2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Active Dynamic Load Adaptation for Quadruped Locomotion on Complex Terrain

Yimin Xiao[1,*], Dianzhong Li[1,*], Wangjun Huang[1], Ying Sha[1,2,3,4,†] and Li Qin[1,2,3,4,†]

***Abstract*— Quadruped robots show important potential for load-carrying tasks due to their terrain adaptability, and a unique challenge of these tasks is to maintain quadrupedal stability when the load has active and dynamic characteristics. The load's mass and center of mass change dynamically, rather than being integrated as a whole-body component of the quadruped. Unlike traditional load-carrying tasks, where the load is typically passive and its influence on the robot's movement is predictable and static, active dynamic loads can actively alter the robot's balance control in real-time, imposing load disturbances on the robot's locomotion. These load disturbances, when combined with the fundamental attitude changes induced by complex terrain, introduce dual dynamic disturbances to the robot. To address these dual disturbances, we propose an active dynamic load modeling approach that captures the active and dynamic characteristics of the load, enabling the robot to adapt to the real-time changes in load movement. This approach is integrated into a Reinforcement Learning (RL) framework that leverages dynamic models: an Inverse Dynamic Model (IDM) which learns the dynamic characteristics of the active load, and a Forward Dynamic Model (FDM) which predicts the effects of complex terrain on the robot's motion, enabling synchronous adaptation to both types of dynamic disturbances. Extensive comparative simulations and physical experiments across diverse terrains, with active dynamic load of varying movements, demonstrate the effectiveness of our method in enhancing balance control and adaptability. More details are available at: Project Page.**

I. INTRODUCTION

Thanks to the adaptability of quadruped robots to complex terrain [1], [2], they show potential for transportation and operational tasks, and are applied in domains like logistics [3], search and rescue [4], and agriculture [5]. Load-carrying is a core requirement in these tasks, but introducing a load changes the system's dynamics, posing challenges for the robot's balance control. Especially when the load has active capabilities and dynamic characteristics, such as a robotic arm operating on a quadruped robot's back or other load systems with internal degrees of freedom. Distinct from passive dynamic loads, such as moving weights that merely slide or oscillate reactively due to the robot's locomotion, an active dynamic load not only changes the mass of system distribution and inertial properties over time, causing a continuous shift in the robot's center of mass, but its active movement further increases the complexity of balance control. Specifically, as the robotic arm adjusts its position and orientation during manipulation, this motion not only alters the load's mass distribution but also imposes actively varying torques on the robot, thereby disrupting its equilibrium and stability. More importantly, when a quadruped robot moves over complex terrain, the base posture disturbance caused by the terrain and the load disturbance can be superimposed, creating terrain–load dual disturbances. This dual disturbance poses a significant challenge to maintaining balance and stability during movement, making adaptation to active dynamic loads over various terrains a critical area for further exploration.

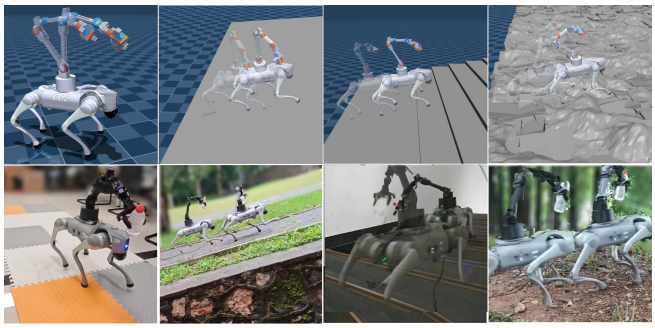


Fig. 1: Our testing setup with the Unitree Go2 and D1 Mechanical Arm as the active dynamic load, featuring complex terrain elements such as flat, stair, rough and slope, and load movement such as front-back, left-right, rotation and random. The first and second columns of photos show the tests of our proposed method in the simulation and real-world deployment.

Recent studies have explored load-carrying tasks. The adaptive-control strategy method handled changes in load mass [6], assuming that the load is static or quasi-static, and ignores the interference caused by the active movement of the load. The explicit load modeling method [7], which is based on parameters such as mass, position, and velocity, characterizes the load while excluding scenarios where the load movement is not defined. The disturbance prediction control method is designed for a known dynamic load of robotic arms and relies on the robotic arm model and available

This research project was supported by the Fundamental Research Funds for the Chinese Central Universities (Grant No. 2662025PY020 and 2662025-XXPY006).

[1]College of Informatics, Huazhong Agricultural University, No.1 Shizishan Street, Hongshan District, Wuhan, 430070, Hubei, China.

[2]Key Laboratory of Smart Farming for Agricultural Animals, No.1 Shizishan Street, Hongshan District, Wuhan, 430070, Hubei, China.

[3]Engineering Research Center of Intelligent Technology for Agriculture, Ministry of Education, No.1 Shizishan Street, Hongshan District, Wuhan, 430070, Hubei, China.

[4]Hubei Engineering Technology Research Center of Agricultural Big-Data, No.1 Shizishan Street, Hongshan District, Wuhan, 430070, Hubei, China.

[*]The first two authors are equally contribution.

[†]Corresponding author.

information [8]. Some studies have implemented whole-body control learning methods for quadruped robots and robotic arms [9], but they rely on a tightly coupled system design, which is not suitable for scenarios with unknown dynamic loads. Learning-based dynamic models have proven effective in predicting the state evolution and external disturbances of robots on complex terrain [10], [11], [12]. All the above-mentioned methods have not fully explored how to address the dual disturbances challenges of terrain-load.

To address this challenge, we propose an active dynamic load modeling method that utilizes an Inverse Dynamics Model (IDM) to learn the active and dynamic characteristics, enabling implicit modeling of unknown load disturbance. Building on this, we further propose a RL framework with dynamics models, where the Forward Dynamic Model (FDM) predicts terrain changes. Through the synergistic effect of dual dynamic models, it adapts to dual disturbances caused by load movement and complex terrain based only on proprioception. This dual dynamic collaborative architecture enables robots to enhance balance control and robustness in handling both terrain and load variations, as shown in Fig. 1. Our main contributions are summarized as follows:

- We propose a method for active dynamic load modeling, and extract applied forces generated by the active load movement using an inverse dynamics model.
- We establish a reinforcement learning framework utilizing dynamics models, enabling both adaptation to active dynamic load and complex terrain based only on proprioception through the synergistic effect of dual dynamics models.
- We conduct extensive comparative simulation experiments and physical experiments to validate the effectiveness of our proposed method and framework.

II. RELATED WORK

A. Reinforcement Learning for Robot Locomotion

Reinforcement learning (RL) has shown effectiveness in controlling quadruped locomotion on complex terrain, and a key component is robust state representation learning (SRL) [13]. Current research primarily focuses on three major approaches. Self-Supervised Learning aims to reconstruct raw information, RMA reconstructs environmental parameters to enable rapid adaptation [14], while Cheng et al. reconstruct estimated collision information [2]. Although effective for low-dimensional states, these methods may lose critical details when processing high-dimensional data. Generative Learning, using models like VAE [15] and Transformer [16], predicts future observations or states. DreamWaQ implicitly learns terrain features with a VAE model [17], and PIE integrates multi-source information with a Transformer for high-performance parkour [18]. However, these methods are susceptible to noise, especially from domain randomization. Contrastive Learning compares different samples to ensure states with similar features are close in the embedding space. HIMLoco uses this approach to acquire representations of environmental disturbance, enhancing the robustness of state estimation [19]. Similarly, SLR learns a high-performance policy using only proprioceptive information [10]. These methods are more effective at handling observational noise and are capable of efficient implicit state reconstruction. In this work, we learn a forward dynamic model to train an environment encoder that infers terrain disturbance with only proprioceptive observation.

B. Dynamics Modeling

Dynamic modeling including forward dynamics models (FDM) and inverse dynamics models (IDM) significantly enhances representation learning in reinforcement learning as auxiliary tasks [13], [20]. FDM assists agents to effectively capture the dynamic changes; this ensures the latent space encodes crucial transition information for predicting future states. For instance, McInroe et al. demonstrated that by predicting future states within the latent space using a multi-step history, agents can learn multi-level, temporally-aware state representations [21]. In contrast, IDM focuses on the controllable aspects of the environment, learning a latent representation that captures what caused a state transition. Pathak et al. utilized an IDM to learn state features by training a network to predict the executed action, which forces the features to retain only relevant information while filtering out environmental noise [22]. Building upon this, our work integrates both FDM and IDM to attend to terrain disturbance changes and dynamic load states.

C. Load Adaptation

In this work, we focus on adaptive control under active dynamic load, where the load actively moves and applies forces to the quadruped robot, and the robot can only rely on proprioception to achieve real-time coordination. Existing work generally abstracts the load as an added mass or an external disturbance. Kurva et al. proposed a two-stage adaptive-RL framework, which introduces an adaptive policy to compensate for unknown mass or center of mass variations [6]. Yao et al. presented a unified framework based on disturbance prediction control, where a high-level RL component is used to predict the disturbances from the robotic arm's load [8]. Furthermore, Chang et al. proposed a load characteristics modeling method for unknown dynamic loads [7], but this method can only capture the passive dynamic changes of the load and does not fully consider the active forces applied by the dynamic load.

III. METHOD

Our goal is to design an end-to-end RL-control framework that enables quadruped robots to perceive active dynamic load and complex terrain using only proprioceptive information.

A. Active Dynamic Load Modeling

The primary challenge for quadruped robots adapting to active dynamic loads is accurately capturing their active and dynamic characteristics. Unlike passive loads, which simply increase mass, active dynamic loads continuously affect robot

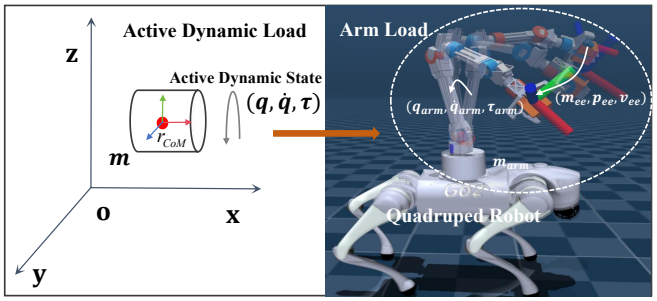


Fig. 2: On the left side is the definition of active dynamic load modeling, consists of mass m and center of mass r_{CoM}, the active state (joint position q, velocity $\dot{q}$, and forces τ). On the right side is a case of a arm load, modeled as the mass of the arm m_{arm}, joints position q_{arm}, velocity $\dot{q}_{arm}$, and forces τ_{arm}, and the mass m_{ee}, position p_{ee}, and velocity v_{ee} of the end effector.

stability through their own movements. The key difference is that the time-varying forces from the load's movements can disrupt balance. To address this, we propose a novel active dynamic load modeling method, characterizing the load using dynamic parameters: mass m, center of mass (CoM) r_{CoM}, and active dynamic state (defined by joint positions q, velocities $\dot{q}$, and efforts τ). These parameters directly influence system stability: the load's mass alters the total mass of system, its CoM affects the overall system CoM shift, and its active dynamic state determines the inertial forces and torques imparted to the system. A case in point is that when a quadruped robot is equipped with a robotic arm, the dynamic load is modeled based on the arm's mass m_{arm}, joint positions q_{arm}, velocities $\dot{q}_{arm}$, efforts τ_{arm}, and the mass of end-effector m_{ee}, position p_{ee}, and velocity v_{ee}, as depicted in Fig. 2.

B. Problem Formulation

We formulate the problem of the active dynamic load in quadruped robots on complex terrain as a Partially Observable Markov Decision Process (POMDP), denoted as $M = (S, A, O, R, P, \Omega, \gamma)$, where S, A and O are the set of states, actions, and observations respectively. At time t, the environment state is represented by $s_t \in S$, and the agent chooses an action $a_t \in A$, the transition probability $P(s_{t+1}|s_t, a_t)$ determines the next state s_{t+1}. The cooperative reward for this state and action is given by $r_t = R(s_t, a_t)$. However, the agent cannot availably obtain the complete state of the environment. Instead, it receives an observation $o_t \in \Omega$ from the observation transition probability $O(o_t|s_t, a_t)$. The agent's goal is to learn a policy π that maximises the expected total discounted reward $\mathbb{E}_\pi \sum_{t=0}^{\infty} \gamma^t r_t$, where $\gamma \in [0, 1)$ is a discount factor.

Observation and State Space: The robot proprioceptive observation $o_t \in \mathbb{R}^{45}$ includes body angular velocity, projected gravity, linear velocity commands, joint angles, joint angular velocities, and the last step action. The state $s_t = (o_t, p_t, l_t)$, including o_t, privileged information $p_t \in \mathbb{R}^{202}$ and load information $l_t \in \mathbb{R}^{25}$, where p_t includes base linear velocity, terrain height map and robot joint effort, l_t includes the mass of arm, joint positions, joint velocities, joint efforts, and the mass of end-effector, position and velocity.

Action Space: The action $a_t \in \mathbb{R}^{12}$ represents the expected 12-dimensional joint torque applied to the actuator relative to the initial posture of the joint angle.

Reward Function: The reward functions we use during the training are shown in Table I, which come from [19]. The terms designed specifically for dynamic load adaptation are load base stability and load balance. The former penalizes sway caused by robotic arm movement, while the latter encourages the robot to counteract load motion to achieve dynamic balance, $I^{load}_{\text{intensity}}$ represents the load motion intensity:

$$I^{load}_{\text{intensity}} = \dot{q}_{load} + 0.1 \cdot \ddot{q}_{load} + 0.01 \cdot \tau_{load}, \tag{1}$$

which includes active state joint velocities $\dot{q}_{load}$, joint acceleration $\ddot{q}_{load}$ and joint efforts τ_{load}. $S_{stability}$ represents the stable state of the robot:

$$S_{stability} = 0.1 \cdot v_z^2 + 0.1 \cdot \|\omega_{xy}\|^2 + \|g\|^2, \tag{2}$$

which includes linear velocity(z) v_z^2, angular velocity(xy) $\|\omega_{xy}\|^2$ and orientation $\|g\|^2$. Critically, as the intensity of the load movement increases, the penalty term scales proportionally, which compels the robot to proactively anticipate the disruptive effects of the active dynamic load.

TABLE I: Reward Terms

Reward Term	Expression	Weight
Linear velocity tracking	$\exp\left(-4\left\|v_{xy}^{\text{cmd}} - v_{xy}\right\|^2\right)$	2.0
Angular velocity tracking	$\exp\left(-4\left\|\omega_z^{\text{cmd}} - \omega_z\right\|^2\right)$	1.0
Linear velocity(z) penalty	v_z^2	-2.0
Angular velocity(xy) penalty	$\|\omega_{xy}\|^2$	-0.1
Base height	$(h^{\text{des}} - h)^2$	-2.0
Joint deviations	$\|q_i - q_{i,\text{default}}\|$	-0.1
Joint accelerations	$\|\ddot{q}\|^2$	-2.5e-7
Joint powers	$\|\dot{q}\|^T \tau$	-3e-3
Action rate	$\|a_t - a_{t-1}\|^2$	-0.01
Foot clearance	$\sum_{i=0}^{3}\left(p_z^{\text{target}} - p_z^i\right)^2 \cdot v_{xy}^i$	-0.01
Load Base Stability	$\left(0.1v_z^2 + \|\omega_{xy}\|^2\right) \cdot I^{load}_{\text{intensity}}$	-1.0
Load Balance	$\|\dot{q_{load}}\| \cdot \exp\left(-S_{stability}\right)$	1.0

C. Reinforcement Learning Framework

An overview of the proposed framework for active dynamic load adaptation is illustrated in Fig. 3. Our framework consists of three modules: an inverse dynamics module, a forward dynamics module, and a RL optimization module.

Forward Dynamics Module(FDM): This module simulates the real state transitions of the environment $p(s_{t+1}|s_t, a_t)$. It includes two networks, an env encoder compresses the current proprioceptive observation history

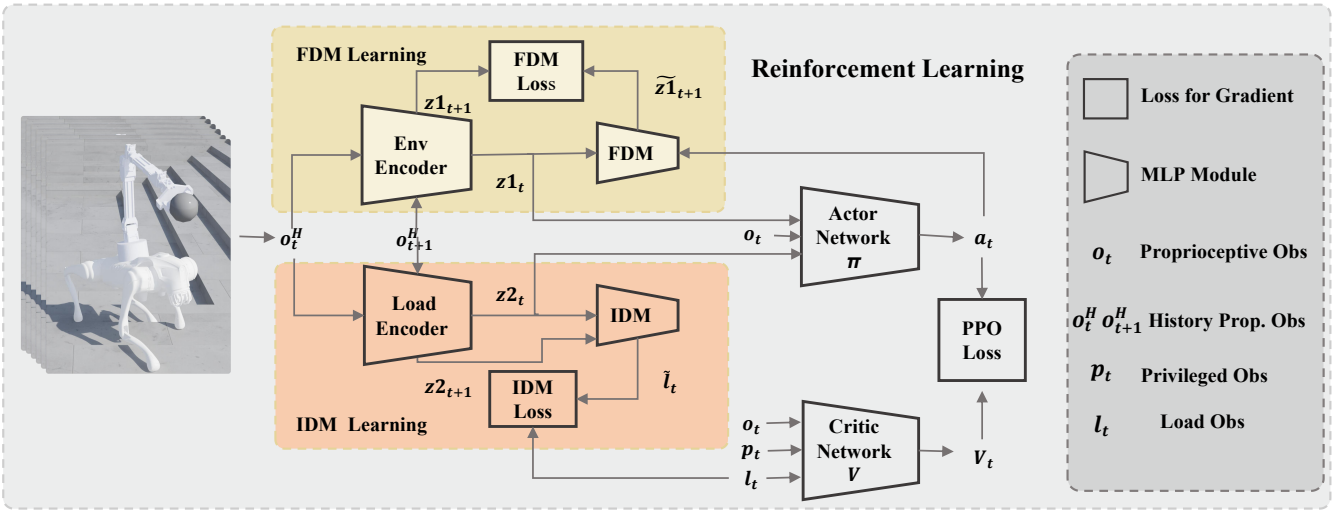


Fig. 3: Overview of the proposed framework for active dynamic load adaptation on the quadruped robots.

o_t^H and the next proprioceptive observation history o_{t+1}^H into the current latent vector $z1_t$ and the next latent vector $z1_{t+1}$, where H represents the length of history, the FDM is conditioned on $z1_t$ and the action a_t, and predicts the next latent vector $\tilde{z1}_{t+1}$:

$$f_{fdm}(\tilde{z1}_{t+1}|z1_t, a_t). \tag{3}$$

During training, FDM is optimized using a mean squared error (MSE) loss:

$$\mathscr{L}_{fdm} = \mathbb{E}\left[\left\|z1_{t+1} - \tilde{z1}_{t+1}\right\|^2\right]. \tag{4}$$

Inverse Dynamics Module (IDM): This module uses a load model and simulates arm load disturbance through IDM. It includes two networks, the arm load encoder compresses o_t^H and o_{t+1}^H into the current latent vector $z2_t$ and the next latent vector $z2_{t+1}$, the IDM is conditioned on $z2_t$ and $z2_{t+1}$, and predicts the estimated arm load information $\tilde{l}_t$ that causes the state transitions:

$$f_{idm}(\tilde{l}_t|z2_t, z2_{t+1}). \tag{5}$$

During training, IDM is optimized using a mean squared error (MSE) loss:

$$\mathscr{L}_{idm} = \mathbb{E}\left[\left\|l_t - \tilde{l}_t\right\|^2\right]. \tag{6}$$

Actor-Critic: The policy (Actor) and Critic networks are trained jointly via Proximal Policy Optimization (PPO) algorithm [23]. The policy takes the current proprioceptive observation o_t, $z1_t$ and $z2_t$ as input, and outputs a_t. The critic takes o_t, privileged information p_t, ground-truth arm load characteristics information l_t as input, and outputs the state value v_t:

$$a_t = \pi(o_t, z1_t, z2_t), \tag{7}$$

$$v_t = V(o_t, p_t, l_t). \tag{8}$$

The policy and critic networks are trained using the PPO objective:

$$\mathscr{L}_{ppo}^{\theta} = \mathbb{E}\left[min\left(r_t(\theta)\hat{A}_t, clip\left(r_t(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_t\right)\right], \tag{9}$$

TABLE II: Network Architectures

Module	Inputs	Hidden Layer	Output
Env encoder	o_t^H	[256,128]	$z1_t$
Load encoder	o_t^H	[256,128]	$z2_t$
FDM model	$z1_t, a_t$	[64]	$\tilde{z1}_{t+1}$
IDM model	$z2_t, z2_{t+1}$	[64]	$\tilde{l}_t$
Actor	$o_t, z1_t, z2_t$	[512, 256, 128]	a_t
Critic	o_t, p_t, l_t	[512, 256, 128]	V_t

where $r_t(\theta)$ is the probability ratio between the new and old policies, A_t is the corresponding advantage estimate and ε is the clipping threshold.

D. Details of Training Setting

1) Network Architecture: All the network modules illustrated in Fig. 3 are designed as the Multi-Layer Perceptron (MLP) with Exponential Linear Unit (ELU) activation. The details of the networks can be found in Table II.
2) Simulated Training Environment: We use the Isaac Lab [24] to build the simulation environment. The policy was trained using 4096 agents with a history size of $H = 6$ on a NVIDIA RTX 4090 GPU. The hyperparameters for the RL training and the dynamics module training are shown in Table III.
3) Curriculum Learning: We adopt curriculum learning [25] to facilitate progressive learning of dynamic load-carrying abilities across challenging terrain. Our training curriculum includes three parts: terrain curriculum, velocity curriculum, and load mass curriculum. The terrain curriculum includes stair range [0.05,0.23] m, a slope range [0°, 45°], flat and rough range [0.01, 0.06] noise. The velocity curriculum adjusts the maximum [-1,1] m/s and [-0.4,0.4] rad/s based on the speed

TABLE III: HYPERPARAMETERS

Module	Inputs
Optimizer	Adam
Gamma γ	0.99
Lambda λ	0.95
Desired KL-divergence	0.01
Learning rate	adaptive
Entropy coefficient	0.01
Clip range	0.2
Environments	4096
Obs history	6
Env enc. Dim	16
Load enc. Dim	16

TABLE IV: Domain Randomizations

Parameters	Range[Min,Max]	Unit
Body Mass	[0.8,1.2]×nominal value	Kg
Arm Link Mass	[-0.1,0.1]×nominal value	Kg
Arm EE Mass	[0.0,3.0]×nominal value	Kg
Arm Actions	[0.1,0.5]	s
Friction	[0.3, 1.2]	-
Initial Joint Positions	[0.5,1.5]×nominal value	rad
Initial Joint Velocity	[-1.0,1.0]×nominal value	rad/s

tracking reward from [-0.1, 0.1] m/s and [-0.1, 0.1] rad/s. The load mass curriculum adjusts the load mass [0,3] kg based on the reward.

4) Domain Randomizations: For the training of our method and to narrow the sim-to-real gap, domain randomization is used. The details are shown in Table IV, which come from[11]. We add noise to the load and simulate different unknown load. In addition, we design a randomized simulation of dynamic load with active effects, where the arm actions are stochastically generated by varying their frequency, amplitude, and phase.

IV. EXPERIMENTS

We use the Unitree Go2 robot to evaluate the performance of our proposed method in both simulated and real-world environments. To evaluate the effectiveness of our proposed method, we trained the algorithm and its ablation variants, and compared with the state-of-the-art model HIMLoco. All four models were trained with an asymmetric actor-critic architecture.

- Ours w/o FDM: Training without FDM.
- Ours w/o IDM: Training without IDM.
- BaseLine: Training directly with PPO.
- HIMLoco: HIMLoco only explicitly estimates velocity and implicitly simulates the system response as an implicit latent embedding by contrastive learning.

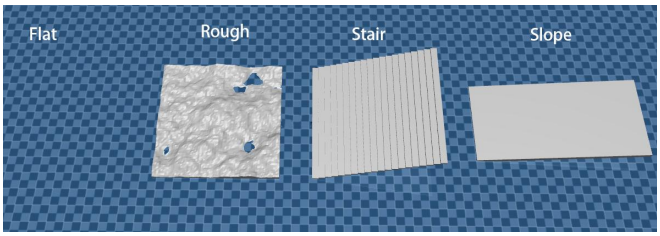


Fig. 4: Four scenarios: flat, stair, rough terrain, and slope.

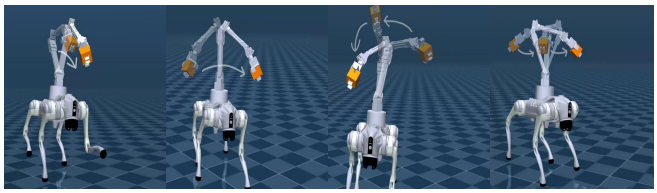
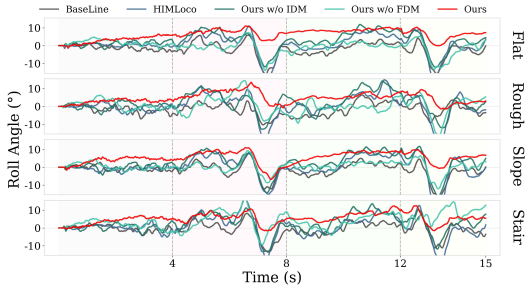


Fig. 5: Four types of active load actions : front-back, left-right, rotation, and random.

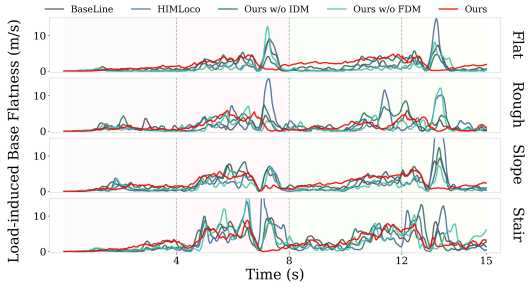
A. Simulation Experiments

The experiments in this study were conducted on the MuJoCo simulator [26]. Dynamic load experiments were designed to evaluate the robot's stability and adaptability under different active load disturbances. We tested four scenarios: flat, stair, rough, and slope, as shown in Fig. 4. To investigate the adaptability of the proposed model to dynamic loads, we employed an actively moving robotic arm as the dynamic load carrier. Four types of active load actions were designed: front-back, left-right, rotation, and random, as shown in Fig. 5. Within a 15-second evaluation window, staged active motion amplitudes were applied: from 0–4 s, the arm remained stationary; from 4–8 s, large-amplitude motion was generated by applying high torques; from 8–12 s, small-amplitude motion was generated by applying low torques; and from 12–15 s, medium-amplitude motion was applied, we collected three metrics in the robot's base frame: base roll deviation angle, load-induced base flatness, and linear velocity tracking error, as shown in Fig. 6.

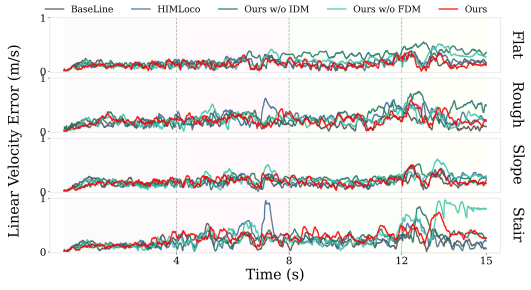
Fig. 6a shows the base roll deviation angle for all methods across four different terrains under active dynamic load conditions. Ours exhibits performance stability, demonstrating a significantly lower roll angle deviation across all terrain. Notably, the time point at which the load movement amplitude changes significantly is noticed more effectively by our method than by other methods, indicating that our method is robust to active load movement. Comparing the BaseLine and Ours w/o FDM shows the effectiveness of our proposed active dynamic load model. Specifically, on flat terrain, where dynamic loads are the dominant disturbance, both Ours w/o FDM and Ours perform similarly. This highlights our model's ability to effectively counteract load disturbances. A comparison between HIMLoco and Ours w/o IDM reveals that our dynamic model is effective in representing the surrounding environment, which is crucial for maintaining stability. Both methods perform comparably across all terrains except for the flat one, where they



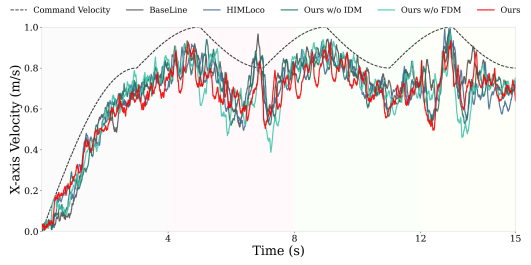
(a) Base roll deviation angle.



(b) Load-induced base flatness.



(c) Linear velocity tracking error.



(d) X-axis Linear Velocity.

Fig. 6: Results of dynamic experiments under a 3 kg robotic arm load. The robot is commanded to move along the x-axis (forward) at velocities of 0.8–1.0 m/s (for case (d)). The stair step is 0.05 m high and 0.3 m wide, and the slope is approximately 10 degrees. Over a 15-second period, the robotic arm employs staged active motion with varying torque: stationary (0–4 s), high-torque large-amplitude motion (4–8 s), low-torque small-amplitude motion (8–12 s), and medium-amplitude motion (12–15 s).

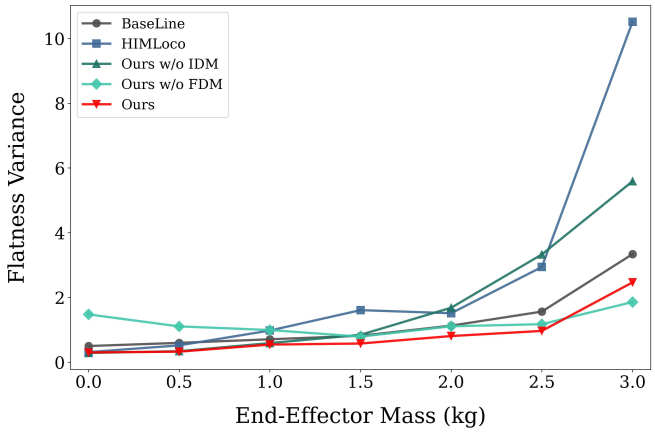


Fig. 7: Comparison of fundamental flatness variance for balancing control of arm end-effector under different load masses.

surpass the BaseLine. Fig. 6b shows the load-induced base flatness penalty, a metric that quantifies the robot's ability to maintain a flat and level base despite active dynamic loads. Unlike the roll angle, which measures rotational stability, this penalty quantifies the overall vertical displacement and deviation of the base's trajectory. A lower penalty indicates better resistance to the load, resulting in smoother and more stable movement. The results clearly show that our method consistently achieves a lower penalty, especially during high-magnitude load motions. This further validates our method's robustness against complex disturbances. Fig. 6c shows the linear velocity tracking error. Overall, our method generally performs slightly inferior to HIMLoco in this metric. A closer examination reveals that when the load exhibits active dynamic disturbance, the robot deliberately sacrifices its velocity tracking performance, as shown in Fig. 6d. We attribute these results to two primary factors: first, HIMLoco employs a dedicated velocity estimator, which provides a more accurate reference for tracking control; second, our method is designed to prioritize stability and balance under active dynamic loads, leading to a deliberate trade-off in velocity performance. Despite this trade-off, during sudden large-amplitude load motions, the error of our method remains comparable to that of HIMLoco. This indicates that, while stability is prioritized, our method is still capable of maintaining competitive velocity tracking performance, particularly under challenging disturbance conditions.

To further evaluate the robustness of our method for balancing control under active dynamic load disturbances, we conducted balancing experiments testing the control performance of robotic end-effectors across a mass range of 0 to 3 kg (in 0.5 kg increments), using fundamental flatness variance as the performance metric. As shown in Fig. 7, our method consistently achieves the minimum variance across the entire mass test range, significantly outperforming all other methods. Particularly in the heavy-load range of 2.0–3.0 kg, variance fluctuations are markedly lower than

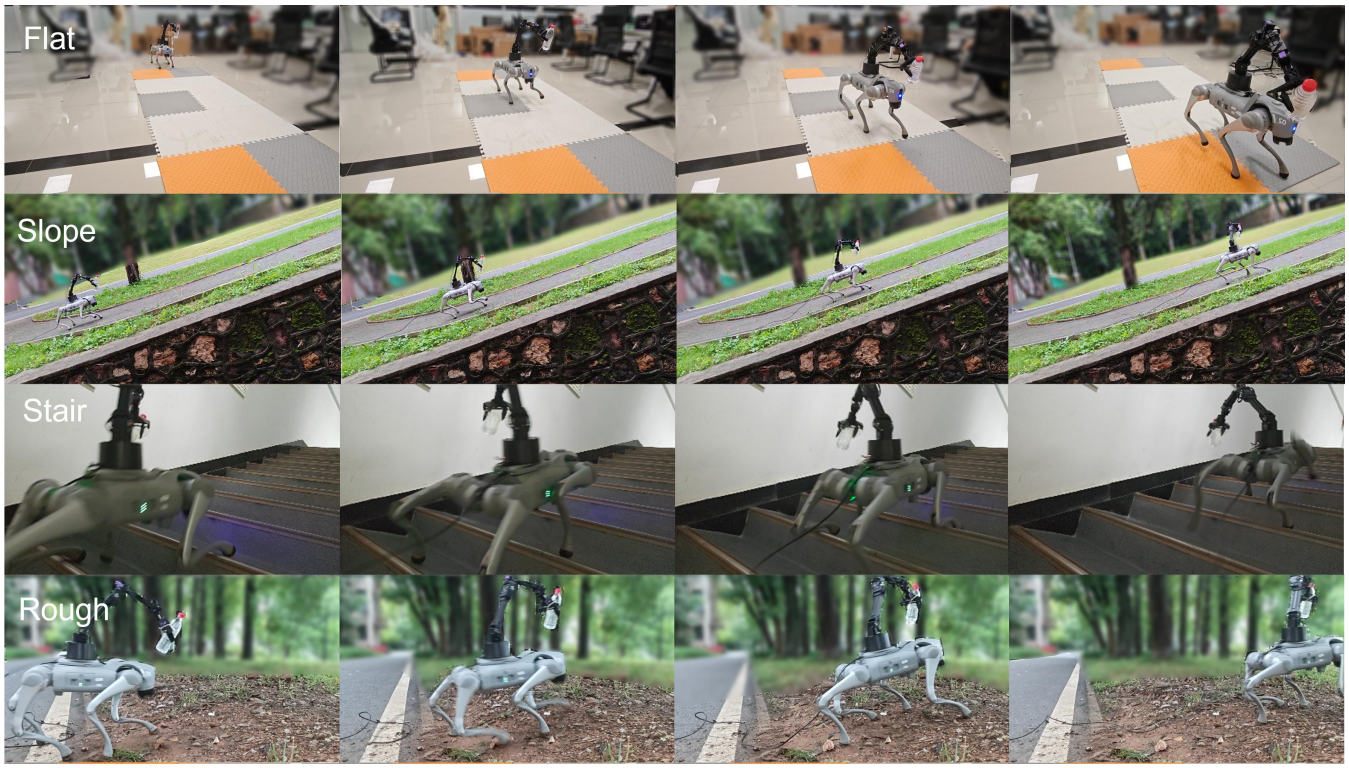


Fig. 8: Sim-to-real experiment with Unitree D1 mechanical arm as dynamic load on four scenarios.

TABLE V: Load Flatness Performance of Different Motion Patterns with a 3.0 kg Mass of End-Effector

Model	F&B	L&R	Rotate	Random
BaseLine	0.591	1.479	1.832	2.081
HIMLoco	0.375	1.550	1.824	2.476
Ours w/o IDM	0.779	1.472	2.211	2.275
Ours w/o FDM	**0.207**	**0.817**	1.664	1.416
Ours	0.482	1.087	**1.390**	**1.193**

those of the baseline models. Notably, as load mass increased from 0.0 kg, baseline methods exhibited performance fluctuations due to the absence of adaptive compensation mechanisms. In contrast, our method ensured stable control performance by leveraging active dynamic load modeling and inverse dynamics modeling for dynamic disturbance prediction. This result demonstrates that our method effectively suppresses base disturbances caused by unknown load variations at the end-effector, maintaining excellent balancing performance. Furthermore, the performance of the w/o FDM version is extremely close to that of the full method and significantly outperforms other methods.

Table V evaluates load flatness performance across four motion modes (front-back, left-right, rotation, and random) on flat terrain with a 3.0 kg mass of end-effector. The ours w/o FDM achieves the lowest variance in Front-Back (F&B) and Left-Right (L&R), while our method demonstrates outstanding performance in rotation and random. This subtle discrepancy in linear motions can be attributed to the high predictability of such trajectories. In these scenarios, the additional information introduced by active modeling may act as a source of information redundancy. Compared to linear motions (F&B, L&R), rotational and random motions exhibit higher variance, revealing differential impacts of motion patterns on fundamental stability. These results fully validate the effectiveness of active dynamic load modeling in balance control.

B. Real-World Experiments

To rigorously validate the effectiveness of the proposed framework, we conducted an extensive set of real-world experiments on a Unitree Go2 quadruped robot integrated with a Unitree D1 mechanical arm. The arm is tasked with grasping objects of varying masses and executing predefined motion trajectories, thereby introducing structured yet nontrivial dynamic loads to the system. As illustrated in Fig. 8, we systematically evaluated the robot across four representative environments: flat, slope, stair, and rough.

The experimental results demonstrate that our method, relying only on proprioceptive feedback, enables seamless zero-shot transfer from simulation to the physical world. Crucially, the robot exhibits consistent adaptability and stability under actively varying dynamic loads on complex terrain.

These findings highlight not only the robustness of the proposed approach but also demonstrate strong generalization capabilities across diverse real-world conditions.

V. CONCLUSIONS

In this work, we proposed an active dynamic load modeling method to enhance the stability of quadruped robots carrying loads with active and dynamic characteristics. Utilizing this method, we have established a reinforcement learning framework based on dynamic models. Within this framework, an inverse dynamics model is used to learn and adapt to the active dynamic characteristics of the load, while a forward dynamics model predicts environmental changes, such as terrain variations. This dual-model architecture allows the robot to simultaneously compensate for load disturbances and adjust to external terrain changes. We validated our method through extensive simulation and real-world experiments across diverse terrains, including slopes, stairs, and rough terrain, and with various active dynamic load conditions. Specifically, we used a robotic arm mounted on the robot's base as the load, which exhibited active movements such as front-back, left-right, rotation, and random actions, and also grasped objects with varying masses. The results demonstrate that our method consistently and significantly outperforms baseline and state-of-the-art methods in balance control, proving its superior adaptability and robustness. While our method did not show an advantage in velocity tracking, we identify this as a key direction for future research.

REFERENCES

[1] 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.

[2] Y. Cheng, H. Liu, G. Pan, H. Liu, and L. Ye, "Quadruped robot traversing 3d complex environments with limited perception," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 9074–9081.

[3] J. Hooks, M. S. Ahn, J. Yu, X. Zhang, T. Zhu, H. Chae, and D. Hong, "Alphred: A multi-modal operations quadruped robot for package delivery applications," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5409–5416, 2020.

[4] N. Li, J. Cao, and Y. Huang, "Fabrication and testing of the rescue quadruped robot for post-disaster search and rescue operations," in *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*. IEEE, 2023, pp. 723–729.

[5] Y. Chen, Z. Wei, S. G. Vougioukas, S. K. Gupta, and Q. Nguyen, "Autonomous visual navigation for quadruped robot in farm operation," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2024, pp. 3518–3524.

[6] V. K. Kurva and S. Kolathaya, "Mule-multi-terrain and unknown load adaptation for effective quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 11, no. 1, pp. 474–481, 2025.

[7] L. Chang, Y. Nai, H. Chen, and L. Yang, "Beyond robustness: Learning unknown dynamic load adaptation for quadruped locomotion on rough terrain," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 10 282–10 288.

[8] Q. Yao, C. Wang, J. Wang, L. Meng, S. Yang, Q. Zhang, and D. Wang, "Adaptive legged manipulation: Versatile disturbance predictive control for quadruped robots with robotic arms," *Robotics and Autonomous Systems*, vol. 167, p. 104468, 2023.

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

[10] S. Chen, Z. Wan, S. Yan, C. Zhang, W. Zhang, Q. Li, D. Zhang, and F. U. D. Farrukh, "SLR: Learning quadruped locomotion without privileged information," in *8th Annual Conference on Robot Learning*, 2024.

[11] A. Mousa, N. Karavis, M. Caprio, W. Pan, and R. Allmendinger, "Tar: Teacher-aligned representations via contrastive learning for quadrupedal locomotion," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)*, 2025.

[12] P. Roth, J. Frey, C. Cadena, and M. Hutter, "Learned perceptive forward dynamics model for safe and platform-aware robotic navigation," in *Robotics: Science and Systems (RSS 2025)*, 2025.

[13] A. Echchahed and P. S. Castro, "A survey of state representation learning for deep reinforcement learning," *Transactions on Machine Learning Research*, 2025.

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

[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[17] I. M. A. Nahrendra, B. Yu, and H. Myung, "Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5078–5084.

[18] S. Luo, S. Li, R. Yu, Z. Wang, J. Wu, and Q. Zhu, "Pie: Parkour with implicit-explicit learning framework for legged robots," *IEEE Robotics and Automation Letters*, 2024.

[19] J. Long, Z. Wang, Q. Li, L. Cao, J. Gao, and J. Pang, "Hybrid internal model: Learning agile legged locomotion with simulated robot response," in *The Twelfth International Conference on Learning Representations*, 2024.

[20] J. Wang, Y. Lu, and H. Zhao, "Cloud: contrastive learning of unsupervised dynamics," in *Conference on Robot Learning*. PMLR, 2021, pp. 365–376.

[21] T. McInroe, L. Schäfer, and S. V. Albrecht, "Multi-horizon representations with hierarchical forward models for reinforcement learning," *Transactions on Machine Learning Research*, 2024.

[22] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," in *International Conference on Learning Representations*, 2019.

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

[24] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[25] X. Wang, Y. Chen, and W. Zhu, "A survey on curriculum learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 4555–4576, 2021.

[26] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4397–4404.