for the Unitree Go2 quadruped task is summarized in Sec. 10.1.3.

10.1.4 Command and Force Sampling

Command. Following Eq. (2), we sample impedance parameters $\mathbf{u} = (\mathbf{x}_{des}, K_p, K_d, \mathbf{m})$ and external forces $\mathbf{f}_{\text{ext}}$ that induce force-adaptive behavior in forceful interactions. To ensure natural motion and avoid oscillation around equilibrium, we always use the critical damping $K_d = 2\sqrt{K_p}$. The virtual mass is sampled from $\{1, 2, 4\}kg$. For the quadruped task, the setpoint $\mathbf{x}_{\text{des}}$ is periodically sampled and updated in one of the following ways:

Impedance Tracking Rewards		
Reward Name	Weight	Function
impedance_yaw_pos	0.75	$\exp(-\|x_{\text{ref}}^{\text{yaw}} - x_{\text{sim}}^{\text{yaw}}\|^2/0.25)$
impedance_yaw_vel	1.0	$\exp(-\|\dot{x}_{\text{ref}}^{\text{yaw}} - \dot{x}_{\text{sim}}^{\text{yaw}}\|^2/0.25) - 0.5\|\dot{x}_{\text{ref}}^{\text{yaw}} - \dot{x}_{\text{sim}}^{\text{yaw}}\|^2$
impedance_pos	0.75	$\exp(-\|\mathbf{x}_{\text{ref}}^{xy} - \mathbf{x}_{\text{sim}}^{xy}\|^2/0.25)$
impedance_vel	1.5	$\exp(-\|\mathbf{v}_{\text{ref}}^{xy} - \mathbf{v}_{\text{sim}}^{xy}\|^2/0.25) - 0.5\|\mathbf{v}_{\text{ref}}^{xy} - \mathbf{v}_{\text{sim}}^{xy}\|^2$
Regularization Rewards		
angvel_xy_l2	0.02	$-\|\boldsymbol{\omega}_{\text{sim}}^{xy}\|^2$
linvel_z_l2	2.0	$-v_{z,\text{sim}}^2$
base_height_l1	2.0	$-\|z_{\text{sim}} - 0.35\|$
energy_l1	0.0002	$-\|\boldsymbol{\tau} \odot \dot{\mathbf{q}}\|_1$
joint_acc_l2	2.5×10^{-7}	$-\|\ddot{\mathbf{q}}_{\text{sim}}\|^2$
joint_torques_l2	2.0×10^{-4}	$-\|\boldsymbol{\tau}_{\text{sim}}\|^2$
survival	1.0	1
action_rate_l2	0.01	$-\|\dot{\mathbf{a}}\|^2$
feet_air_time	0.4	$\sum_{\text{num_feet}} \max(0, t_{\text{air}} - 0.3)$
undesired_contact	2.0	$-\sum_{\text{undesired_bodies}} I(\text{contact})$

1. **Fixed Target Position**: A new setpoint is sampled around the current base position $\mathbf{x}_{\text{sim}}$ in the world frame and stays fixed for a few seconds before the next one is sampled.
2. **Fixed Target Velocity**: A target velocity $\dot{\mathbf{x}}^{\text{target}}$ is sampled and the setpoint is updated at each step as $\mathbf{x}_{\text{des}} = \frac{K_d}{K_p}\dot{\mathbf{x}}^{\text{target}} + \mathbf{x}_{\text{sim}}$ so that the velocity converges to $\dot{\mathbf{x}}^{\text{target}}$.
3. **Fully Compliant**: The setpoint moves with $\mathbf{x}_{\text{sim}}$, i.e., we set $\mathbf{x}_{\text{ref}} := \mathbf{x}_{\text{sim}}$ at each step. The virtual spring force f_{ext} stays zero so that the robot moves fully compliant to external forces.

For the loco-manipulation task, we sample the end effector setpoint $\mathbf{x}_{\text{des}}^{(1)}$ in the body frame of the quadruped base. To ensure the setpoint is feasible to reach (subject to some error threshold), an archive of such poses is obtained in advance using Inverse Kinematics. In this work, the end-effector's desired orientation is always set as forward in the base's frame.

External forces. To simulate forceful interactions with the environment, three types of forces are randomly sampled and applied:

1. **Constant**: A steady force is sampled from a predefined range and applied continuously for a $1 \sim 4s$ time. For the quadruped task, the maximum force along x, y and z axis are $40N$, $40N$ and $15N$, respectively.
2. **Impulse**: A high-magnitude force applied for a short duration $0.4 \sim 0.6s$. We adopt a ramp impulse that linearly increases to the peak and then decreases to zero. The resulting impulse is therefore $0.5 \cdot \text{duration} \cdot f_{peak}$. For the quadruped task, the peak forces along the x, y, and z axes are sampled from $80 \sim 200N$, $80 \sim 200N$, and $0 \sim 20N$, respectively.
3. **Spring**: A force that behaves as if the point of interest is attached to a highly stiff spring similar to that described in [8].

We also introduce small positional offsets relative to the center of mass so that these forces produce torques as well.

10.2 Simulation Experiments

Robustness to large perturbations. We command the robot to walk forward along the x-axis at $\dot{\mathbf{x}}^{\text{target}} = 1.5m/s$ by setting the setpoint in front of the robot according to $\mathbf{x}_{\text{des}} = \frac{K_d}{K_p}\dot{\mathbf{x}}^{\text{target}} + \mathbf{x}_{\text{sim}}$. We add the horizontal ramp impulse at $t = 2.0s$, i.e., after the locomotion is stable. The impulse lasts for $0.4 \sim 0.6s$ and has varying magnitudes. For each policy, we evaluate the policy in 64 parallel environments with randomizations. Failure is defined as bodies other than feet colliding

with the ground. The robust baseline is trained with the same random impulse described above as our method.

Soft collision. We implement the virtual wall with a soft collision model, i.e., the robot will experience an external force when colliding into the wall:

$$f_{\text{wall}} = \begin{cases} -K_p^{\text{wall}} d_{\text{penetration}} - K_d^{\text{wall}} \dot{d}_{\text{penetration}}, d_{\text{penatration}} > 0 \\ 0, d_{\text{penetration}} \leq 0 \end{cases} \tag{10}$$

where $d_{\text{penetration}}$ is the distance of penetration normal to the wall plane direction, measured at the robot's base. We measure the impulse by integrating the wall contact force f_{wall} within $0.5s$ after collision takes place.

10.3 Ablation Study

We random sample the impedance parameters $\mathbf{x}_{\text{des}}, K_p, K_d$ and external force f_{ext} following the strategy described in Sec. 10.1.4.

For all policies trained with different tracking targets, we measure tracking error with a unified metric: the position and velocity tracking error is computed with respect to the reference targets $\mathbf{x}_{\text{tef}}^{t'}, \dot{\mathbf{x}}_{\text{tef}}^{t'}$ for $t' \in \{t - 8\Delta t, t - 16\Delta t, t - 32\Delta t, 0\}$ and the acceleration error is computed against the closed-loop target $\ddot{\mathbf{x}}_{\text{ref}}^{t-1}$:

$$\text{pos error} = \sum_{t'} ||\mathbf{x}_{\text{ref}}^{t'} - \mathbf{x}||_2^2, \tag{11}$$

$$\text{vel error} = \sum_{t'} ||\dot{\mathbf{x}}_{\text{ref}}^{t'} - \dot{\mathbf{x}}||_2^2, \tag{12}$$

$$\text{acc error} = ||\ddot{\mathbf{x}}_{\text{ref}} - \ddot{\mathbf{x}}||_2^2. \tag{13}$$

Such metrics are chosen because we want the policy to resample the reference model as close as possible, while accurately following the open-loop targets $(\mathbf{x}_{\text{ref}}^0, \dot{\mathbf{x}}_{\text{ref}}^0)$ is infeasible and also unnecessary. Similar to reward computation, we use mixtures of targets for the position and velocity errors. As for acceleration, we directly compute the error w.r.t the reference at that step since we care more about its instantaneous behavior.

10.4 Real-World Experiments

In most cases, we do not have access to a well-defined world frame on the hardware. When deployed to a real robot, we can only specify the translation part of the setpoint $\mathbf{x}_{\text{des}}$ directly in the robot's frame following the **fixed target velocity** mode or **fully compliant mode** described in Sec. 10.1.4. The yaw rotation part of $\mathbf{x}_{\text{des}}$ can be specified in the rotation frame created when the robot starts up.

Compliant following. The compliant following behavior can be produced with the **fully compliant mode** with any valid K_d within the training range. A smaller K_d makes it easier to drive the robot.

Large force pulling. In the large force pulling experiment, we gradually increase K_p and setpoint displacement so that $K_p(\mathbf{x}_{\text{ref}} - \mathbf{x})$ increases to $100N$. Videos and further results can be found in the supplementary materials.

10.5 Rebuttal Questions and Answers

To let the reader have a full view of the paper, we include some of the rebuttal content here. **The live demo is updated to include the added baselines.** Note that there is slight sim2sim gaps to run Mujoco in the browser.

Choice of impedance parameters The impedance parameters (K_p, K_d, m) are chosen to be intuitive and physically grounded. To ensure the spring force is reasonable within the robot's physical limits, we sample the stiffness K_p from a range guided by constraints like $K_p \times 2\,\mathrm{m} \leq 15\,\mathrm{kg} \times 9.8 \times 0.5$, where we assume a maximum displacement of 2m and a rough friction coefficient of 0.5. The damping K_d is consistently set near critical damping to ensure smooth, non-oscillatory motion for ease of operation. The virtual mass m does not have to align with the real mass and usually is lighter than the real mass: This is because the robot's body acts as a 'floating base' supported by its legs, an inherently unstable configuration. Consequently, its effective inertia is lower than its physical mass, as the policy can react to forces by taking a step to re-balance rather than through pure inertial resistance. Thus we use smaller values $\{1, 2, 4\}$ kg for more compliant behavior. All of the above are sampled multiple times in one training episode. During deployment, these parameters can be set and **changed on-the-fly as policy inputs** by a human operator or, for advanced applications, by a high-level policy, which is outside this paper's scope.

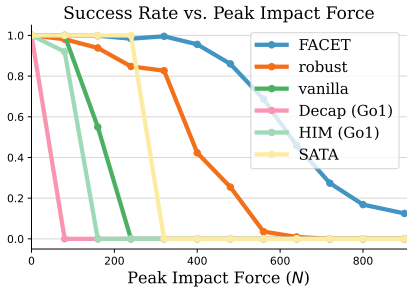

Figure 7: **Left:** Additional comparison with baselines Decap [27], HIMLoco[1], and SATA[2]. We remark that robust is a strong baseline.

Comparison with other works that introduce compliance We differentiate FACET from methods that tune PD gains or use torque commands by highlighting a key conceptual difference: joint-space versus task-space compliance. While those methods make individual joints "soft," FACET directly controls compliance at the robot's center-of-mass (CoM). These two are decoupled; a robot can have stiff joints for precise foot placement yet exhibit compliant CoM behavior by taking a step to yield to a push. FACET is therefore complementary to joint-level methods and could be combined with them for enhanced performance.

Meanwhile, joint-level methods usually require additional effort for effective training and deployment, e.g., higher control frequency [26], action priors [27], or more complex actuation models[2].

Relation with other works on training recipes We would like to clarify that the main contribution of FACET is the formulation, which is **orthogonal and complementary** to previous work on training recipes, such as RMA, DreamWaQ, and HIMLoco. These methods could have replaced the teacher-student training recipe used in this work to achieve similar results. Policies trained by them are indeed robust, but this is mainly attributed to the external force perturbations applied during training. We remark that robustis a strong baseline with comparable and even better performance than HIMLoco, as shown in Fig. 7 and the live demo.

In addition, the robustness improvements are in different dimensions. Previous works learn to resist or reject external forces when possible, while FACET policy learns to optionally follow external forces. Therefore, while the compliance of other policies is an **emergent and often unpredictable** property, FACET provides a **principled and specifiable** way to modulate force-adaptive behavior on demand.

Regular locomotion/walking performance Policies trained by FACET have comparable locomotion capabilities on flat and planar rough terrains, since under the hood, it also learns to track a wide range of velocities under different external force patterns. On more complex discrete terrains, e.g., stairs, additional design is needed since the desired behavior is nontrivial. For example, in what way do we want compliance when going upstairs, if at all? We acknowledge that it is currently a limitation of our work. Meanwhile, we didn't aim for very pretty gaits, which is not relevant to our contribution.

[1]Long, Junfeng, et al. "Hybrid internal model: Learning agile legged locomotion with simulated robot response." arXiv preprint arXiv:2312.11460 (2023).

[1]Li, Peizhuo, et al. "SATA: Safe and Adaptive Torque-Based Locomotion Policies Inspired by Animal Learning." arXiv preprint arXiv:2502.12674 (2025).