MorAL: Learning Morphologically Adaptive Locomotion Controller for Quadrupedal Robots on Challenging Terrains

Zeren Luo, *Graduate Student Member, IEEE*, Yinzhao Dong, Xinqi Li, Rui Huang, Zhengjie Shu, Erdong Xiao, *Graduate Student Member, IEEE*, and Peng Lu

***Abstract*—Due to the rapid development of the quadruped robot industry in the past decade, various commercial quadruped robots have emerged with distinct physical attributes. Different from the previous work in which the designed controller is robot-specific, this article proposes a learning-based control framework – MorAL, which is adaptive to different morphologies of quadruped robots and challenging terrains. Our framework concurrently trains the control policy and an adaptive module, which considers the temporal robot states. This module empowers the control policy to implicitly online identify different robot platforms' properties and estimate body velocity. Extensive experiments in the real world and simulation demonstrate that our controller enables robots with significantly different morphology to overcome various indoor and outdoor harsh terrains.**

***Index Terms*—Deep reinforcement learning, legged robots, self-adaptation.**

I. Introduction

In RECENT years, there has been a notable upsurge in the quadrupedal robotics sector, which finds application in diverse fields, including industrial inspection and exploration. Unlike their wheeled counterparts, quadrupedal robots excel at navigating unstructured terrains but are vulnerable to toppling due to their elevated center of gravity [1]. As the availability of legged robots grows, there is a corresponding surge in the demand for controllers that empower these robots to perform a range of valuable tasks. However, the majority of learning-based controllers are tailored to specific robots [2], [3], [4], a process that needs to be repeated for every new robot and can take several weeks or even months to fine-tune hyperparameters. The development of a robust adaptive control policy capable of serving all categories of quadrupedal robots is particularly valuable for unified and quick deployment.

A. Morphological Adaptive Controllers

To facilitate the transfer of a controller across different platforms, the field of robotics traditionally relies on the practice of system identification [5], [6], [7]. These methods typically involve the modeling of dynamics and kinematics based on the system's responses, which can significantly vary from one type of robot to another. However, the modeling demands substantial labor and may necessitate additional computational resources during the deployment [5]. Moreover, the deviation in identified parameters when the robots are subjected to unexpected disturbance will result in undermined performances [8].

An alternative line of research in the robot learning field is transfer learning (TL), in which the controller with the generalization can be deployed on various targeted robots [9], [10], [11]. On the one hand, in the cross-platform transfer tasks, some of the existing TL framework requires clear and laborious decoupling of the tasks and the robot morphologies [12], which undermines the generalizability as different model types can not share the learned knowledge. On the other hand, these researches on TL in robotics are only validated in the simulation while their performance on real robot platforms remains unknown [10], [11]. Only a limited number of studies test morphological transferable controllers on real machines [13], but they only rely on the common domain randomization techniques. On the contrary, our framework establishes a shared module, which can be jointly updated by all robot categories during the training. This module can also circumvent the system identification during real-world deployment and guide the learned policy to adapt varying morphology of real quadrupedal robots.

B. Rough Terrain Locomotion

Compared with exteroceptive sensors, which may not always be reliable, proprioceptive sensors are relatively light and robust. Previous studies also have shown that by combining different proprioception modalities, a quadrupedal robot can learn to traverse various unstructured terrains.

Manuscript received 27 October 2023; accepted 1 March 2024. Date of publication 12 March 2024; date of current version 21 March 2024. This letter was recommended for publication by Associate Editor S. Ha and Editor J. Kober upon evaluation of the reviewers' comments. This work was supported by General Research Fund under Grant 17204222, and in part by the Seed Funding for Collaborative Research and General Funding Scheme-HKU-TCL Joint Research Center for Artificial Intelligence. *(Zeren Luo and Yinzhao Dong contributed equally to this work.) (Corresponding authors: Peng Lu; Yinzhao Dong.)*

The authors are with the Adaptive Robotic Controls Lab (ArcLab), Department of Mechanical Engineering, the University of Hong Kong, Hong Kong 999077, SAR, China (e-mail: zerluo@connect.hku.hk; dongyz@connect.hku.hk; xinqili@connect.hku.hk; u3009750@connect.hku.hk; shuzj@connect.hku.hk; lupengt@hku.hk).

A video summarizing the proposed method, simulation and hardware tests is available at https://youtu.be/EjR2OkiLzTA.

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3375086, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3375086

Pioneering works [14], [15] use model-based control architecture including the foothold and body trajectory optimization to achieve fast quadruped locomotion over rough terrain. Follow-up works use a single policy incorporating a family of locomotion strategies in terms of walking gaits to overcome uneven surfaces [16], [17]. Deep reinforcement learning (DRL) has recently been employed using proprioceptive feedback to realize the blind locomotion on rough terrains and realize zero-shot transfer from simulation to natural environments [2], [4]. Among these DRL works that use the teacher-student paradigm, the integration of privileged information proves to facilitate the robot's acquisition of the desired skill. Nevertheless, this distillation training methodology comes with a notable limitation that the student policy's performance is hard to outperform the teacher policy. Most importantly, the above strategy for challenging terrain is limited to specific robot types, such as ANYmal, Go1, Spot, etc., lacking cross-platform ability.

C. Asymmetric Actor Critic (AAC)

To mitigate the limitation in the distilled way of student policy training, the full state observability can be exploited within a unified actor-critic framework [18], in which the privileged knowledge is incorporated solely into the critic net throughout the training process. This method that uses partial observable state has been successfully deployed on solving manipulation tasks on robot arm [19], multi-fingered robotic hands [20], [21] and multi-agent robotics system [22].

As for the quadrupedal robot research community, AAC enables the control policy to accurately estimate the velocity along with other physical states [23]. The incorporation of the exteroceptive terrain map into the critic enables the actor to infer and traverse uneven terrains effectively [24], [25]. In this study, we further advance this approach by incorporating exteroceptive and morphological knowledge into the critic net. The resulting controller, perceiving only proprioceptive robot state, exhibits adaptability across a wide range of quadrupedal robot models and challenging terrains.

In summary, we propose a **Mor**phologically **A**daptive **L**ocomotion (MorAL) framework for quadrupedal locomotion skills. The controller trained by the framework can be readily applied to a large variety of robot morphologies and enable them to locomote over rough terrain without further fine-tuning. As shown in Fig. 1 and the accompanying video, the MorAL can be directly transferred to 12 mainstream quadrupedal robots, including HyQ, Go1, Spot, ANYmal, etc. The key contributions of this work can be listed as follows:

- In the proposed framework, we train a general DRL-based control policy that uniformly considers the robot's structural variation and terrain diversity. This saves the labor of training a dedicated controller for each specific robot.
- The proposed controller possesses the capability to implicitly identify the morphology of the robot during the deployment, which eliminates the need for the conventional procedure of system identification for the robotics system.
- The MorAL enables all types of robots to robustly conquer challenging terrains. The adaptive module can also be utilized as the robot state estimator that outperforms other existing approaches. The verification of hardware experiments is conducted in various indoor and outdoor harsh environments.

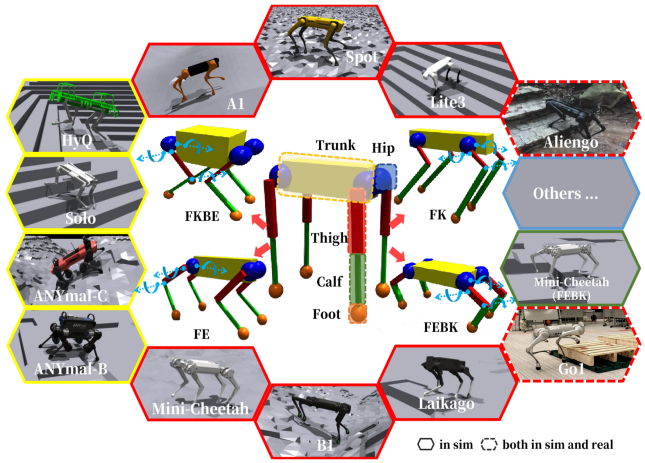

Fig. 1. Four major quadrupedal robot configurations are used for training, including the front-knee back-elbow (FKBE), front-elbow back-knee(FEBK), full-elbow (FE), and full-knee (FK) with distinct joint limit and initial configuration pose. The rotation axis of each joint is also denoted in the figure. The robots belonging to a similar category are highlighted with the same color frame.

II. Method

A. Preliminaries

The blind locomotion controller case is formulated as an infinite-horizon Partially Observable Markov Decision Process (POMDP), which is a framework to model a discrete-time stochastic control process. This is because the terrains are not fully observable without exteroceptive sensors.

The POMDP can be defined by a 7-tuple $\mathcal{M} = \{\mathcal{S}, \Omega, \mathcal{A}, \mathcal{R}, \mathcal{T}, \mathcal{P}, \gamma\}$, where $\mathcal{S}$ is the set of states and $\mathcal{A}$ is the set of actions. At state $\boldsymbol{s}_t \in \mathcal{S}$, the learning agent interacts with the environment with the action $\boldsymbol{a}_t \in \mathcal{A}$ and receives rewards $\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$, leading to the transition of the environment to the next state $\boldsymbol{s}_{t+1}$ with the probability $\mathcal{T}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. Meanwhile, the observation $\boldsymbol{o}_{t+1} \in \Omega$ depends on the new state $\boldsymbol{s}_{t+1}$ and the action $\boldsymbol{a}_t$ with the conditional probability $\mathcal{P}(\boldsymbol{o}_{t+1}|\boldsymbol{s}_{t+1}, \boldsymbol{a}_t)$. DRL problem aims to figure out the optimal policy π^* that maximize the accumulated rewards $J_{\mathcal{M}}(\boldsymbol{\pi})$ of this POMDP with a discount ratio $\gamma \in [0, 1]$, i.e.:

$$J_{\mathcal{M}}(\boldsymbol{\pi}) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}(\gamma^t \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t))\right]. \tag{1}$$

In addition, we also define a temporal observation $\boldsymbol{o}_t^H = [\boldsymbol{o}_t, \boldsymbol{o}_{t-1}, \ldots, \boldsymbol{o}_{t-H}]$ and a privileged morphology observation $\boldsymbol{o}^{mor}$, where $\boldsymbol{o}_t^H$ denotes the history of state and actions over the past H time steps ($H = 5$ in this task), and $\boldsymbol{o}^{mor}$ represents the masses and sizes of the agent's trunk and legs.

B. Generalized Morphological Controller

As shown in Fig. 2, the MorAL framework consists of three sub-networks: the policy net, the value net, and the morph net, which work together to achieve real-time adaptation to different

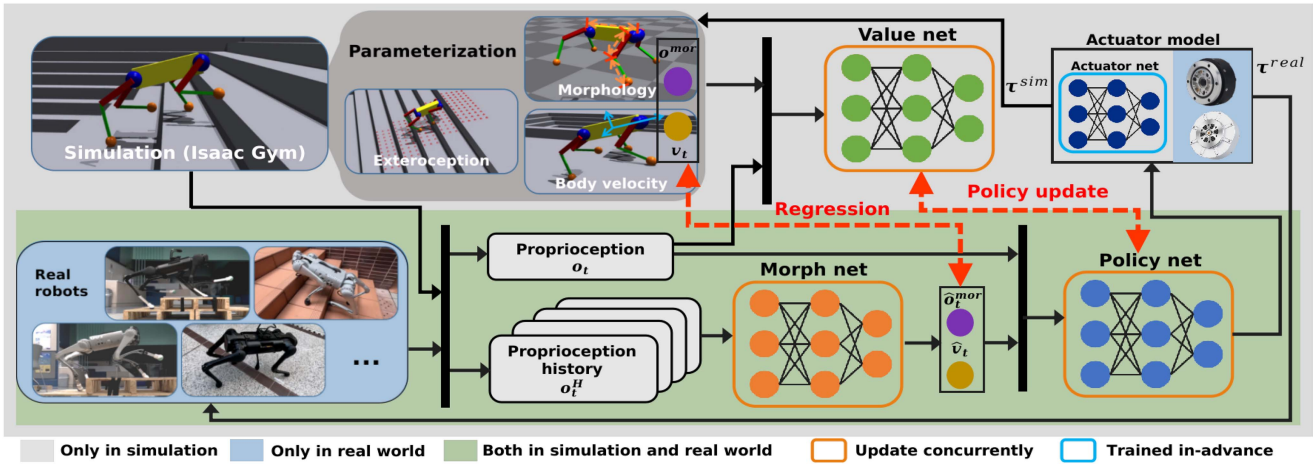

Fig. 2. Schematic of the proposed training methodology. The green blocks represent the modules that exist in both simulation training and real-world deployment. The value net and policy net in the PPO algorithm, together with the morph net that encodes the physical information are updated concurrently (Section II-B). The actuator network is trained offline for the benefit of sim-to-real transfer (Section II-D).

morphologies and terrains. These three networks are trained together via PPO [26] in simulation, and the parameters of the morph net are also updated via a regression algorithm. Next, each part of the MorAL will be discussed in detail.

1) Policy Net: The policy net is applied to infer the action $\boldsymbol{a}_t \in \mathbb{R}^{12}$ by taking the input $\boldsymbol{p}_t$ as the estimated body velocity $\hat{\boldsymbol{v}}_t \in \mathbb{R}^3$, the proprioception $\boldsymbol{o}_t \in \mathbb{R}^{45}$, and the estimation of morphology observation $\hat{\boldsymbol{o}}_t^{mor} \in \mathbb{R}^9$, i.e.

$$\boldsymbol{p}_t = [\hat{\boldsymbol{v}}_t,\ \boldsymbol{o}_t,\ \hat{\boldsymbol{o}}_t^{mor}] \tag{2}$$

where $\hat{\boldsymbol{v}}_t$ and $\hat{\boldsymbol{o}}_t^{mor}$ are both the output of the morph net. $\boldsymbol{o}_t$ can be collected from the proprioceptive sensors, including body angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$, projected gravity $\boldsymbol{g}_t \in \mathbb{R}^3$, body linear velocity command $\boldsymbol{v}_t^* \in \mathbb{R}^3$, joint angles $\boldsymbol{q}_t \in \mathbb{R}^{12}$, joint velocities $\dot{\boldsymbol{q}}_t \in \mathbb{R}^{12}$, and the action of the last step $\boldsymbol{a}_{t-1} \in \mathbb{R}^{12}$, which can be written as:

$$\boldsymbol{o}_t = [\boldsymbol{\omega}_t,\ \boldsymbol{g}_t,\ \boldsymbol{v}_t^*,\ \boldsymbol{q}_t,\ \dot{\boldsymbol{q}}_t,\ \boldsymbol{a}_{t-1}]\,. \tag{3}$$

2) Morph Net: In real-world deployment, reliable and accurate body velocity estimation is crucial for locomotion while cannot be directly acquired through the robot's IMU. Previous works [23], [24] employ a neural network to reason about the body velocity, which can significantly improve the robustness of the controller. In addition, the policy trained by RL cannot be directly transferred among various quadruped robots due to the significant differences and modeling inaccuracy in morphology.

Thus, we use the morph net to estimate body velocity and morphology simultaneously, which takes the $\boldsymbol{o}_t^H$ as input and outputs the estimated body velocity $\hat{\boldsymbol{v}}_t$ and estimated morphology $\hat{\boldsymbol{o}}_t^{mor}$. The morph net is not only trained with supervised learning to reduce the Mean Squared Error (MSE) $loss_{reg}$ between the estimation and the corresponding ground-truth values but also optimized by the policy loss $loss_{\text{policy}}$ from the PPO:

$$\begin{aligned} Loss &= \beta \cdot loss_{reg} + (1-\beta)\cdot loss_{\text{policy}}, \\ loss_{reg} &= MSE(\hat{\boldsymbol{v}}_t,\ \boldsymbol{v}_t) + MSE(\hat{\boldsymbol{o}}_t^{mor},\ \boldsymbol{o}^{mor}). \end{aligned} \tag{4}$$

TABLE I
REWARD TERMS

Term	Reward	Equation	Weight
Task	Lin. velocity tracking	$\text{-}e^{4\|\boldsymbol{v}_{xy}^* - \boldsymbol{v}_{xy}\|^2}$	1.0
	Ang. velocity tracking	$\text{-}e^{4(\omega_z^* - \omega_z)^2}$	0.5
Smoothness	Linear velocity (z)	v_z^2	-2.0
	Angular velocity (xy)	$\|\boldsymbol{w}_{xy}\|^2$	-0.05
	Joint torque	$\|\boldsymbol{\tau}\|^2$	$-e^{-5}$
	Action rate	$\|\boldsymbol{a}_t - \boldsymbol{a}_{t-1}\|^2$	-0.01
	Joint accelerations	$\|\ddot{\boldsymbol{q}}\|^2$	$-2.5e^{-7}$
Safety	collisions	$-n_{collision}$	1.0
Pose	Orientation	$\|\boldsymbol{q}_{xy}\|^2$	-0.2
	Joint motion limit	$\sum_{j=0}^{12}\|\boldsymbol{q}_{t,j} - \mathring{\boldsymbol{q}}_j\|$	-0.02
	Feet air time	$\sum_{f=0}^{4}(t_{air,f} - 0.5)$	1.0

where β is a hyperparameter with a range [0,1] ($\beta = 0.5$ in our case). It is noteworthy that as the gradient of $loss_{\text{policy}}$ is also backpropagated to the morph net, the morph net is guided towards the direction beneficial to the rough terrain locomotion.

3) Value Net: Apart from the $\boldsymbol{o}_t$, another portion of the input of the value net are those that are expensive to obtain or readily deviate induced by the inaccurate measurement. These components involve 1) the exteroceptive information about the terrain, which is the egocentric height map of the robot body $\boldsymbol{h}_t \in \mathbb{R}^{187}$; 2) the physical states including body velocity $\boldsymbol{v}_t \in \mathbb{R}^3$, feet contact states indicated by contact boolean $\boldsymbol{c}_t \in \mathbb{R}^4$, feet height $\boldsymbol{h}_t^{\text{feet}} \in \mathbb{R}^4$, disturbance force in x-y directions $\boldsymbol{f}_t \in \mathbb{R}^2$, and properties parameters including bodies friction coefficient $\boldsymbol{\mu} \in \mathbb{R}^1$, proportional–derivative (PD) controller gains of each joint and PD gains' scale $\boldsymbol{k}_{PD} \in \mathbb{R}^{14}$; 3) the privileged morphology observation $\boldsymbol{o}^{mor} \in \mathbb{R}^9$ of rigid bodies, which are indicated by the bold value in Table II. Thus, the input of the value net can be organized in the following form:

$$\boldsymbol{s}_t = \left[\boldsymbol{v}_t,\ \boldsymbol{o}_t,\ \boldsymbol{o}^{mor},\ \boldsymbol{h}_t,\ \boldsymbol{h}_t^{\text{feet}},\ \boldsymbol{c}_t,\ \boldsymbol{f}_t,\ \boldsymbol{\mu},\ \boldsymbol{k}_{PD}\right]. \tag{5}$$

4) Action Space: The action $\boldsymbol{a}_t$ represents the desired increment of the joint angle w.r.t the initial pose $\mathring{\boldsymbol{q}}$, i.e. $\boldsymbol{a}_t = \boldsymbol{q}_t^* - \mathring{\boldsymbol{q}}$.

TABLE II
RANDOMIZATION RANGE OF PARAMETERS. THE BOLD VALUES IN MORPH ROW FORM $\boldsymbol{o}^{mor}$

Type	Parameters	Range	Parameters	Range
Morph	**Trunk Mass**	[4.00, 28.00)	**Trunk Length**	[0.37, 0.65)
	Trunk Width	[0.09, 0.30)	**Trunk Height**	[0.11, 0.19)
	Hip Mass	[0.30, 0.69)	Hip Length	[0.03, 0.05)
	Thigh Mass	[0.60, 4.00)	**Thigh Length**	[0.21, 0.35)
	Thigh Width	[0.02, 0.04)	Thigh Height	[0.03, 0.05)
	Calf Mass	[0.10, 0.86)	**Calf Length**	[0.21, 0.35)
	Calf Width	[0.016, 0.020)	Calf Height	[0.013, 0.019)
Others	$\boldsymbol{K}_p$	[20, 80]	$\boldsymbol{K}_d$	[0.6, 2.0]
	$\boldsymbol{\alpha}$	[0.9, 1.1)	$\Delta \boldsymbol{M}$	[-0.02, 0.02)
	Payload	[-2.0, 2.0)	Motor frictions	[0.2, 1.25)

The final desired angle $\boldsymbol{q}_t^*$ is tracked by the torque generated by the joint-level PD controller of the joint-level actuation module, i.e. $\boldsymbol{\tau} = \boldsymbol{K}_p \cdot (\boldsymbol{q}_t^* - \boldsymbol{q}_t) + \boldsymbol{K}_d \cdot (-\dot{\boldsymbol{q}}_t)$. To mitigate the reality gap in real-world deployment, this actuation module on certain robot models is replaced by the actuator network trained with data collected from real machines (Section II-D).

5) Reward Function: The reward functions are inherited from the previous works [23], [24], [27], as shown in Table I. The reward functions regarding task, smoothness, safety, and pose are used to track the commanded velocity, penalize the unsmoothness of robot locomotion, avoid collisions with the environment, and constrain joint motion.

C. Morphology Generation

To train generalizable strategies, we generate a diverse array of robot morphologies during the initialization of the simulated environment. In essence, most quadrupeds share similar body structures, characterized by a trunk and four limbs. Meanwhile, each leg can be naturally decomposed into three links of hip, thigh, and calf, interconnected by two joints. Inspired by these, we designed a simplified template for a quadruped robot, whose critical links (like trunk, thigh, and calf) are composed of a series of rectangular boxes, as shown in Fig. 1.

By adjusting the initial posture $\dot{\boldsymbol{q}}$ and the rotation axis of specific joints from the nominal template, we can generate four major configurations of quadruped (Fig. 1), namely front-knee back-elbow (FKBE), front-elbow back-knee (FEBK), full-elbow (FE), and full-knee (FK).

In addition, the parameters of critical links (such as mass and size) are modified with the detailed range outlined in Table II. It is worthwhile to mention that heavier robots with greater mass require greater actuation force to complete the same locomotion task. To achieve this, the PD gain of robots is scaled by hyper-parameters η, which is determined by the ratio of the overall mass m of the robot to the nominal mass m_b, i.e. $\eta = f(\frac{m}{m_b})$. The nominal mass m_b is set as 8.03 kg (the lightest value among all types) and the basis PD gain is set as 20 and 0.6. η is specified as the third power function of $\frac{m}{m_b}$, as the required actuation force sharply increases as robot mass grows:

$$\eta = a \cdot \left(\frac{m}{m_b}\right)^3 + b \cdot \left(\frac{m}{m_b}\right)^2 + c \cdot \left(\frac{m}{m_b}\right) + d \qquad (6)$$

where parameters $a = 0.03499$, $b = -0.3338$, $c = 1.382$, and $d = -0.1001$ are determined by the polynomial interpolation using a few representative points.

The MorAL synchronously trains the randomly generated quadruped robots, and this controller can extend to new robots that are not observed during the training process (Section III-C).

D. Actuator Modeling for Sim-to-Real Transfer

The simulation-trained controllers often cannot be directly transferred to real robots due to the sim-to-real discrepancies induced by the distinction in the actuator's nonlinear properties, e.g. quasi-direct drive actuators (QDD) on Mini-cheetah, Spot, and Unitree robots, series elastic actuators (SEA) on ANYmal and hydraulic actuators on HyQ.

To solve this problem, we train an actuator network $\boldsymbol{E}_A(\cdot)$ to capture the nonlinear correlation between the PD error and the torque [28]. The pre-trained actuator network is used to generate the torque in simulation $\boldsymbol{\tau}^{\text{sim}}$ with the input being the joint state ($\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$) of the current step and the previous two steps. The predicted torque at a given time step t can be obtained as:

$$\boldsymbol{\tau}_t = \boldsymbol{E}_A\left([\Delta \boldsymbol{q} + \Delta \boldsymbol{M}]_{t-2\delta t:t}\,,\ \dot{\boldsymbol{q}}_{t-2\delta t:t}\right) \qquad (7)$$

where δt is the control interval in simulation (5 ms in our case), $\Delta \boldsymbol{q}_t$ is the position error, namely the difference between the desired position $\boldsymbol{q}_t^*$ and the current position $\boldsymbol{q}_t$, and $\Delta \boldsymbol{M}$ is the motor offset acting as noise to the $\Delta \boldsymbol{q}_t$ measurement. Finally, considering the latency, $\boldsymbol{\tau}^{\text{sim}}$ is taken as the torque at the previous ΔT second (measured from the real actuator and set as 12 ms), i.e. $\boldsymbol{\tau}^{\text{sim}} = \boldsymbol{\tau}_{t-\Delta T}$.

In order to train $\boldsymbol{E}_A$, we collect the data from all the 12 actuators simultaneously and it is carried out on two different types of actuators mounted on Unitree Go1 and Unitree Aliengo.[1] The robots are commanded to track a series of random trajectories with different body heights, which aims to cover the overall actuator operation range.

III. EXPERIMENTAL RESULTS AND DISCUSSION

A. Implementation Details

1) Simulation: We train 4096 agents with different morphologies in parallel on the Isaac Gym simulator [29] for 6000 episodes. An episode is terminated and reset under specific termination criteria, which include the robot's trunk or hip collisions with the ground, or being trapped for a long period of time. We utilized a game-inspired curriculum [30] to ensure progressive locomotion policy learning over difficult terrains, which consist of smooth, rough, discretized, and stair terrains with five levels.

As discussed in Section II-C, different morphologies can be generated by randomizing the selected parameters within a predefined range (Table II). In addition, the critical parameters of the robots, such as motor frictions, PD controller gains,

[1]The comparison of the performance between the policies trained with and without the actuator network is presented in the linked video, which demonstrates its efficacy to bridge the sim-to-real gap.

TABLE III
DIFFERENCE BETWEEN REAL-WORLD ROBOTS TYPES. GO1-L = GO1 LONG; GO1-F = GO1 FAT; AL = ALIENGO; AL-L = ALIENGO LONG; AL-F: ALIENGO FAT

Robot	Mass $[kg]$				Link size $[m]$		
	Total	Trunk	Thigh	Calf	Trunk (x, z)	Thigh	Calf
Go1	12.05	5.20	0.39	0.13	(0.39, 0.09)	0.21	0.21
Go1-L	12.49	5.20	0.39	0.24	(0.39, 0.09)	0.21	0.33
Go1-F	17.46	7.73	0.82	0.42	(0.39, 0.17)	0.21	0.21
Al	22.90	9.04	0.64	0.22	(0.66, 0.11)	0.25	0.25
Al-L	23.42	9.04	0.64	0.35	(0.66, 0.11)	0.25	0.40
Al-F	32.47	14.29	1.18	0.76	(0.66, 0.22)	0.25	0.25

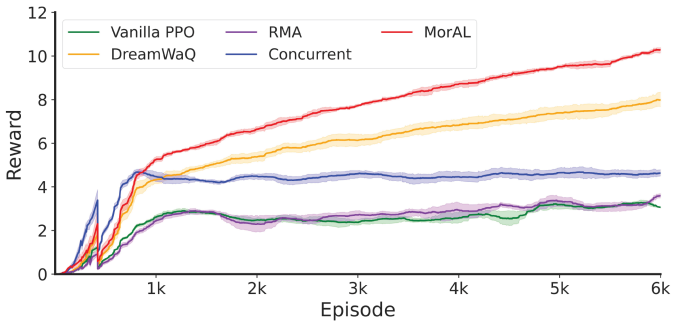

Fig. 3. Learning performance of five different controllers over 4096 different morphologies. The results shown are obtained from 6 different random seeds.

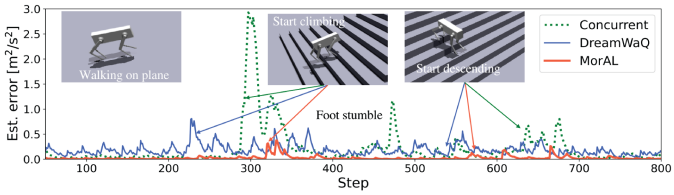

Fig. 4. Squared velocity estimation error of MorAL, Concurrent, and DreamWaQ.

motor strength $\boldsymbol{\alpha}$, and motor offset $\Delta \boldsymbol{M}$, are randomized at the initialization stage.

The policy net and value net are both 4-layer multi-layer perception (MLP) and the architecture of these two MLPs are $[512, 256, 128, 12]$ and $[512, 256, 128, 1]$, respectively. Both of them use Exponential Linear Units (ELU) as the activation in between hidden layers. The architecture of morph net is $[258, 128, 12]$, and it uses Rectified Linear Unit (ReLU) as the activation. The entire training is performed on a desktop PC with an Intel(R) Xeon(R) Gold 6226R CPU @ 2.90 GHz, 32 GB RAM, and an NVIDIA RTX 3090 GPU. Training of the MorAL algorithm costs approximately four hours.

2) Hardware: We validate MorAL's policy on different types of real-world quadrupedal robots with distinct physical attributes (Section III-D $\sim$ F). Their major difference can be parameterized in Table III. According to the definition in Section II-C, $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ of these robots can be calculated with their respective total weights. The observation is computed from the sensor at 250 Hz and the control policy including the policy net and the morph net is executed at 125 Hz. The deployed policy is exported from Isaac Gym and then reconstructed via Libtorch (C++ Distributions of PyTorch).

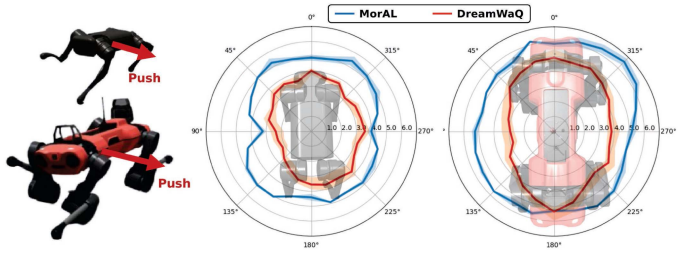

Fig. 5. Push A1 and ANYmal C from different directions and record the maximum velocities that lead to the fall-over. The shaded area represents the standard deviation of different trials at a specific angle.

TABLE IV
LEARNING PERFORMANCE OF DIFFERENT MORPH PARAMETERS

Controller	Reward	Length	Tracking Error $[m/s]$		
			V_x	V_y	ω_z
MorAL	**12.17**	**873**	**0.025**	**0.031**	**0.088**
w/o all estimator	8.76	803	0.094	0.260	0.118
w/o size estimator	11.78	863	0.035	0.050	0.102
w/o mass estimator	11.67	849	0.027	0.049	**0.088**
w/o mass and size estimator	11.50	844	0.029	0.075	0.116

The bold values indicates the best performing values among these ablated controllers.

B. Compared Controllers

The comparative evaluation is performed among the following locomotion controllers that only use proprioception:

- *Vanilla PPO [27]:* The policy is trained without any privileged knowledge about the environment.
- *RMA [4]:* A teacher-student training paradigm with an implicit environmental encoder. The learned controller uses a 1D Convolutional Neural Network (CNN) to generate the features of the environment adaptively.
- *Concurrent [23]:* The policy is trained concurrently with an MLP that explicitly estimates the body states, without providing the terrain information.
- *DreamWaQ [24]:* The policy is trained concurrently with an autoencoder estimating the body velocity and a context vector, without the morphology knowledge.

Fig. 3 illustrates the learning performance of five different controllers in terms of the average rewards. It indicates that MorAL is the most efficient controller in this multi-morphology task. Compared with Vanilla PPO and RMA, the learning performance of Concurrent, DreamWaQ, and MorAL has significant improvement because they all contain a module to estimate the body velocity. However, Concurrent has poor learning performance in the later stages due to the lack of privileged information on terrain height. MorAL consistently shows superior performance after certain iterations when the learning agent encounters more difficult terrains. Please also see the letter's homepage https://arclab-hku.github.io/MorAL_Quadruped_Robots/.

We also evaluate the performance of morph net in terms of velocity estimation by comparing it with Concurrent and DreamWaQ, especially on rough terrain. The robots are commanded to traverse a long path containing flat, upward stairs, and downward stairs. As is demonstrated in Fig. 4, morph net realizes the smallest estimation error and it shows more evident superiority when traversing the stairs. This is attributed to the inclusion of the morphological prior knowledge within the

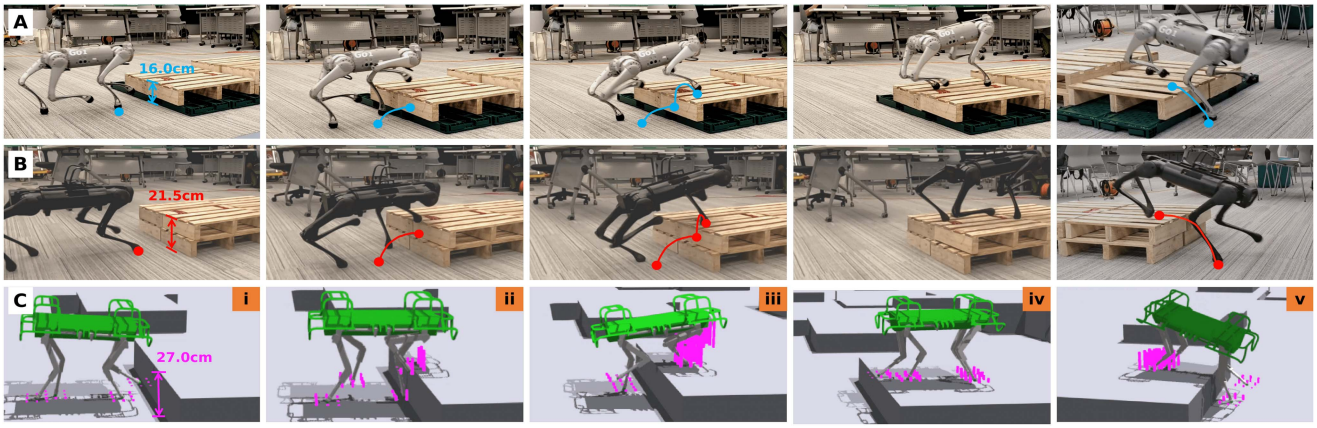

Fig. 6. Go1 (A), Aliengo (B) and HyQ (C) climb up and down obstacles with their perceived height visualized (taking HyQ as an example, more in the linked video). Note that the heights of the bars in (C) indicate the terrain information surrounding the feet. (A) and (B) depict the trajectories of the front-right foot that collides with the obstacle.

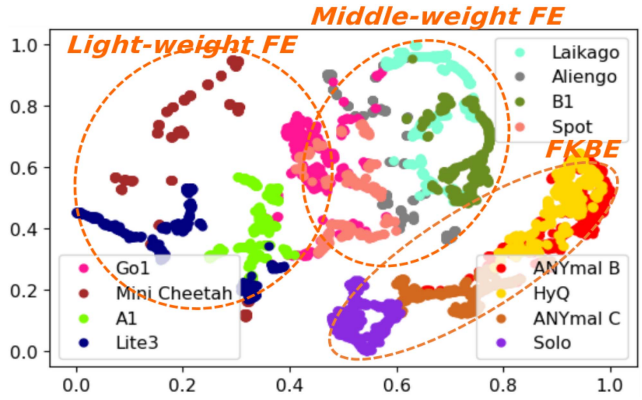

Fig. 7. t-SNE plot of the encoded $\hat{m}_t$, which implicitly classifies the morphologies of the quadrupedal robots.

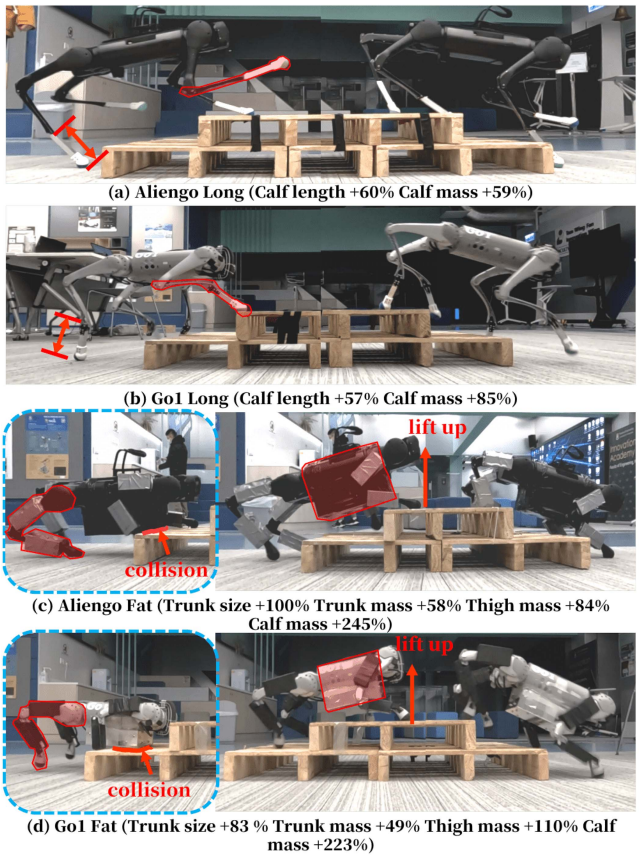

Fig. 8. Robot morphology varies in a wide range, while the controller implicitly identifies those differences via proprioception and adjusts the action accordingly (The number represents the varied percentage compared to its nominal value).

morph net, which facilitates the control policy to reason about the state of an arbitrary robot type.

Among all controllers, only MorAL and DreamWaQ can accomplish most of the uneven terrains (Fig. 3), so we further compare the robustness by pushing a robot from different directions. Fig. 5 shows the maximum push velocities that the robot can withstand without collapsing. It can be seen that MorAL outperforms the DreamWaQ by withstanding higher push velocities from omni-direction. This also indicates the impact of the presence of ground-truth morphology during the training.

By excluding one or more estimated states from the output of morph net, we can obtain four new ablated controllers (Table IV). The policy trained with all components reaches the highest converged reward, episode length, and command tracking performance compared with other ablated policies. In comparison to the policy trained without any estimator, the remaining three ablated policies (w/o size, w/o mass Estimator, and w/o mass and size Estimator) still exhibit significant improvement.

C. Implicit Classification

In order to visualize the capabilities of our controller in implicitly identifying specific robot types, we conducted an experiment involving 12 distinct commercial robots. These robots are commanded to move forward at a consistent speed (0.7 m/s) and we record the $\hat{o}_t^{mor}$ for 400 steps. Next, we use the t-SNE (t-distributed Stochastic Neighbor Embedding) technique to categorize these points (Fig. 7).

It is evident that $\hat{o}_t^{mor}$ can be classified into different regions based on their prominent features. The robots with FKBE

(a) Outdoor experiments are carried out for three routes in the HKU campus, and different routes are featured with different types of terrains. Our controller can notably conquer all these terrains with high robustness.

(b) The indoor environment experiments that are also arranged in various extreme terrains (We refer the reader to the linked video for full details).

Fig. 9. Robustness test for distinctive real robots in versatile challenging terrains. (a) Outdoor experiments are carried out for three routes in the HKU campus, and different routes are featured with different types of terrains. Our controller can notably conquer all these terrains with high robustness. (b) The indoor environment experiments that are also arranged in various extreme terrains (We refer the reader to the linked video for full details).

morphology such as ANYmal and HyQ are clustered far away from those with FE morphology. MorAL can even identify the individuals within the FE region according to their weights. It is worthwhile to note that this adaptive module is able to identify the robot that is not encountered during training, e.g. Solo whose weight (2.5 kg) is much smaller than the minimum of the training range.

D. Attribute Analysis for Rough-Terrain

We repetitively test this controller on a variety of quadrupedal robots in the real world and simulations (Fig. 6). The controller equips any robot with the capability to traverse challenging terrains effectively. To further investigate the interior mechanism, a decoder net $\boldsymbol{E}_\varphi$ is trained to predict the perceived height of 9 egocentric points around each foot $\hat{\boldsymbol{m}}_\varphi^{\text{feet}} \in \mathbb{R}^{36}$ using the latent vector $\hat{\boldsymbol{v}}_t$ and $\hat{\boldsymbol{o}}_t^{mor}$. The $\boldsymbol{E}_\varphi$ is updated with the MSE loss calculated by the difference between the predicted perceived height map $\hat{\boldsymbol{m}}_\varphi^{\text{feet}}$ and its ground truth.

By visualizing decoder $\hat{\boldsymbol{m}}_\varphi^{\text{feet}}$, we can see that when a foot collides with the obstacles and gets tripped, the predicted height of the targeted points at the front of the collided foot progressively increases (Fig. 6(c)(ii)). It is noteworthy that once finished climbing up the obstacle, the perceived height reverted back to the same scenarios as the one on the flat terrain (Fig. 6(c)(iv)). $\boldsymbol{m}_\varphi^{\text{feet}}$ manifests difference only when there is a relatively large difference in the elevation of the feet, such as climbing up and climbing down.

This result strongly indicates that even though the preconception $\boldsymbol{o}_t$ does not involve any information regarding the foothold terrain map, the encoded variables $\hat{\boldsymbol{v}}_t$ and $\hat{\boldsymbol{o}}_t^{mor}$ are able to implicitly perceive and reconstruct this information and adjust the action by modifying their values. This capability can be attributed to the optimization pattern of morph net, in which it is updated by the gradient of both $loss_{reg}$ and $loss_{\text{policy}}$. This process facilitates the robot to better estimate and make inferences about robot morphology and the terrain.

E. Generalization Across Morphologies

To further validate the controller's capability to adapt itself to various morphological changes, a series of robots with distinct lengths, weights, and sizes are tested with the details presented in Table III.

As showcased in Fig. 8, our generalized controller shows adaptability across a broad range of limb and body masses and sizes (up to 245%). The policies online identify the robot morphology exclusively from historical proprioceptive observations. Next, to overcome the rough terrains, the robot either elevates the foot to a height adaptive to its leg length (Go1 / Aliengo Long in Fig. 8(a) and (b)) or empowers the motor with increased actuation suitable for its increased gravity (Go1 / Aliengo Fat in Fig. 8(c) and (d)). Notably, in the case where the torso easily collides with the obstacle, the controller identifies this collision intelligently and learns to lift up the body height, ensuring the bottom surface of the torso is above the obstacle.[2]

[2]We refer the reader to the linked video for viewing two more morphological variation experiments and the controller's adaptability to the dynamic morphology changes.

F. Robust Locomotion Over Harsh Terrains

Extensive tests are carried out on a wide range of terrains in outdoor environments, as demonstrated in Fig. 8 and the linked video. The outdoor experiments consist of three different routes including rugged hills, stairs, slippery ground, and high bushes with hidden obstacles (Fig. 8(a)). Moreover, the indoor experiments include a highly unstable seesaw, deformable foam, stacked wood blocks, etc (Fig. 9(b). These environments even contain the terrain properties that the robots are unseen during training.

The above tests are successfully repeated in Go1 and Aliengo with different physical attributes because of the implicit identification module of the proposed methods (Section III-C). Regarding the step traversing task, our controller even outperforms the manufacturer's dedicated stair-mode which is specially designed based on the machine's limit. In contrast, our controller empowers the robot to traverse higher obstacles beyond the dedicated stair mode limit, and the traversing can occur in any direction such as sideway on the obstacles.

A hiking route in Lung Fu Shan Country Park is even conducted to evaluate the controller's robustness (Yellow curve in Fig. 9(a)). Accomplishing this hiking requires the robot to traverse soft gravel, dense woods, slippery stone roads, etc. Notably, the robot finishes this long path without a single fall.

IV. Conclusion

Our work demonstrates that a controller can realize remarkable generalizability among quadrupedal robots. Specifically, it equips a diverse array of quadrupedal robots, each with distinctive morphologies, to effectively traverse an extensive variety of challenging terrains. The adaptive module, known as the 'morph net' within our framework, ensures that the learned policy can perform system identification for actual robots and accurately estimate the body velocity.

A natural extension of this blind locomotion is to incorporate exteroceptive perception sensors such as LiDAR or stereo cameras. Besides, the sim-to-real solution depends on the pre-trained neural networks, a promising direction for further exploration is leveraging real machine data during the training process to narrow the gap with reality.

References

[1] P. Biswal and P. K. Mohanty, "Development of quadruped walking robots: A review," *Ain Shams Eng. J.*, vol. 12, no. 2, pp. 2017–2031, 2021.

[2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Sci. Robot.*, vol. 5, no. 47, 2020, Art. no. eabc5986.

[3] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "RLOC: Terrain-aware legged locomotion using reinforcement learning and optimal control," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 2908–2927, Oct. 2022.

[4] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid motor adaptation for legged robots," in *Proc. Robot.: Sci. Syst.*, 2021.

[5] S. Lallée et al., "Towards a platform-independent cooperative human robot interaction system: III an architecture for learning and executing actions and shared plans," *IEEE Trans. Auton. Ment. Develop.*, vol. 4, no. 3, pp. 239–253, Sep. 2012.

[6] G. Zhao, P. Zhang, G. Ma, and W. Xiao, "System identification of the nonlinear residual errors of an industrial robot using massive measurements," *Robot. Comput.- Integr. Manuf.*, vol. 59, pp. 104–114, 2019.

[7] Z. Luo, E. Xiao, and P. Lu, "FT-Net: Learning failure recovery and fault-tolerant locomotion for quadruped robots," *IEEE Robot. Automat. Lett.*, vol. 8, no. 12, pp. 8414–8421, Dec. 2023.

[8] J. Wu, J. Wang, and Z. You, "An overview of dynamic parameter identification of robots," *Robot. Comput.- Integr. Manuf.*, vol. 26, no. 5, pp. 414–419, 2010.

[9] A. Gupta, C. Devin, Y. Liu, P. Abbeel, and S. Levine, "Learning invariant feature spaces to transfer skills with reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2017.

[10] W. Huang, I. Mordatch, and D. Pathak, "One policy to control them all: Shared modular policies for agent-agnostic control," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4455–4464.

[11] T. Chen, A. Murali, and A. Gupta, "Hardware conditioned policies for multi-robot transfer learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9333–9344.

[12] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine, "Learning modular neural network policies for multi-task and multi-robot transfer," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 2169–2176.

[13] G. Feng et al., "GENLOCO: Generalized locomotion controllers for quadrupedal robots," in *Proc. Conf. Robot Learn.*, 2023, pp. 1893–1903.

[14] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *Int. J. Robot. Res.*, vol. 30, no. 2, pp. 236–258, 2011.

[15] R. Orsolino et al., "Feasible region: An actuation-aware extension of the support region," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1239–1255, Aug. 2020.

[16] G. B. Margolis and P Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Proc. Conf. Robot Learn.*, 2023, pp. 22–31.

[17] D. Kang, J. Cheng, M. Zamora, F. Zargarbashi, and S. Coros, "RL model-based control: Using on-demand optimal control to learn versatile legged locomotion," *IEEE Robot. Automat. Lett.*, vol. 8, no. 10, pp. 6619–6626, Oct. 2023.

[18] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1008–1014.

[19] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," in *Proc. Robot.: Sci. Syst.*, 2018.

[20] I. Akkaya et al., "Solving rubik's cube with a robot hand," 2019, *arXiv:1910.07113*.

[21] O. M. Andrychowicz et al., "Learning dexterous in-hand manipulation," *Int. J. Robot. Res.*, vol. 39, no. 1, pp. 3–20, 2020.

[22] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 7234–7284, 2020.

[23] 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 Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4630–4637, Apr. 2022.

[24] I. M. A. Nahrendra, B. Yu, and H. Myung, "DreamWAQ: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 5078–5084.

[25] J. Wu, G. Xin, C. Qi, and Y. Xue, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robot. Automat. Lett.*, vol. 8, no. 8, pp. 4975–4982, Aug. 2023.

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

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

[28] J. Hwangbo et al., "Learning agile and dynamic motor skills for legged robots," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaau5872.

[29] V. Makoviychuk et al., "Isaac GYM: High performance GPU-based physics simulation for robot learning," in *Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track (Round 2)*, 2021.

[30] X. Wang, Y. Chen, and W. Zhu, "A survey on curriculum learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4555–4576, Sep. 2022.