ALARM: Safe Reinforcement Learning with Reliable Mimic for Robust Legged Locomotion

Qiqi Zhou, Hui Ding, Teng Chen, Han Jiang, Guoteng Zhang, Bin Li, Xuewen Rong, Yibin Li

***Abstract*—Legged robots are supposed to traverse complicated environments, which makes it challenging to design a model-based controller due to their functional complexity. Currently, using deep reinforcement learning to improve the adaptability of robots in complex scenarios has been a major research trend. In this paper, we propose Adaptive Latent Aggregation for Reliable Mimicry (ALARM), a reinforcement learning framework that enables both safe and robust locomotion in legged robots using only proprioception. This work features a one-step teacher-student training paradigm by constructing an adaptive aggregation strategy, which combines the merits of imitation learning and reinforcement learning splendidly. The framework integrates normalized penalized proximal policy optimization, which penalizes constraint-violating behaviors while optimizing locomotion policy. This effectively ensures the safety of the robot during locomotion. Experiments demonstrate ALARM's adaptability across challenging terrains, with superior safety and robustness. Our method also facilitates efficient sim-to-real transfer, outperforming the state-of-the-art method in constraint adherence and performance, offering a promising approach for real-world legged robot applications.**

***Index Terms*—Legged robots, imitation learning, reinforcement learning, deep learning for robot control.**

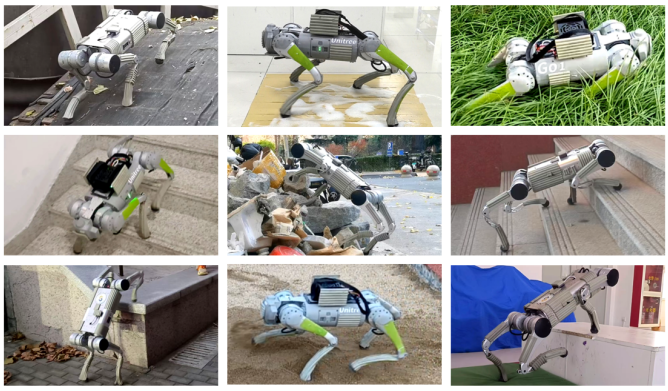

Fig. 1. Our method enables legged robots to safely traverse a variety of challenging terrains, such as dense grass, wet sand, slippery tile, pile of rocks, continuous stairs, and high platform, relying solely on proprioception. Our robots demonstrate excellent robustness and generalization in the real world.

I. Introduction

ANIMALS evolved legs and feet to adapt complex terrains and high-speed movement. This inspired researchers to design legged robots to work in some demanding scenarios, such as factory inspections, fire rescue and unknown environment exploration. The locomotion of legged robots in complex terrain has become a current research hotspot.

Traditional control methods, such as Virtual Model Control(VMC), Model Predictive Control(MPC) and Whole-Body Control (WBC), play a critical role in the locomotion control of legged robots [1]–[3]. These methods rely on precise mathematical models and sophisticated algorithms that consider the robot's complex whole-body dynamics, demonstrating good stability and reliability. However, these control methods often rely on specific tasks, scenarios, and highly idealized assumptions. As task requirements or environments change, these methods typically require lengthy design processes and extensive parameter tuning. This high dependency not only leads to increased complexity but also limits their generalization ability in diverse and dynamic environments.

To overcome these limitations, reinforcement learning (RL) has emerged as a promising tool, showing significant potential in synthesizing robust legged locomotion [4]–[6]. Unlike model-based methods, RL learns policies through continuous interactions between the agent and the environment, making it particularly suitable for designing locomotion controllers. To enable a robot to perform agile movement in complex environments, terrain information is essential. Previous studies [7], [8] have leveraged external sensors, such as cameras and LiDARs, to capture terrain information, demonstrating excellent traversal capabilities in complex terrains. However, external image sensors are limited by factors such as lighting conditions. In environments such as grasslands, snowfields, vegetation, and puddles, external sensors lack direct perception of surface physical properties. The information they provide often fails to reliably represent the real environment. The proprioception-based methods provide more direct and real-time feedback on the robot's own state. Terrain information can be effectively encoded through the robot's posture and joint data, enabling lightweight and robust locomotion without reliance on external sensors.

Building upon this concepts, approaches inspired by imitation learning [9]–[11] have been proposed, encoding terrain information effectively using only proprioceptive data, such as the robot's posture and joint information. These methods employ the teacher network to generate expert behaviors using privileged information, while the student network rely solely on proprioception to replicate these decisions in sim-

This work was supported in part by the National Natural Science Foundation of China(62203268,62373217), in part by the Youth Innovation Team Project of Higher Education Institutions in Shandong Province(2023KJ029).

Qiqi Zhou, Hui Ding, Teng Chen, Han Jiang, Guoteng Zhang, Xuewen Rong, and Yibin Li are with the School of Control Science and Engineering, Shandong University, Jinan 250061, China (e-mail: kkzhou@mail.sdu.edu.cn; huiding@mail.sdu.edu.cn; chenteng100@sdu.edu.cn; 202420782@mail.sdu.edu.cn; guoteng@email.sdu.edu.cn; rongxw@sdu.edu.cn; liyb@sdu.edu.cn).

Bin Li is with the School of Mathematics and Statistics, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China (e-mail: binli@qlu.edu.cn).

Teng Chen is the corresponding author.

Videos and code release: https://sucro-legged.github.io/ALARM/ .

ilar situations. Although the student can approximate the teacher, subtle differences may accumulate, leading to state distribution shifts. When such discrepancies cause the student to encounter adverse conditions, the teacher often fails to provide adequate guidance. Moreover, the sequential training of the teacher and student results in low data efficiency. To mitigate this issue, Ross et al. introduced controllable factors to adjust the reliance on expert behavior, ensuring sufficient learning of the student policies from the expert [12]. The dataset aggregation (DAGGER) algorithm, proposed as part of this work, applies the concept of no-regret learning from online learning to imitation learning. Subsequent studies Wu et al. adopted DAGGER to train teacher-student networks, achieving promising results in quadrupedal locomotion [13], [14]. However, this parameter adjustment mechanism, which changes monotonically over time, is overly idealized. If the parameter adjustment mechanism is not optimized, simple no-regret learning frameworks often struggle to adapt to complex environments and dynamic task requirements.

To further enhance the performance of legged robots in complex environments, representation learning has been employed to extract hidden dynamics from historical data. For instance, Fu et al. introduced the regularized online adaptation (ROA) [15], which eliminates the need for traditional two-phase training by enabling unified policies to be directly deployed in real environments through online adaptation and regularization techniques. Additionally, Nahrendra et al. utilized an asymmetric actor critic (A2C) [16] framework combined with variational auto-encoder for environmental feature extraction, achieving improved terrain adaptability for quadrupedal robots [17]. Long et al. introduced a hybrid internal model based on contrastive learning to estimate disturbances encountered by quadrupedal robots in external environments [18]. These methods are sensitive to parameters, and their effectiveness often depends on precise parameter configurations; otherwise, they can negatively impact the stability and performance of training.

Safety in locomotion behavior is a fundamental requirement for long-term exploration by legged robots. In model-based approaches, researchers typically incorporate physical constraints, such as joint velocity limits, torque limits, or safety regulations, to ensure safe locomotion in the real world [19], [20]. Ensuring safe and efficient exploration of locomotion policies in unknown environments is also a major challenge for RL in robotic applications. Gangapurwala et al. used CPPO to optimize dynamic policies for quadrupedal robots while enforcing specific safety and performance constraints during locomotion [21]. Zhang et al. proposed the penalized proximal Policy optimization (P3O) algorithm, which simplifies constrained policy iteration by minimizing an equivalent unconstrained problem, thereby improving computational efficiency [22]. Kim et al. employed interior point policy optimization (IPO) to handle constraints by transforming them into equivalent unconstrained problems, simplifying reward engineering [23]. Additionally, Chane et al. introduced the constrained as termination (CaT) algorithm, which converts constraints into stochastic termination conditions, ensuring high compliance without introducing excessive complexity or computational overhead in real-world scenarios [24]. While these methods effectively constrain agent behavior, their complexity poses challenges for practical applications.

In this paper, we proposed a computationally efficient algorithm, called **A**daptive **L**atent **A**ggregation for **R**eliable **M**imicry **(ALARM)**, which trains a safe and robust locomotion policy for legged robots with only proprioception through constrained reinforcement learning. Our main contributions are as follows:

1) We have developed a novel one-step reinforcement learning framework that leverages an adaptive aggregation strategy guided by hybrid advantage estimation to provide more explicit update metrics for no-regret learning. This framework facilitates a smooth transition from teacher to student, prevents distributional shifts inherent in asynchronous teacher-student frameworks, and improves data efficiency.
2) This paper introduces cost indicator function as a state constraint for the robot and constructs the optimization objective for reinforcement learning based on the normalized penalized proximal policy optimization. We extend the multi-head critic network within the A2C framework, aiming to efficiently and concisely implement the normalized generalized advantage estimation for both reward and costs. The experimental results demonstrate that our algorithm enables the robot to exhibit significantly fewer constraint violations while adapting to complex terrain.
3) We conducted comparative experiments with current mainstream algorithms and performed ablation studies on the components of the framework. The experimental results indicate that each module of the ALARM framework contributes positively to the overall enhancement in performance. Furthermore, our policy has been successfully deployed on multiple quadrupedal robot platforms, such as Go1 and SDUQuad48 shown in Fig. 1, demonstrating exceptional agility and adaptability on complex and unstructured terrains, thus validating its practicality and generalizability in real-world robotic applications.

II. METHOD

An overview of the proposed reinforcement learning framework, ALARM, for legged locomotion is shown in Fig. 2. In this section, we formulate our problem and develop the proposed framework.

A. Problem Formulation of Reinforcement Learning

The blind locomotion of legged robots is proprioceptive without any external sensors. In this work, we model the agent's interaction with the environment as an infinite-horizon partially observable markov decision process (POMDP), which can be defined by the tuple $\mathcal{M} = (S, A, P, R, p_0, \gamma)$. $s_t \in S$ define the full state including the observable state o_t and the privileged state s_t^p at time step t. Given the current state, the agent's policy takes an action $a_t \in A$, and then the environment moves to the next state, according to the probability of state transition $p(s_{t+1}|s_t, a_t)$. This movement is evaluated by the reward r_t from the reward function $R(r_t|s_{t+1}, a_t)$. p_0 is

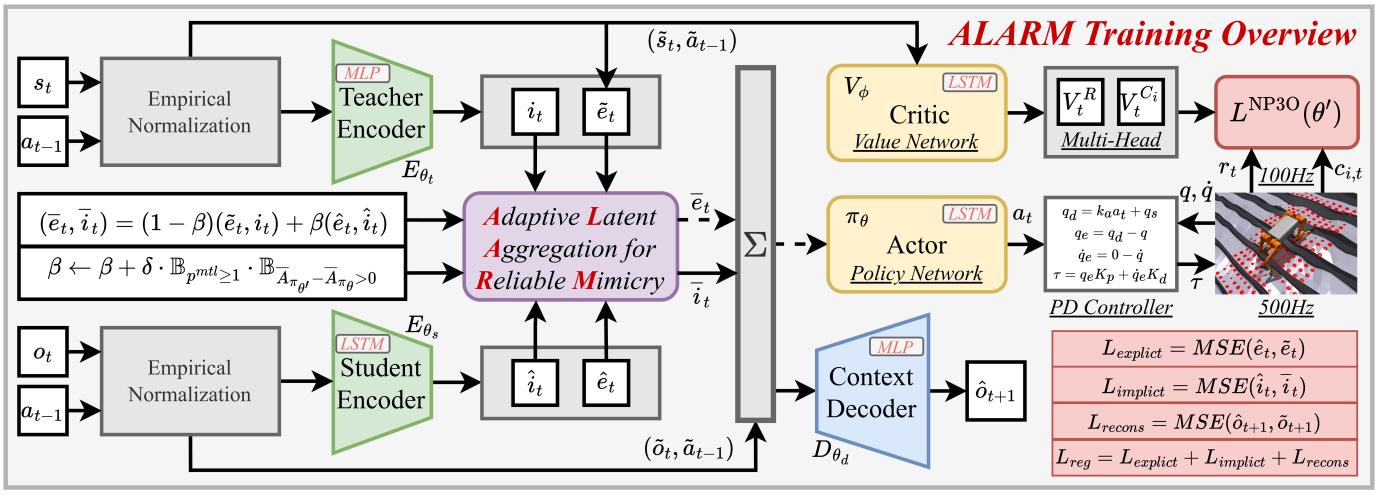

Fig. 2. Overview of the training method. We first normalize state information o_t and last action a_{t-1}. Then, the teacher encoder uses the processed privileged information to output the reference latent variables, while the student encoder imitates the implicit encoding and estimates the explicit states. By adopting the adaptive aggregation strategy, the RL network can leverage the privileged information to accelerate convergence during the early stages of training, ultimately relying solely on proprioceptive feedback to navigate challenging terrains. During this process, the decoder provides contextual guidance for the latent representation. NP3O effectively constrains the robot's behavior. The overall optimization objective consists of the supervised learning loss L_{reg} and the reinforcement learning loss $L^{\text{NP3O}}(\theta')$. (Dashed arrows indicate that gradient backpropagation is not performed).

the initial state distribution. The discount factor $\gamma \in [0, 1]$ represents the attention to the future reward.

For safe reinforcement learning, the POMDP is extended to the constrained Markov decision process (CMDP) [25]. The state transition is augmented with constraint violations $\{c_1, c_2, \ldots, c_k\}$ from cost functions $\{C_1, C_2, \ldots, C_k\}$ and corresponding limits $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_k\}$. Let $J_R(\pi)$ and $J_{C_i}(\pi)$ denote the expected discounted return of policy π with respect to the reward and cost functions, defined as follows:

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

$$J_{C_i}(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t C_i(s_t, a_t, s_{t+1})\right], \tag{2}$$

yielding the constrained optimization problem:

$$\begin{aligned} \max_{\pi} \quad & J_R(\pi) \\ \text{s.t.} \quad & J_{C_i}(\pi) \leq \epsilon_i, \forall i \in \{1, \ldots, k\}, \end{aligned} \tag{3}$$

Derived by the performance difference lemma by the P3O, the constrained optimization problem above can be reformulated as follows:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{s\sim d^{\pi}}[A_R^{\pi}(s, a)] \\ \text{s.t.} \quad & J_{C_i}(\pi) + \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{a\sim\pi}\left[A_{C_i}^{\pi}(s, a)\right] \leq \epsilon_i, \forall i. \end{aligned} \tag{4}$$

where $d^{\pi}(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$ denote the discounted future state distribution, A_R^{π} is the reward generalized advantage estimation (GAE), and $A_{C_i}^{\pi}$ is the ith cost GAE.

In this work, we extend the optimal objective of P3O to the normalized penalized proximal policy optimization (NP3O) objective [26] with penalties on the constraint violations, defined as:

$$L^{\text{NP3O}}(\theta') = \tilde{L}_R^{\text{CLIP}}(\theta') - \sum_i \kappa_i \cdot \max\left\{0, \tilde{L}_{C_i}^{\text{VIOL}}(\theta')\right\} \tag{5}$$

where κ_i is the weight of each constraint and $\tilde{L}_{C_i}^{\text{VIOL}}$ is the normalized constraint violations defined as:

$$\tilde{L}_{C_i}^{\text{VIOL}}(\theta') = \tilde{L}_{C_i}^{\text{CLIP}}(\theta') + \frac{(1-\gamma)(J_{C_i}(\pi_\theta) - \epsilon_i) + \mu_{C_i}}{\sigma_{C_i}} \tag{6}$$

with the surrogate objective for the reward and costs

$$\tilde{L}_R^{\text{CLIP}}(\theta') = \mathbb{E}\left[\min(r_t(\theta')\tilde{A}_{R,t}^{\pi_\theta}, \text{clip}(r_t(\theta'))\tilde{A}_{R,t}^{\pi_\theta}\right], \tag{7}$$

$$\tilde{L}_{C_i}^{\text{CLIP}}(\theta') = \mathbb{E}\left[\max(r_t(\theta')\tilde{A}_{C_i,t}^{\pi_\theta}, \text{clip}(r_t(\theta'))\tilde{A}_{C_i,t}^{\pi_\theta})\right] \tag{8}$$

where $r_t(\theta')$ denotes the probability ratio $\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}$. $\tilde{A}_{R,t}^{\pi_\theta} = \frac{A_{R,t}^{\pi} - \mu_R}{\sigma_R}$ and $\tilde{A}_{C_i,t}^{\pi_\theta} = \frac{A_{C_i,t}^{\pi} - \mu_{C_i}}{\sigma_{C_i}}$ are normalized GAE of the reward and costs, respectively.

State Space: The observable state, i.e., the proprioception of legged robots $o_t = (\omega_t, \varphi_t, q_t - q_{stand}, \dot{q}_t, c_t)$. Specifically, ω_t, φ_t, q_t and $\dot{q}_t$, are the angular velocity, the base orientation, the joint position, and the joint velocity, measured directly from the joint encoders and the inertial measurement unit (IMU). q_{stand} is the default joint position in which the robot stands still on the plane ground. The given velocity command $c_t = (v_x^{des}, v_y^{des}, \omega_z^{des})$ contains the desired linear and angular velocity. For blind locomotion, the privileged state $s_t^p = (s_t^h, s_t^r)$. s_t^h is the uniform rectangular sampling of the elevation map around the robot. s_t^r is the randomization information of the robot such as rigid bodies and joints with specific explanation in Section III-B. $s_t = (e_t, o_t, s_t^p)$ is the full state space, where e_t is the explicit states including the base linear velocity v_t, the foot height and the foot contact probability.

Action Space: We train the policy to output the adjustment

of joint angle around q_{stand}. The joint torque of the robot is defined as follows:

$$\tau = K_p(k_a a_t + q_{stand} - q_t) - K_d \dot{q}_t, \tag{9}$$

where a_t is the policy network output and k_a is the action scale to ensure the stability during training. $K_p = 20$ and $K_d = 0.5$ are the proportional and derivative gains in this work, respectively.

Empirical Normalization (EN): The numerical uniformity of the data can accelerate the network convergence process, but in RL tasks, the state space and the action space are often dynamically changing and unbounded. We constructed two empirical normalization networks to output $(\tilde{o}_t, \tilde{a}_{t-1})$ and $(\tilde{s}_t, \tilde{a}_{t-1})$ respectively, which empirically normalize the mean and variance of the proprioception o_t, the full state s_t and the past action a_{t-1}. In training mode, the empirical mean and variance of the accumulated data are continuously updated and the network is trained and inferred based on the regularized data.

B. Adaptive Latent Aggregation for Reliable Mimicry

To facilitate learning and reduce the difficulties during the training process, motivated by privileged learing [27], a latent representation i_t is constructed as an implicit feature dimensionality reduction of the privileged state s_t^p. However, the privileged state and its encoded extrinsic vector are not accessible during deployment in the real world. To achieve sim-to-real transfer, we not only build the teacher encoder (TE) E_{θ_t} as an environment factor network but also the student encoder (SE) E_{θ_s} as an adaptation network, parameterized by θ_t and θ_s respectively. In addition, [17], [28], [29] emphasize that explicit state estimation using a learned network significantly improves the robustness of the locomotion policy by accelerating learning processes and eliminating the accumulated drift. Therefore, SE is also designed to output the explicit state estimation $\hat{e}_t$. TE and SE are detailed in (10).

$$\begin{aligned} i_t &= E_{\theta_t}(\tilde{s}_t, \tilde{a}_{t-1}) \\ \hat{e}_t, \hat{i}_t &= E_{\theta_s}(\tilde{o}_t, \tilde{a}_{t-1}). \end{aligned} \tag{10}$$

Our innovative ALARM framework enables one-step training of the teacher-student network. Motivated by adaptive asymmetric DAgger (A2D) [30], we design an adaptive coefficient β for each agent to adjust the weight parameter between ground truth and mimicry, which is detailed as follows:

$$(\bar{e}_t, \bar{i}_t) = (1-\beta)(\tilde{e}_t, i_t) + \beta(\hat{e}_t, \hat{i}_t) \tag{11}$$

In A2D, the update conditions for the adaptive coefficient β often depend solely on the number of training iterations, with the implicit assumption that the policy improves progressively during training. This update strategy may exacerbate the instability in the RL training process, as it fails to account for the potential fluctuations or regressions in policy performance during training. Therefore, we propose hybrid advantage estimation (HAE) $\overline{A}_{\pi_\theta}$ in (12), to determine whether updating from the current policy π_θ to the new policy $\pi_{\theta'}$ has led to an increase in the reward advantage and a decrease in the cost advantage.

$$\overline{A}_{\pi_\theta} = \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta} \\ a\sim\pi_\theta}} \left[\tilde{A}_R^{\pi_\theta}(s,a) - \sum_{i=1}^{k} \kappa_i \cdot \tilde{A}_{C_i}^{\pi_\theta}(s,a) \right] \tag{12}$$

This can be used as a metric for policy improvement to dynamically adjust β. Generally, we update β following the rule:

$$\beta \leftarrow \beta + \delta \cdot \mathbb{B}_{p^{mtl}\geq 1} \cdot \mathbb{B}_{\overline{A}_{\pi_{\theta'}} - \overline{A}_{\pi_\theta} > 0}. \tag{13}$$

where $\mathbb{B}$ represents the indicator function of its corresponding condition detailed in (14).

$$\mathbb{B}_x = \begin{cases} 1 & \text{for} \quad x \ satisfied \\ 0 & \text{for} \quad x \ not \ satisfied \end{cases} \tag{14}$$

p^{mtl} represents whether the agent has reached the maximum terrain level. The curriculum learning is detailed in Section III-A. The β is always equal to 0 at the beginning of training. When the agent reaches the maximum terrain level, we believe that the latent representation of teacher and HAE are relatively accurate. After that, the teacher is no longer updated, and β receives an increment δ based on whether the difference in HAE between π_θ and $\pi_{\theta'}$ is greater than 0. Based on this comprehensive metric, the student can achieve reliable mimicry of the teacher while avoiding excessive dependence.

C. Reward and Cost Terms Design

In this work, the reward function consists of a task component r_t^g, a style component r_t^s, and an energy component r_t^e, such that

$$r_t = r_t^g + r_t^s + r_t^e \tag{15}$$

The task reward term consists of desired body velocity tracking and undesired regularization. The style reward term is used to track the foot position prior $p_{t,\text{feet}}^{\text{slip}}$ generated by the spring-loaded inverted pendulum (SLIP) [31]. To improve the robustness of the SLIP model, the target step height is related to the terrain variance around the robot. Meanwhile, the prior confidence in the SLIP model is inversely proportional to the terrain level l^{terrain}. To achieve energy-saving behavior and smooth movement in the real world, we add the energy reward term to the total reward. The joint velocity, acceleration, and torque are penalized with a smaller coefficient and the action change is also limited. All reward terms are detailed in Table I.

Taking into account the safety of our task, we construct five constraints as follows:

- Joint limits $(c_{1,t}, c_{2,t}, c_{3,t})$: The upper and lower bounds of the joint positions, the joint velocities, and the joint torques are limited. These cost functions return 0 if they are within bounds, and $1/n_{\text{joints}}$ if the constraint violation occurs.
- Undesired collision $(c_{4,t})$: Contact between terrains and the rest of the legged robot except the feet should be avoided. The cost function returns 1 if there are any collisions between the terrain and links of the robot, and 0 if not.

TABLE I
REWARD TERMS

Reward Terms	Equation	Weight
Task r_t^g	$\exp\left(-4\|v_{t,xy} - v_{t,xy}^{\text{des}}\|_2\right)$	2.0
	$\exp\left(-4\|\omega_{t,z} - \omega_{t,z}^{\text{des}}\|_2\right)$	1.0
	$\|v_{t,z}\|_2$	-2.0
	$\|\omega_{t,xy}\|_2$	-0.05
Style r_t^s	$\exp\left(-\frac{4\|p_{t,\text{feet}} - p_{t,\text{feet}}^{\text{slip}}\|_2}{1+l^{\text{terrain}}}\right)$	0.2
Energy r_t^e	$\|\dot{q}_t\|_2 + 2\|\tau\|_2$	-1e-4
	$\|\ddot{q}_t\|_2$	-2.5e-7
	$\|a_t - a_{t-1}\|_2$	-0.01
	$\|a_t - 2a_{t-1} + a_{t+1}\|_2$	-0.01

- Center of Mass (COM) frame ($c_{5,t}$): The COM height h_b from the ground and the attitude angle of the COM frame are restricted to a reasonable range with respect to gravity. The cost function returns 1 if h_b is too close to the ground or if the robot orientation is badly tilted, and 0 if not.

All cost functions are indicator functions, and each corresponding limit ϵ_i is 0 and each corresponding weight κ_i is 1, effectively avoiding parameter design.

III. TRAINING

Simulation: In this work, We trained 4096 agents using massive parallelism in the IsaacGym simulator [32] on a single NVIDIA RTX 4090D GPU. All networks acquire the ability to traverse complex terrains after a single training phase, which takes 500 million simulation timesteps, equivalent to three hours of wall-clock time. Every RL step runs at 100Hz and the actuator servos the desired joint position and velocity at 500Hz. The episode lasts for a maximum of 1000 steps and terminates early if bad situations occur, such as the body attitude tilting too much or the trunk touching the ground.

A. Curriculum Learning

In this work, we construct five types of terrain: up-slopes, down-slopes, up-stairs, down-stairs, and discrete obstacles. We have designed a terrain curriculum to improve training efficiency based on [13]. A 20×10 rectangle grid environment is arranged, where every grid has a length and width of 8 m with an elevation map. Each of the 20 rows has the same terrain type with increasing level l^{terrain} that ranges in 10 columns. The corresponding terrains are as follows:

- Up-slopes: with an inclination of $\frac{l^{\text{terrain}} \times 30^\circ}{9}$ cm;
- Down-slopes: with an inclination of $-\frac{l^{\text{terrain}} \times 45^\circ}{9}$ cm;
- Up-stairs: with step height of $18 \times \frac{l^{\text{terrain}}}{9} + 5$ cm;
- Down-stairs: with step height of $-20 \times \frac{l^{\text{terrain}}}{9} - 5$ cm;
- Discrete obstacles: with 20 rectangles whose height are $23 \times \frac{l^{\text{terrain}}}{9} + 5$ cm.

All terrains are given random height noise $U(-5, 5)$ cm. At the beginning of training, all agents are assigned to the terrain with the lowest level. Only robots crossing the terrain boundary with more than 85% of the average linear velocity tracking reward can be moved to the corresponding higher-level terrain training. In contrast, they should be moved to a lower-level terrain if the absolute path the robot has traveled before termination is less than 50% of the product of the speed command and the maximum step time. To avoid skill forgetting, when the robot solves the most difficult terrain, it should be assigned a corresponding terrain of random difficulty and continue to execute this terrain curriculum repeatedly.

TABLE II
DOMAIN RANDOMIZATIONS

Parameters	Range[Min, Max]	Unit
Ground Friction	[0.1, 3.0]	None
Ground Restitution	[0.0, 1.0]	None
Payload Mass	[-1.0, 3.0]	Kg
Payload Position	[-0.1, 0.1]	Kg
Motor Strength	[0.8, 1.2]	Nm
Motor K_p	[0.8, 1.2]	None
Motor K_d	[0.8, 1.2]	None
Initial Joint Position	[0.5, 1.5]	rad
Push velocity	[-1.0, 1.0]	m/s
System delay	[0.0, 15.0]	ms

B. Domain Randomizations

It is a common but tough issue to address the simulation to the real-world transfer in the end-to-end control methods. Dynamics randomization is a promising way to improve the robustness of the policy and facilitate transfer. Therefore, every 10 seconds during training, we randomize ground friction and restitution, the mass and position of payload applied to the body of the robot, the strength and PD gains of the motors, the initial joint positions, the push velocities with random directions, and the system delay. Some of these randomized parameters are considered as s_t^r, which is part of the privileged state. The randomization ranges for each parameter are detailed in Table II.

C. Network Details

Auto-Encoder (AE): In this work, the teacher encoder E_{θ_t} directly encodes the noise-free and normalized full state space and the last action $(\tilde{s}_t, \tilde{a}_{t-1})$ using multilayer perceptron (MLP) and outputs the reference latent z_t. This paper uses LSTM as the pre-memory module of the student encoder E_{θ_s}, which can memorize and forget proprioceptive observations and past LSTM states M_{t-1}^e, including hidden state h_{t-1}^e and cell state c_{t-1}^e selectively, to obtain the current LSTM states $M_t^e = (h_t^e, c_t^e)$ and the current LSTM output O_t^e. Then O_t^e passes through a three-layer MLP to produce the estimation of the linear velocity $\hat{v}_t$ and the latent representation $\hat{z}_t$. We also adopt a context decoder $D_{\theta_d}(\hat{o}_{t+1} \mid \overline{v}_t, \overline{z}_t, \tilde{o}_t)$ to form the auto-encoder (AE) structure. By reconstructing the subsequent proprioceptive state, the latent representation can

TABLE III
NETWORK ARCHITECTURES

Module	Inputs	Hidden dims	Outputs
E_{θ_t}	$\tilde{s}_t, \tilde{a}_{t-1}$	MLP: [256,128,64]	i_t
E_{θ_s}	$\tilde{o}_t, \tilde{a}_{t-1}, M^e_{t-1}$	LSTM: 256	O^e_t, M^e_t
	O^e_t	MLP: [256,128,64]	$\hat{e}_t, \hat{i}_t$
D_{θ_d}	$\tilde{o}_t, \tilde{a}_{t-1}$	MLP: [256,128,64]	$\hat{o}_{t+1}$
π_θ	$s^a_t, \tilde{a}_{t-1}, M^a_{t-1}$	LSTM: 256	O^a_t, M^a_t
	O^a_t	MLP: [256,128,64]	μ_t
V_ϕ	$\tilde{s}_t, \tilde{a}_{t-1}, M^c_{t-1}$	LSTM: 256	O^c_t, M^c_t
	O^c_t	MLP: [256,128,64]	$V^R_t, V^{C_i}_t$

be enriched with implicit information about the robot state and the surrounding environment. Generally, the total regression loss of AE using mean-squared-error (MSE) can be expressed as:

$$L_{reg} = L_{explict} + L_{implict} + L_{recons} \tag{16}$$

where $L_{explict} = MSE(\hat{e}_t, \tilde{e}_t)$ is the loss of explicit state estimation, $L_{implict} = MSE(\hat{i}_t, i_t)$ is the loss of implicit latent representation, and $L_{recons} = MSE(\hat{o}_{t+1}, \tilde{o}_{t+1})$ is the loss of proprioceptive reconstruction. It is worth mentioning that $\hat{z}_t$ should mimic the aggregated latent z_t to ensure that the student network can eventually adapt to the environment without any guidance from the teacher.

Actor and Multi-Head Critic: We utilize an asymmetric actor-critic framework (A2C) [16] to discover the interplay between actor and critic networks defined as follows. The actor network $\pi_\theta(\mu_t \mid s^a_t, a_{t-1})$ parameterized by θ outputs the mean of target action μ_t where $s^a_t = (\tilde{o}_t, \overline{v}_t, \overline{z}_t)$. $\overline{v}_t$ and $\overline{z}_t$ are the aggregation of the base linear velocity and the latent representation, respectively, defined in Section II-B. The standard deviation of the target action σ_t is also optimized by maximizing the entropy of the Gaussian distribution $\mathcal{N}(\mu_t, \sigma_t)$ that a_t samples from during training. Extending the critic network, the multi-head critic network $V_\phi(V^R_t, V^{C_i}_t \mid s_t, a_{t-1})$ parameterized by ϕ approximates the value function of the reward and costs. Due to the non-negativity of the cost value function, with reference to [26], we append a softplus output layer to the output of the cost part so that the cost return obtains a smaller variance. Unlike the actor network, the critic network uses the full state s_t without any observation noise for a more accurate guidance. The actor and the multi-head critic are both similar to the student encoder structure of the LSTM connected with the MLP, which can help the RL networks to extract sequence information better. More details of the network architectures are shown in Table III.

IV. RESULTS

Our controller is deployed on the Unitree Go1 Edu and our self-developed SDUQuad48, which stands 0.32 m tall with the default joint position and weighs 15 kg. The sensors used on the robot consist of the joint encoder and IMU. The trained encoder and policy network are exported and optimized using ONNX Runtime [33] and reloaded within the finite state machine (FSM), which operates on an onboard Jetson Orin NX 16GB computer, with network inference latency maintained at approximately 1 ms.

A. Comparison and Ablation Experiments

To compare with the proposed methods and explore the necessity of each module in this paper, we set up the following algorithm comparison experiments for the robot.

- **Concurrent** [28]: The framework concurrently trains the explicit state estimator and the policy network based on the historical proprioceptive state.
- **DreamWaQ** [17]: The framework implements explicit estimation of the base linear velocity and implicit inference of privileged state using the context-aided estimator network.
- **Two Stage Teacher-Student (T-S)** [9], [13], [23]: In the first stage, the teacher is trained to use privileged information to instruct the policy network. In the second stage, the student reconstructs the latent space encoded by the teacher and imitates the actions of teacher based on supervised learning.
- **ALARM w/o EN:** The proposed method without empirical normalization.
- **ALARM w/o LSTM-AC**: The proposed method without LSTM Memory for the actor and multi-head critic.
- **ALARM w/o Decoder**: The proposed method without reconstructing the subsequent proprioceptive state.

All of the above methods are performed on the same terrain curriculum and with the same reward function for 10,000 iterations. Due to the mechanism to avoid skill forgetting in terrain courses, directly observing the average terrain level cannot accurately represent the ability of the policy. In the simulation, we set a flag for each agent to indicate whether it has reached the maximum terrain level. For a fair comparison, our method counts only agents with $\beta = 1$. The growth trend of the average value of the flags of 4096 agents during training is shown in Fig. 3. Each curve represents the mean of the experimental results obtained across five different random seeds, with the shaded region around each curve indicating the standard deviation among these results. The figure demonstrates that Concurrent and DreamWaQ, based solely on A2C, exhibit faster exploration in the early stages but ultimately show limited adaptability to the terrain. In the two-stage T-S method, the teacher is directly trained using privileged information, enabling nearly all agents to reach the maximun terrain level after 3000 iterations. When the student trains with the teacher's policy for 7,000 iterations, the performance stabilizes around 80%, which is attributed to distributional shifts between the teacher and student policies. In contrast, ALARM, leveraging an adaptive aggregation mechanism, effectively mitigates this issue, enabling the final policy to closely approximate the teacher's performance, with only a 2% loss in effectiveness. The figure further indicates that ALARM, when applied without the decoder, exhibits

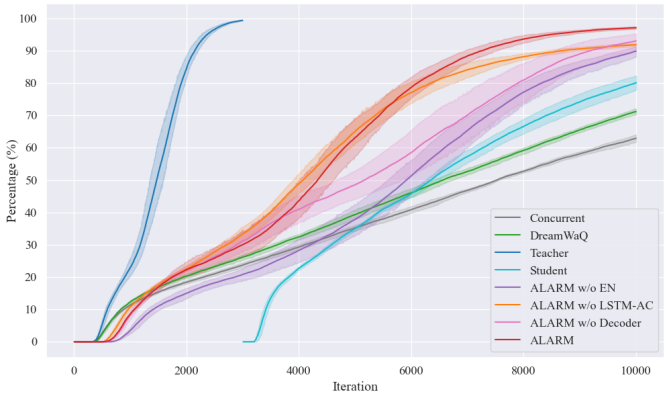

Fig. 3. Percentage of agents that have ever reached the maximum terrain level.

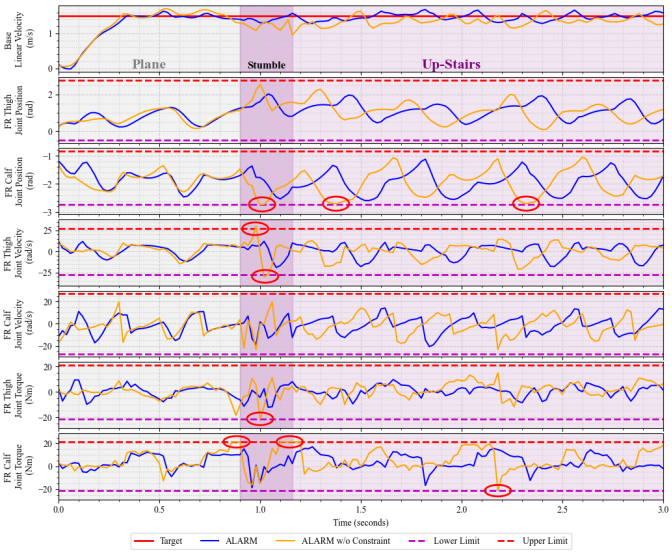

Fig. 4. Real-time data comparison of the joint position, velocity and torque of the front right (FR) thigh and calf when the robot climb over 16 cm steps, tracking the 1.5 m/s forward velocity command. The circled parts of the red ellipse represent the constraint violations in the process of adapting to the terrain change.

instability when training on more complex terrains, resulting in a decline in final policy performance. Additionally, incorporating an LSTM into the AC network contributes to an approximate 6% performance improvement. Incorporating an experience regularization module significantly enhances the training efficiency of our method. In summary, ALARM enables the student to maintain exploration and autonomy during the imitation process, thereby enhancing the overall efficiency and stability of the training.

B. Evaluation of Safe Locomotion

We explored the effectiveness of constraints and NP3O optimization to ensure the safety of the robot when traversing complex terrain. By recording the real-time data of the joint position, velocity, and torque of the front right (FR) thigh and calf when the Go1 traverses 16 cm up-stairs at a forward speed of 1.5 m/s, as shown in Fig. 4, it is intuitively demonstrated whether there are dangerous behaviors exceeding the limits.

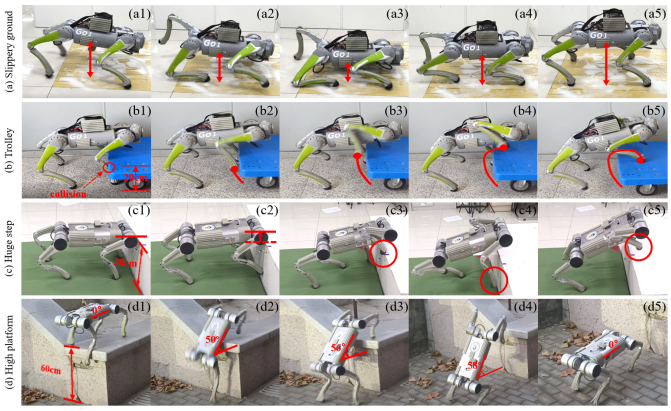

Fig. 5. Our controller exhibits remarkable adaptability on highly challenging terrains such as trolley, huge step and high platform.

The results demonstrate that the ALARM framework, effectively prevents dangerous behaviors that exceed the predefined limits during challenging locomotion tasks. In contrast, the unconstrained variant exhibits frequent violations of these limits, as highlighted by the red circles. These violations not only pose a risk to hardware safety but also indicate instability in locomotion control. The ability of ALARM to maintain stability and safety across joint metrics highlights the efficacy of the NP3O optimization in regulating robot behavior under complex terrains, ensuring a balance between task performance and safety constraints.

C. Evaluation of Robust Locomotion

To demonstrate the robustness of our controller in the real world, we evaluate the robot's ability to traverse some challenging environments, which is detailed bellow.

Slippery tile: We created an extremely low-friction terrain by covering a smooth tile surface with a large amount of lubricating foam. As shown in Fig. 5(a3), the robot's trunk came dangerously close to the ground when its feet experienced severe sliding. Our controller demonstrated the ability to quickly adapt to this significant constraint violation and responded promptly, allowing the robot to a stable state recovery.

Trolley: When attempting to climb onto an unpredicted hollow trolley platform in the simulation, the robot's calf collided with the platform edge, as shown in Fig. 5(b1). By leveraging a combination of encoded historical states and memory, the robot accurately detected this unexpected collision constraint violation and adjusted its locomotion strategy in real-time. It retracted its foot and raised its leg, forming a circular trajectory, as shown in Fig. 5(b5), successfully climbing onto the 20 cm high trolley platform.

Huge step: The robot detected a 30 cm high obstacle using front leg contact and preliminarily estimated its height. It adjusted the base height and swing leg trajectory in an attempt to climb over the obstacle. However, during the first attempt, its front foot failed to grasp the platform edge, resulting in a brief fall. Thanks to the robust controller, the robot reacted quickly and accurately, as illustrated in Fig. 5(c5). It adjusted

its posture and succeeded in overcoming the obstacle on the second attempt.

High platform: We challenged the robot to jump off a 60 cm high platform. As shown in Fig. 5(d2), when the front foot landed, its body tilted sharply. To avoid violating joint and posture constraints, the robot swiftly adjusted its posture, maintaining a safe state, as shown in Fig. 5(d5).

The experimental details showcase the ability of robots to traverse real-world terrains that are significantly more complex than the simulated environments used during training. Leveraging strong generalization capabilities and exceptional robustness of ALARM, the robot achieves agile and safe navigation. This demonstrates the stability and adaptability of our controller in addressing diverse and challenging terrains under dynamic, real-world conditions.

V. Conclusion

In this paper, we propose ALARM, an end-to-end locomotion control framework for legged robots based on imitation learning and reinforcement learning. Through a single training process, it achieves a seamless transition from teacher to student. By introducing NP3O optimization, it effectively and concisely constrains robot behavior, enabling both safe and robust locomotion on complex terrains using only proprioception. Additionally, ALARM facilitates direct transfer from simulation to the real world, featuring low computational resource consumption and fast inference speed. Through comparative and ablation experiments, we demonstrate that our method surpasses current state-of-the-art reinforcement learning controllers in terms of training efficiency and locomotion performance. Our policy has been deployed on various quadrupedal robots, showcasing strong adaptability and resilience in complex environments.

References

[1] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1154–1171, 2021.

[2] Z. Zhu, G. Zhang, Z. Sun, T. Chen, X. Rong, A. Xie, and Y. Li, "Proprioceptive-based whole-body disturbance rejection control for dynamic motions in legged robots," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7703–7710, 2023.

[3] Y. Wang, T. Chen, X. Rong, G. Zhang, Y. Li, and Y. Xin, "Design and control of skater: A wheeled-bipedal robot with high-speed turning robustness and terrain adaptability," *IEEE/ASME Transactions on Mechatronics*, pp. 1–12, 2024.

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

[5] C. Yu and A. Rosendo, "Multi-modal legged locomotion framework with automated residual reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 312–10 319, 2022.

[6] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "Dtc: Deep tracking control," *Science Robotics*, vol. 9, no. 86, p. eadh5401, 2024.

[7] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *6th Annual Conference on Robot Learning*, 2022.

[8] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," 2024.

[9] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[10] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[11] G. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," in *Robotics: Science and Systems*, 2022.

[12] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[13] J. Wu, G. Xin, C. Qi, and Y. Xue, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4975–4982, 2023.

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

[15] Z. Fu, X. Cheng, and D. Pathak, "Deep whole-body control: Learning a unified policy for manipulation and locomotion," in *Conference on Robot Learning (CoRL)*, 2022.

[16] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," in *Robotics*, ser. Robotics: Science and Systems, 2018.

[17] I. M. Aswin 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)*, 2023, pp. 5078–5084.

[18] 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.

[19] D. J. Braun, F. Petit, F. Huber, S. Haddadin, P. Van Der Smagt, A. Albu-Schäffer, and S. Vijayakumar, "Robots driven by compliant actuators: Optimal control under actuation constraints," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1085–1101, 2013.

[20] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *ArXiv*, vol. abs/1909.06586, 2019.

[21] S. Gangapurwala, A. Mitchell, and I. Havoutis, "Guided constrained policy optimization for dynamic quadrupedal robot locomotion," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3642–3649, 2020.

[22] L. Zhang, L. Shen, L. Yang, S.-Y. Chen, B. Yuan, X. Wang, and D. Tao, "Penalized proximal policy optimization for safe reinforcement learning," in *International Joint Conference on Artificial Intelligence*, 2022.

[23] Y. Kim, H. S. Oh, J. H. Lee, J. Choi, G. Ji, M. Jung, D. H. Youm, and J. Hwangbo, "Not only rewards but also constraints: Applications on legged robot locomotion," *IEEE Transactions on Robotics*, vol. 40, pp. 2984–3003, 2023.

[24] E. Chane-Sane, P.-A. Léziart, T. Flayols, O. Stasse, P. Souéres, and N. Mansard, "Cat: Constraints as terminations for legged locomotion reinforcement learning," *ArXiv*, vol. abs/2403.18765, 2024.

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

[26] J. Lee, L. Schroth, V. Klemm, M. Bjelonic, A. Reske, and M. Hutter, "Evaluation of constrained reinforcement learning algorithms for legged locomotion," *arXiv preprint arXiv:2309.15430*, 2023.

[27] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*. PMLR, 2020, pp. 66–75.

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

[29] Z. Wang, W. Wei, R. Yu, J. Wu, and Q. Zhu, "Toward understanding key estimation in learning robust humanoid locomotion," 2024.

[30] A. Warrington, J. W. Lavington, A. Scibior, M. Schmidt, and F. Wood, "Robust asymmetric learning in pomdps," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 013–11 023.

[31] T. Chen, X. Rong, Y. Li, C. Ding, H. Chai, and L. Zhou, "A compliant control method for robust trot motion of hydraulic actuated quadruped robot," *International Journal of Advanced Robotic Systems*, vol. 15, no. 6, p. 1729881418813235, 2018.

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

[33] O. R. developers, "Onnx runtime," https://onnxruntime.ai/, 2021, version: x.y.z.