Learning Complex Motor Skills for Legged Robot Fall Recovery

Chuanyu Yang[1*], Can Pu[1*†], Guiyang Xin[2], Jie Zhang[3], and Zhibin Li[4]

***Abstract*—Falling is inevitable for legged robots in challenging real-world scenarios, where environments are unstructured and situations are unpredictable, such as uneven terrain in the wild. Hence, to recover from falls and achieve all-terrain traversability, it is essential for intelligent robots to possess the complex motor skills required to resume operation. To go beyond the limitation of handcrafted control, we investigated a deep reinforcement learning approach to learn generalized feedback-control policies for fall recovery that are robust to external disturbances. We proposed a design guideline for selecting key states for initialization, including a comparison to the random state initialization. The proposed learning-based pipeline is applicable to different robot models and their corner cases, including both small-/large-size bipeds and quadrupeds. Further, we show that the learned fall recovery policies are hardware-feasible and can be implemented on real robots.**

***Index Terms*—Machine Learning for Robot Control, Reinforcement Learning, Sensorimotor Learning, Legged Robots**

I. Introduction

FAILURE-RESILIENT locomotion is crucial for the mission success of legged robots, including humanoid and quadruped robots, in unstructured environments. When deploying those robots in real-world unstructured applications, situations are so unpredictable that falling becomes inevitable. When a robot falls, it is important for it to recover back to a canonical operational state and continue its tasks.

Humans and animals are remarkably resilient to falls, as they have the ability to recover from various fall postures, being good inspirations for designing controllers for fall recovery maneuvers. Previously, fall recovery controllers for legged robots were constructed by handcrafting trajectories that resemble a human or animal fall recovery maneuver through heuristics, which is labor intensive [1],[2],[3]. Other methods predict falls [4] or automate the process by calculating the trajectory depending on the specific falling posture offline [5], [6]. These automated approaches are able to operate under a wider range of fall postures compared to heuristic approaches. However, the nature of such offline planning is not event-based and thus lacks the real-time responsiveness that is critical to react to external disturbances.

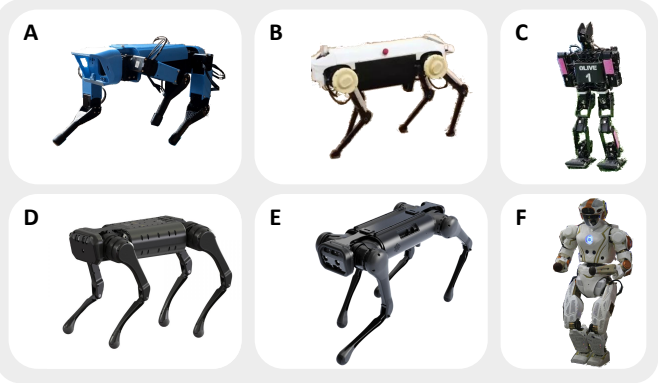

Fig. 1. Successful learning of fall recovery benchmarked on six different types of small-/large-size legged robots: (A) Spotmicro; (B) Jueying Pro; (C) Sigmaban; (D) A1; (E) B1; (F) Valkyrie.

To overcome the limits of laborious handcrafting trajectories, optimization-based approaches can compute feasible solutions for fall recovery without direct manual handcrafting of trajectories [7], [8]. However, they still demand a large amount of computational time due to complex robot dynamics and uncountable possibilities of a mixture of continuous and discrete whole-body contacts. This paradigm relies on explicit specification of contacts, either manually or automatically, and requires advanced optimization schemes, in which computational power and time grow exponentially when complexity scales up, being too slow for real-time solutions and making closed-loop control impractical for robot fall recovery. There exists contact-implicit trajectory optimization approaches that do not require a priori specification of contact sequence[9], [10], [11], [12]. To the best of our knowledge, contact implicit optimization has not yet been implemented to achieve fall recovery for legged robots.

An alternative for obtaining fall recovery motions is model-free reinforcement learning (RL), where an agent interacts with its environment and learns the control policy through a process of trial and error. The major advantages of using RL are: it requires less prior knowledge from human experts

Manuscript received: November, 22, 2022; Revised February, 28, 2023; Accepted May, 1, 2023.

This paper was recommended for publication by Jens Kober upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Shenzhen Amigaga Technology Co. Ltd. under the Gagabot2022 project (Grant Agreement No. P987001), by the Human Resources and Social Security Administration of Shenzhen Municipality under Overseas High-Caliber Personnel project (Grant NO. 202102222X, Grant NO. 202107124X) and by Human Resources Bureau of Shenzhen Baoan District under High-Level Talents in Shenzhen Baoan project (Grant No. 20210400X, Grant No. 20210402X). For access of the released code, please contact AI@amigaga.com.

* These authors contributed equally to this work.

† Corresponding author. Email: can.pu@amigaga.com

[1]Chuanyu Yang and Can Pu are with Shenzhen Amigaga Technology Co Ltd. {chuanyu.yang, can.pu}@amigaga.com

[2]Guiyang Xin is with the School of Optoelectronic Engineering and Instrumentation Science, Dalian University of Technology, CN.

[3]Jie Zhang is with School of Automation and Electrical Engineering, University of Science and Technology Beijing, CN.

[4]Zhibin Li is with the Department of Computer Science, University College London, UK.

Digital Object Identifier (DOI): see top of this page.

and is less labor intensive compared to manual handcrafting; the trained neural network is a feedback policy that can fast compute actions in real-time, compared to the optimization-based methods. Deep Reinforcement Learning (DRL) has shown its successful use for learning fall recovery policies in both simulation and real-world robots [13], [14].

This work has developed a DRL-based framework to solve the problem of learning fall recovery control policies for legged robots. We also proposed a contact transitional graph to represent the possible transitions between different contact configurations during a long sequence of fall recovery motions. Key postures within the graph are used to initialize the robot during training. Such initialization of the robot state narrows down the sample distribution in the solution space during exploration, allowing successful learning of fall recovery policies. Our initialization approach enables effective learning of fall recovery for humanoids and speeds up the learning for quadrupeds. Our framework is able to: (i) generate necessary movements in real-time and recover to an upright standing posture given any initial fall configurations, while requiring minimal human design efforts; (ii) respond to external disturbances in a robust manner and adapt to unknown irregular terrains not seen during training; (iii) generalize across various quadruped and humanoid robots of different sizes and morphologies (Fig. 1).

The contributions of this work are summarised as follows:

1) We proposed a guideline based on ground contact patterns to design key poses for the state initialization to facilitate the RL exploration;
2) We developed a generic fall recovery learning framework that can generalize across quadruped and humanoid robots of various sizes and morphologies;
3) The framework successfully learned fall recovery policies that are responsive and robust to changes in friction, external pushes, and irregular terrains;
4) We validated the feasibility and effectiveness of the learning framework with the successful implementation on a real quadruped robot.

II. Related Work

For humanoids, getting up and standing on two feet is not an easy task, as it involves a sequence of using multiple contact points with the ground. Such scenarios are difficult to model and thus impose many challenges for optimization-based controllers. One common approach to achieve fall recovery for humanoid robots is to handcraft a standing motion that imitates that of humans. Stückler et al. designed a controller for standing up by scripting the target joint angles of the entire trajectory of the standing routine manually [3]. Kanehiro et al. designed a controller for fall recovery for the HRP-P2 robot by constructing a graph consisting of the key contact states within the standing motion with a Zero Moment Point (ZMP)-based controller for the transitions between contact states [2].

Compared to humanoid robots, quadrupedal robots are more stable and are less prone to falling failures that will render the robot inoperable. Nevertheless, there are still quite a few papers that have tackled fall recovery in quadrupedal robots. HyQ2MAX quadruped robot used a self-righting sequence for the recovery [1]. Castano et al. used a finite-state machine to achieve fall recovery for the wheeled quadrupedal robot CENTAURO [15]

DRL has shown new results in many fields in recent years. Model-free DRL proves to be a viable alternative for solving the problem of fall recovery. With model-free DRL, the learning agent is able to obtain the policy through massive amounts of interactions with the environment, avoiding the need to model complex interactions involving real-world dynamics explicitly. The effectiveness of DRL for robot control has been demonstrated by existing works [16], [13], [17], [18]. Fall recovery policies have been successfully trained with DRL and deployed on various different quadruped robots, such as ANYmal and Jueying [19], [13], [20]. Also, DRL has been used to learn fall recovery for humanoid character animation in physics simulation [14].

III. Methodology

A. Contact Transitional Graph

We developed a contact transitional graph for humanoid and quadruped robots to serve as a guideline for designing initialization states for learning robot fall recovery policies. The graph describes the possible transitions between different contact configurations during a fall recovery motion sequence (Fig. 2). Inspired by Borràs et al.'s approach of using simplified contact models for analyzing whole-body multi-contact motions [21], instead of specifying and numerating multiple contact points. we simplify our description of body-ground contact configurations by specifying which and how many body segments are in contact with the ground. The selection of key body postures follows three criteria: (i) It needs to represent common ground contact patterns; (ii) It needs to be stationary configurations; (iii) It has to contain a diverse set of contact configurations.

During training, initializing the robot with the key body postures provides statically balanced configurations that are closer to the desired solutions, effectively narrowing down the sample distribution and the search space. This minimizes invalid exploration of states far from the desired solution compared to the random initialization. The transition and connectivity illustrate the order of the contact configurations during a fall recovery motion. Key posture transitions are not explicitly specified during training, therefore enabling the policy to naturally explore feasible dynamic motions and learn diverse transitional motor skills between key postures.

B. Sample Distribution Augmentation

Sample distribution affects the learning result of the control policy. We augment the sample distribution using Key State Initialization (KSI) and early termination [22].

1) Initialization from Key States: Key postures are drawn randomly from the contact transitional graph (Fig. 2), which will be then used as the key state to initialize Valkyrie and Spotmicro robots during training. The same initial postures are used for Sigmaban, A1, B1, and Jueying Pro after slight adjustments to the height and joint angles.

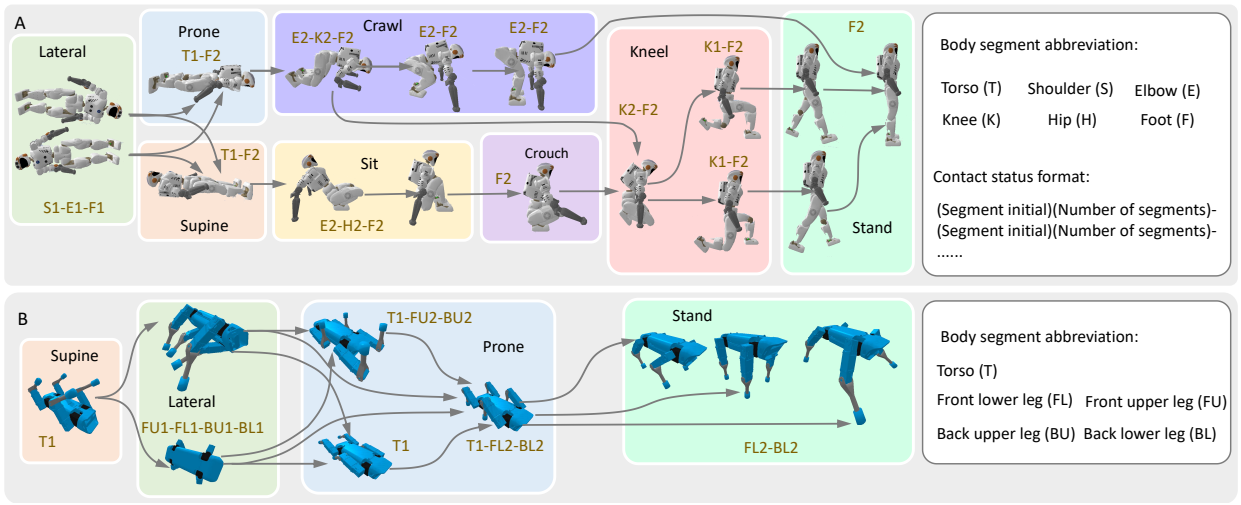

Fig. 2. Contact transitional graphs with designed key poses and sequences of standing up motions for (A) Valkyrie robot and (B) the Spotmicro robot.

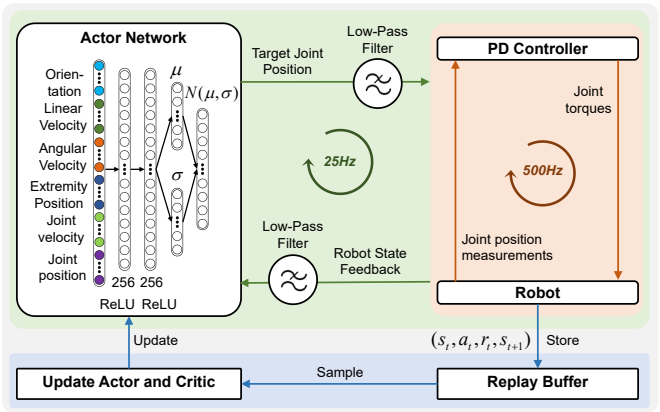

Fig. 3. Overview of the control framework .

2) Early Termination: During the early training iterations, the replay buffer will be dominated by samples in which the robot is struggling to get up due to the existence of gravity, resulting in a lack of sample diversity. Therefore, we set a time limit of 10 s to terminate the episode early and initialize the episode in different states to ensure the sample diversity.

C. Control Framework

The control framework consists of a neural network policy that generates position references for all joints at 25 Hz and PD control that interpolates the target positions and generates torque at 500 Hz. The PD controller receives target joint angles and converts them to joint torques for the motor using equation $\tau = K_p(\hat{q} - q) + K_d(0 - \dot{q})$. The PD gains K_p and K_d are manually specified for each joint.

The policy network consists of two hidden layers that uses ReLU activation function, each with 256 neurons. The size of input and output neurons depends on the DoF of the robot (Fig. 3). The policy network outputs are the mean $\mu_\theta(s_t)$ and covariance $\sigma_\theta(s_t)$ for the Gaussian distribution $u_t \sim \mathcal{N}(\mu_\theta(s_t), \sigma_\theta(s_t)^2)$, acting as a stochastic policy. A $tanh$ function is used to project the output of the stochastic policy $a_t = tanh(u_t)$ within $a_t \in (-1, 1)$, which is then re-scaled to the limits of each joint as the target joint angle in radians.

TABLE I
STATE INPUT DIMENSION.

Physical quantity	Input dimension		
	Quadruped robot	Sigmaban	Valkyrie
Gravity vector	3	3	3
Base angular velocity	3	3	3
Base linear velocity	3	3	3
Joint position	12	18	23
Joint velocity	12	18	23
Extremity position	12	12	12

1) State Representation: The state representations selected are: (i) gravity vector, (ii) base angular velocity, (ii) base linear velocity, (iv) joint position, (v) joint velocity, (vi) limb extremity position. The orientation has a vectorized representation using a gravity vector, a 3D unit vector pointing along the direction of gravity in the local frame of the robot base. The limb extremity positions are the xyz coordinates of the feet and hands in the local frame of the robot base. The state input dimension is in Table I. All feedback states are filtered by a low-pass Butterworth filter with a cut-off frequency of 10 Hz.

D. Deep Reinforcement Learning

1) Soft Actor Critic: The off-policy Soft Actor Critic (SAC) algorithm [23] is used to maximize an expected sum of rewards augmented with an additional maximum entropy objective:

$$J_{\mathrm{SAC}}(\pi) = \sum_{t=0}^{T} E_{(s_t,a_t)\sim\rho_\pi}(r(s_t,a_t)) + \alpha H(\pi(\cdot|s_t)), \quad (1)$$

where $\sum_{t=0}^{T} E_{(s_t,a_t)\sim\rho_\pi}(r(s_t,a_t))$ is the expected sum of rewards, and $H(\pi(\cdot|s_t))$ is the expected entropy of the policy π over the sample distribution ρ. The temperature term α controls the relative importance of the entropy term. Higher α results in more exploration.

E. Smoothing Output Action

Control policies learned by deep reinforcement learning in simulation may occasionally generate abrupt and jerky motions that are of high-frequencies and large amplitudes. We applied a first-order Butterworth filter with a 5 Hz cut-off frequency on the policy output to restrict undesired jerky actions [18], [20], [24], [25].

TABLE II
SAC TRAINING HYPERPARAMETERS.

Hyperparameter	Value	Hyperparameter	Value
Discount factor	0.995	Batch size	128
Target network update	0.999	Gradient update steps	4
Learning rate	3e-4	Smoothing loss	1e-3
Weight decay	1e-6	Data samples per episode	5000

TABLE III
MATHEMATICAL NOTATIONS FOR THE REWARD TERMS

Nomenclature	
φ_{base}	A unit vector in the robot base frame that points towards the direction of gravity
φ_{torso}	A unit vector in the robot torso frame that points towards the direction of gravity
h_{base}	The robot base height (z) in the world frame
h_{head}	The robot head height (z) in the world frame
v_{base}	The linear velocity of the robot base in the world frame
τ	The vector of all joint torques
q	The vector of all joint angles
$\dot{q}$	The vector of all joint velocities
$\hat{(\cdot)}$	The desired quantity of placeholder property $(\cdot)$
$p_{foot,n}$	The n-th foot horizontal placement in the world frame
p_{base}	The xy coordinates of the base in the world frame

TABLE IV
DETAILED DESCRIPTION OF TASK REWARD TERMS FOR HUMANOIDS.

Task reward terms	
Base pose	$w_1 \times K(\varphi_{base}, [0,0,-1], c_1)$
Base height	$w_2 \times K(h_{base}, \hat{h}_{base}, c_2)$
Base velocity	$w_3 \times K(v_{base}, [0,0,0], c_3)$
Joint torque regularization	$w_4 \times K(\tau, 0, c_4)$
Joint velocity regularization	$w_5 \times K(\dot{q}, 0, c_5)$
Body-ground contact	$w_6 \times \begin{cases} 0, & \text{upper body contact with ground} \\ 1, & . \end{cases}$
Upper torso pose	$w_7 \times K(\varphi_{torso}, [0,0,-1], c_7)$
Head height	$w_8 \times K(h_{head}, \hat{h}_{head}, c_8)$
Left foot placement	$w_9 \times K(p_{foot,left}, p_{base}, c_9)$
Right foot placement	$w_{10} \times K(p_{foot,right}, p_{base}, c_{10})$

Using action filtering solely is not sufficient, as it only limits the frequency but not the amplitude of the action. Hence, a loss function called *smoothing loss* is implemented to regulate the amplitude of the action [20].

$$L_{\text{smooth}}(\mu(s_t)) = \|\mu(s_t) - q\|_2^2, \tag{2}$$

where $\mu(s_t)$ is the deterministic mean outputs of the stochastic policy that are used as joint references, and q is the measured joint angles. The smoothing loss L_{smooth} minimizes the difference in joint angles between the target $\mu(s_t)$ and the current measurement q. The smoothing loss L_{smooth} is added to the SAC training loss $L_{\text{SAC}}(\pi) = -J_{\text{SAC}}(\pi)$. The final loss function to train the policy network during backpropagation is:

$$L_{\text{SAC}}(\pi) + \lambda L_{\text{smooth}}(\mu(s_t)). \tag{3}$$

The training hyperparameters are listed in Table II.

F. Reward Design

We use a Radial Basis Function (RBF) to design the bounded reward function:

$$K(x, \hat{x}, c) = e^{c(\hat{x}-x)^2}, \tag{4}$$

TABLE V
WEIGHTS w_i AND NORMALIZATION FACTOR c_i OF THE REWARD TERMS.

	Quadruped Robots					Humanoid Robots		
		Spotmicro	JueyingPro	A1	B1		Sigmaban	Valkyrie
i	w_i	c_i	c_i	c_i	c_i	w_i	c_i	c_i
1	$\frac{5}{16}$	-1.02	-1.02	-1.02	-1.02	$\frac{1}{17}$	-1.02	-1.02
2	$\frac{5}{16}$	-22.22	-8	-22.2	-8	$\frac{4}{17}$	-12.5	-2
3	$\frac{3}{16}$	-2	-2	-2	-2	$\frac{2}{17}$	-2	-2
4	$\frac{1}{16}$	-0.222	-2e-4	-2e-3	-2e-4	$\frac{1}{17}$	-0.031	-2e-5
5	$\frac{1}{16}$	-1.183	-0.012	-5e-3	-5e-3	$\frac{1}{17}$	-0.109	-0.025
6	$\frac{1}{16}$	N/A	N/A	N/A	N/A	$\frac{1}{17}$	N/A	N/A
7	0	N/A	N/A	N/A	N/A	$\frac{1}{17}$	-1.02	-1.02
8	0	N/A	N/A	N/A	N/A	$\frac{4}{17}$	-5.556	-0.692
9	0	N/A	N/A	N/A	N/A	$\frac{1}{17}$	-16.33	-2
10	0	N/A	N/A	N/A	N/A	$\frac{1}{17}$	-16.33	-2

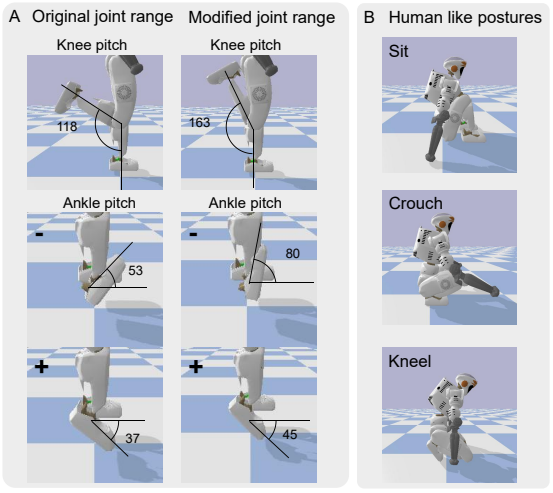

Fig. 4. Joint configurations of the Valkyrie robot. (A) Original joint range (left column) and modified joint range (middle column) of the Valkyrie robot. (B) Human-like key postures that are enabled by enlarged joint range.

where x is the physical quantity used for the evaluation, $\hat{x}$ is the desired value, and c is the parameter that controls the width of the RBF [20], [22].

The nomenclature used for the reward are described in Table III. Table IV shows the list of reward terms designed for humanoids. The base and head height terms encourage the robot to stand up and maintain the desired height. The upper torso and base poses term regulate the upper body posture of the robot. The base velocity terms penalize high velocity and hence encourage learning a smooth standing-up motion. The joint torque and velocity regularization terms penalize high torques and velocities of the joints respectively. The body ground contact term rewards the agent when upper body parts are not in contact with the ground. The foot placement term guides the left and right feet to be close to the projected pelvis position on the ground. The weights and normalization factors of the reward terms are shown in Table. V.

IV. RESULTS

A. Simulation Setup

To validate how generic our proposed learning method is for fall recovery, we used six different robot models of different sizes: (A) Spotmicro, (B) Jueying Pro, (C) Sigmaban, (D) A1, (E) B1, and (F) Valkyrie (Fig. 1). The physics simulation

uses PyBullet [26] in which the policy is trained and tests are conducted.

The Spotmicro is an open-source project to replicate the Spot quadruped from Boston Dynamics with a smaller form-factor. The A1, B1, and Jueying Pro robot are high-performance commercial quadrupeds with a weight of $12\,\mathrm{kg}$, $50\,\mathrm{kg}$, and $70\,\mathrm{kg}$, respectively. All quadrupeds have 3 Degrees of Freedom (DoF) per leg.

The Sigmaban robot is a humanoid robot developed by the Rhoban football team for the humanoid kid-size league of the Robocup soccer tournament [27]. It has a weight of approximately $6\,\mathrm{kg}$ and a height of approximately $0.6\,\mathrm{m}$. Valkyrie is a humanoid robot designed by NASA for extra-terrestrial space missions, and stands $1.87\,\mathrm{m}$ tall and weighs approximately $130\,\mathrm{kg}$. Valkyrie's original design of joint ranges are overly restrictive, which prevents human-like standing-up behaviours. We modified the joint limits and collision mesh to increase the range of motions to resemble that of humans. The augmented Valkyrie robot is able to perform human-like squatting, crouching, kneeling, and sitting motions (Fig. 4). The rest of the robots use the original models with real hardware restrictions for the joint range, joint velocity limit, and joint torque limit.

B. Comparison of State Initialization

We compare our KSI to Random State Initialization (RSI). Under RSI training configuration, the base orientation of the quadruped robots is randomly initialized with $\theta_{roll} \sim U(-\pi, \pi)$, $\theta_{pitch} \sim U(-\frac{\pi}{2}, \frac{\pi}{2})$, $\theta_{yaw} = 0$, while the base orientation of humanoid robots is initialized with $\theta_{roll} \sim U(-\frac{\pi}{2}, \frac{\pi}{2})$, $\theta_{pitch} \sim U(-\frac{\pi}{2}, \frac{\pi}{2})$, $\theta_{yaw} = 0$. The joint angles are uniformly sampled within the joint range specified by the robot model. Joint angle configurations that cause self-collision are abandoned. The robots are then dropped above the ground at the start of each training iteration.

All four quadrupeds are able to learn successful fall recovery policies with both KSI and RSI training configurations. For Jueying Pro, A1 and B1, KSI converges to a successful fall recovery policy faster than RSI. For Spotmicro, KSI offers no significant advantage over RSI. However, humanoid robots struggle to learn successful fall recovery policies with RSI, as indicated by the lower sum of rewards (Fig. 5).

We conducted an ablation study to investigate how the selection of key states affects the learning of humanoid fall recovery policies. Training configuration KSI-A uses only standing and kneeling key postures for initialization, whereas KSI-B removed the standing and kneeling key postures. When the standing and kneeling postures are removed, the humanoid robots fail to learn a successful policy (Fig. 5E & F). Results show that stable upright standing postures are crucial for the successful learning of humanoid fall recovery policies. With RSI, the randomly generated initial configurations are mostly far away from stable standing postures. Standing upright is a process of going through configurations that can counterbalance gravity, thus not random. This is the underlying reason that can explain why RSI is less effective.

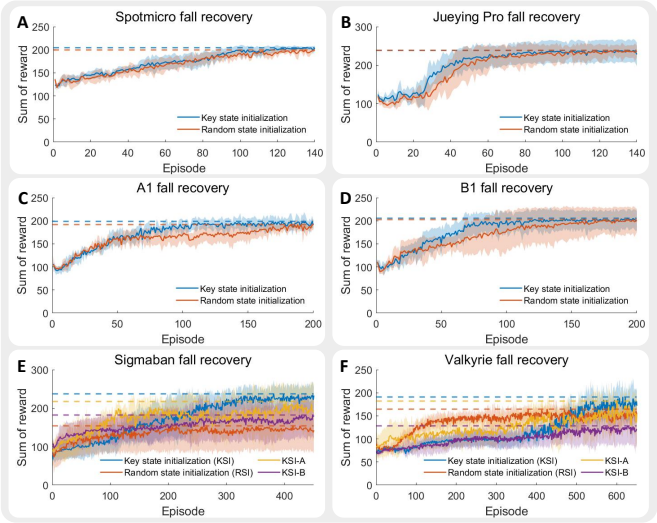

Fig. 5. Learning curve for the fall recovery policies of (A) Spotmicro, (B) Jueying Pro, (C) A1, (D) B1, (E) Sigmaban, (F) Valkyrie. The results are averaged over 6 trials, each with a different random seed. A separate testing rollout is conducted at the end of each episode to evaluate the performance. The deterministic mean of the learned Gaussian policy is executed during the testing rollout. All policies are evaluated under the exact same environmental condition and initial state configuration during the testing rollout, regardless of initial state configuration during the training rollouts.

C. Fall Recovery of Quadrupeds

The fall recovery policies for Spotmicro and Jueying Pro are able to recover from both supine and lateral postures (Fig. 6A1 & B1) (See accompanying video for quadruped fall recovery maneuvers from lateral postures.).

From Fig. 6A3 and B3 we can observe that fall recovery can be classified by three phases: (i) Self righting, (ii) standing up, and (iii) stabilization. In the first phase, the robot reorients itself to minimize the postural error compared to the nominal standing posture. In the second phase, the robot starts to support its weight and lift up its body. In the final phase, the robot adjusts the body posture and stabilizes itself.

D. Fall Recovery of Humanoids

Compared to quadrupeds, humanoids have higher center of mass and smaller support polygon, which makes them prone to falling and thus learning effective fall recovery becomes much more challenging. This is reflected in the learning curve, as humanoid policies require more episodes to converge (Fig. 5).

The fall recovery policies for both Sigmaban and Valkyrie are able to recover from supine, prone, and lateral postures (Fig. 7A1 & B1) (See accompanying video for humanoid fall recovery maneuvers from lateral and prone postures.). From Fig. 7A3 and B3, we can observe that humanoid fall recovery also has three phases similar to that of quadrupeds: (i) Self righting, (ii) standing up, and (iii) stabilization. Despite the differences in the size and shape between Sigmaban and Valkyrie, when recovering from a supine posture, both robots learn to roll and adjust into prone posture first which provides the robots with enough clearance to utilize their arms. We can see that the arm movements that support the upper body while standing up are fairly human-like and natural.

In contrast to regular locomotion where only feet are in contact, a successful fall recovery requires the quadruped and

Fig. 6. Snapshots of Spotmicro (A1) and Jueying Pro (B1) performing fall recovery maneuvers in simulation. (A2 & B2) Contact status of body segments over time corresponding to (A1) and (B1). (A3 & B3) Orientation error and Height of robot base corresponding to (A1) and (B1). See accompanying video for fall recovery maneuvers of quadruped robot A1 and B1.

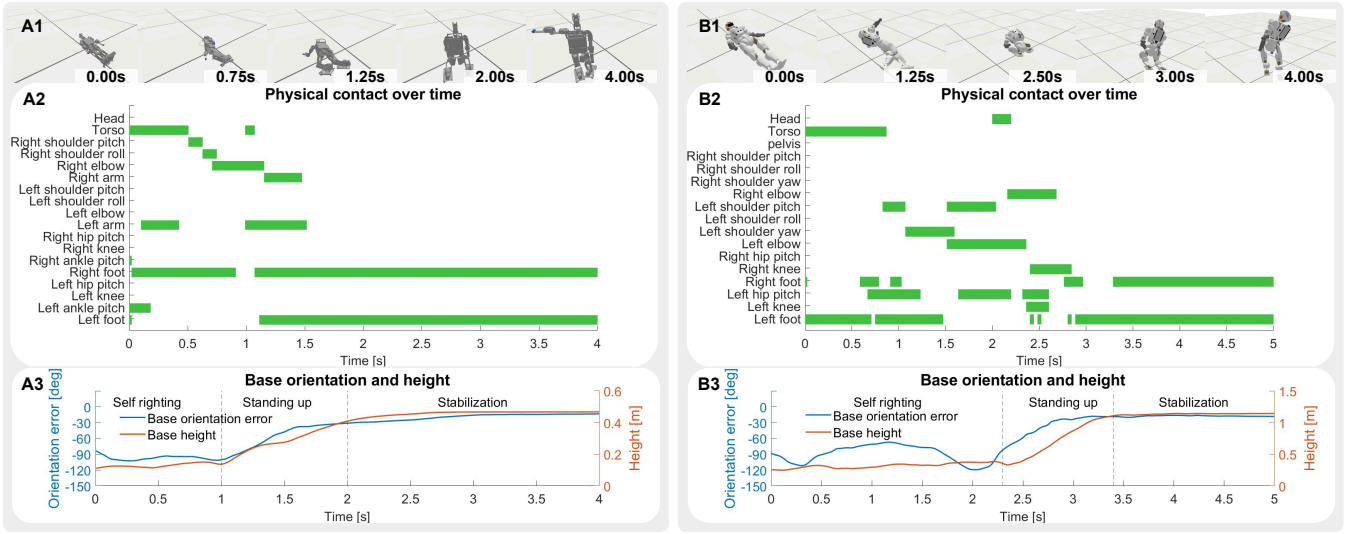

Fig. 7. Snapshots of Sigmaban (A1) and Valkyrie (B1) performing fall recovery maneuvers in simulation.(A2 & B2) Contact status of body segments over time corresponding to (A1) and (B1). (A3 & B3) Orientation error and Height of robot base corresponding to (A1) and (B1).

humanoid to undergo a complex sequence of ground contacts, as shown by the data analysis of physical contact of all body segments over time (Fig. 6A2, 6B2, 7A2, and 7B2). Such contact sequences are difficult to handcraft and never exactly the same twice, which show that fall recovery involves whole-body, any-point contacts with the ground, and hence is indeed a complex and interactive behavior.

E. Robustness against Uncertainties

To demonstrate the advantage of the learning-based feedback policy, we designed three extreme, unseen scenarios to evaluate the robustness: 1. Uneven terrain. 2. Low friction. 3. Push disturbance. We show the capabilities of reactive adaptation to external perturbations robustly with consistent performance across different robots, which clearly a predefined or replanned motion sequence cannot do. Due to concern of hardware damage, the three extreme robustness test scenarios are conducted in simulation (See accompanying video for robot fall recovery maneuvers in robustness test scenarios.).

The performance metric is the success rates $sr = n_s/n_t$, where n_s is the number of successful runs and n_t is the total

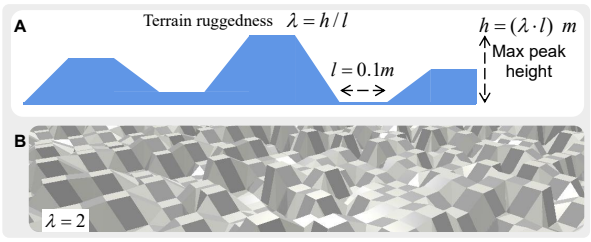

Fig. 8. Fall recovery on unseen rugged terrains: (A) Cross section view of uneven terrains; (B) Overall view of uneven terrain.

number of runs. We trained 6 policies for each robot, each has 10 runs, resulting in $n_t = 60$. Fall recovery is considered successful when the robot is able to recover to an upright standing posture, defined as: (i) Feet have to be the only body part in contact with the ground; (ii) Base orientation has to satisfy $|\theta_{pitch}| < \frac{\pi}{4}$ and $|\theta_{roll}| < \frac{\pi}{4}$; (iii) Base height has to satisfy $h_{base} > h^{success}$. The height criteria $h^{success}$ is set to $0.13m$, $0.4m$, $0.25m$, $0.4m$, $0.38m$, $0.8m$ for Spotmicro, Jueying Pro, A1, B1, Sigmaban, and Valkyrie respectively.

1) Uneven terrain: The uneven terrain is automatically generated using the following configurations. The terrain consists

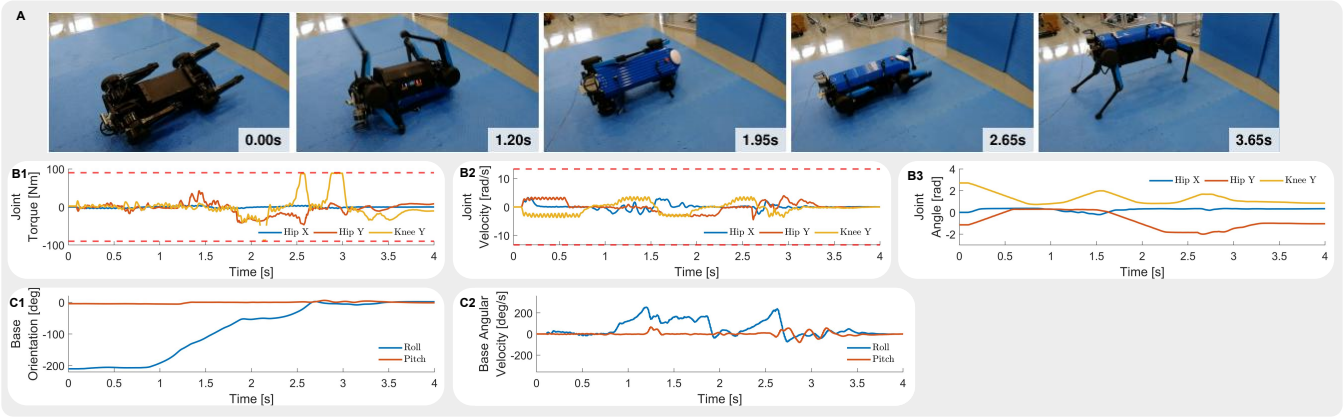

Fig. 9. Fall recovery experiment on real Jueying Pro robot: (A) Time-elapsed snapshots; (B1-B3) The joint torque, velocity, and angle of a single leg; and (C1-C2) Robot base orientation and angular velocity.

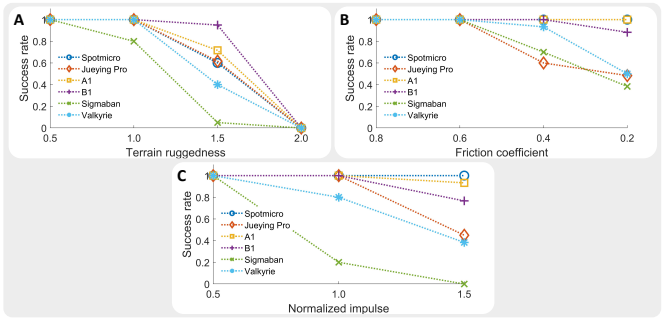

Fig. 10. Success rate calculated over 60 runs. We train 6 policies for each robot and run the policies 10 times with different initial configurations. (A) Terrain ruggedness. (B) Friction Coefficient. (C) Normalized Impulse.

of interconnected slopes each covering an area with width of $l = 0.1\,\mathrm{m}$. Adjacent slopes connect and form "hills" and "valleys". We measure the ruggedness of the terrain as the ratio $\lambda = h/l$, where h is the maximum height of the hill peak formed by the slopes. The height of each slope are randomly sampled uniformly between $(0, \lambda \cdot 0.1\,\mathrm{m})$ (Fig. 8). All policies are able to successfully perform fall recovery on uneven terrain to a certain extent (Fig. 10A).

2) Low friction: Policies are tested under different slippery grounds, with a range of friction coefficient μ of 0.2, 0.4, 0.6, 0.8. In general, quadruped robots are more robust to ground friction changes than humanoids (see Fig. 10B).

3) Extremely large force disturbance: We applied force disturbances to the robot sideways for a period of $0.2\,\mathrm{s}$. The mass and inertia of the robot affect how the robot responds to disturbances. Therefore, we normalize the impulse by the mass of the robot and use it for comparison across different robots. The unit of the normalized impulse is $\mathrm{N\,s\,kg^{-1}}$. It can be seen that humanoid robots are less resilient to push (Fig. 10C). Even if the robots fail to resist push and fall over, they are still able to recover and stand up again (see accompanying video).

It shall be noted that for the generality of the method and straightforwardness of replication, we kept a small number of parameters to set up and minimal complexity for training. For example, all policies were trained on a flat ground only with a friction coefficient of $\mu = 1.0$, and has no knowledge of the terrain unevenness, actual friction coefficient, or push disturbance. The fact, that the policies are capable of generalizing to situations outside of the training dataset, demonstrated the robustness of the learned policy and the effectiveness of the DRL framework.

TABLE VI
MAXIMUM JOINT TORQUE, VELOCITY, AND POWER OF SPOTMICRO, A1, AND B1 AVERAGED OVER MULTIPLE TRIALS.

Joint	Torque [N m]		Velocity [rad s^{-1}]		Power [W]	
Spotmicro	mean±std	limit	mean±std	limit	mean±std	limit
HipX	1.91±0.59	3	4.67±0.83	8	5.19±2.07	32
HipY	2.69±0.32	3	6.31±1.47	8	10.87±5.97	32
kneeY	1.48±0.41	3	6.79±1.45	8	5.28±2.70	32
A1	mean±std	limit	mean±std	limit	mean±std	limit
HipX	12.16±4.96	33.5	4.63±1.02	21	40.1±28.7	704
HipY	28.01±4.45	33.5	7.49±0.78	21	137.5±35.2	704
kneeY	14.97±1.99	33.5	3.50±1.22	21	21.5±4.4	704
B1	mean±std	limit	mean±std	limit	mean±std	limit
HipX	76.5±14.5	91.0	6.62±1.80	19.7	300±145	1792
HipY	88.4±13.0	93.3	8.05±1.71	23.3	526±192	2174
kneeY	40.1±25.0	140	4.78±1.78	15.6	215±150	2184

TABLE VII
MAXIMUM JOINT TORQUE, VELOCITY, AND POWER OF SIGMABAN AVERAGED OVER MULTIPLE TRIALS.

Joint	Torque [N m]		Velocity [rad s^{-1}]		Power [W]	
	mean±std	limit	mean±std	limit	mean±std	limit
HipZ	3.87±1.92	8.4	1.56±0.51	4.7	4.38±2.60	39.5
HipX	5.43±1.77	8.4	1.35±0.34	4.7	4.40±2.01	39.5
HipY	6.60±0.95	8.4	2.42±0.23	4.7	9.44±2.63	39.5
KneeY	3.52±1.02	8.4	2.36±0.41	4.7	5.18±1.00	39.5
AnkleY	6.50±1.76	8.4	2.17±0.37	4.7	6.73±3.40	39.5
AnkleX	6.87±0.98	8.4	1.77±0.25	4.7	2.59±2.32	39.5
ShoulderY	7.11±0.63	8.4	3.57±0.86	4.7	15.81±4.42	39.5
ShoulderX	4.67±1.12	8.4	1.41±0.49	4.7	5.11±2.74	39.5
ElbowY	4.04±1.20	8.4	3.55±0.38	4.7	9.80±7.46	39.5

F. Hardware Experiments

We validated our approach on a real Jueying Pro quadruped robot. The real Jueying Pro robot is able to successfully perform fall recovery maneuver using the trained policy (Fig. 9). The joint torques and velocities stay within the motor limits (Fig. 9B1 & B2). The overall fall recovery motion is smooth and steady as can be seen from the smooth movement of joints (Fig. 9B3), and gradual change in base orientation and base angular velocity (Fig. 9C1 & C2).

Due to the lack of hardware accessibility, we analyzed the torque, velocity, and power of critical joints in simulation

to determine the feasibility of hardware implementation of Spotmicro, A1, B1, and Sigmaban. The torque constraint is handled by clipping the commanded torque for the joint motor. The mean of the maximum joint torque and joint velocity respect the constraints of the joint motors. The mean of the maximum joint power is within the power limit, where max joint torque and joint velocity do not occur simultaneously (Table. VI and Table. VII). Note that the power limit is calculated by multiplying the torque limit and velocity limit. The calculated power limit serves as a reference and is not the actual power limit.

V. Conclusion

This work proposed a DRL framework that can learn versatile fall recovery policies for different humanoid and quadruped robots. We also proposed a design guideline for the contact transition graph which is used for the selection of key robot states for initialization. Compared to random initializations, our approach speeds up the learning of quadruped fall recovery policies and improves the performance of humanoid fall recovery policies.

The fall recovery policies generated natural and animal-like behaviors, demonstrating the feasibility of using DRL to automatically produce standing-up behaviours for legged robots. The proposed learning framework is agnostic to robots with very different morphologies, shapes, and sizes. The learned policies are robust towards the environmental uncertainties, as shown by the successful fall recovery in the unseen new cases of rough terrains, low ground friction, and large push disturbances. Moreover, the effectiveness and feasibility of the learning framework were validated on the real Jueying Pro quadruped robot.

In future work, our framework can be extended to learn fall recovery motions for other robot types, including hexapod robots and wheel-leg robots. Future work can further enhance the performance of our framework by implementing dynamic randomization for sim2real transfer and tailoring the training curriculum for specific robots and applications. This can lead to improved performance on real robotic systems in a wider range of scenarios.

References

[1] C. Semini, J. Goldsmith, B. U. Rehman, M. Frigerio, V. Barasuol, M. Focchi, and D. G. Caldwell, "Design overview of the hydraulic quadruped robots," in The Fourteenth Scandinavian International Conference on Fluid Power, 2015, pp. 20–22.

[2] F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, S. Kajita, K. Yokoi, H. Hirukawa, K. Akachi, and T. Isozumi, "The first humanoid robot that has the same size as a human and that can lie down and get up," in 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), vol. 2. IEEE, 2003, pp. 1633–1639.

[3] J. Stückler, J. Schwenk, and S. Behnke, "Getting back on two feet: Reliable standing-up routines for a humanoid robot." in IAS, 2006, pp. 676–685.

[4] Z. Li, C. Zhou, J. Castano, X. Wang, F. Negrello, N. G. Tsagarakis, and D. G. Caldwell, "Fall prediction of legged robots based on energy state and its implication of balance augmentation: A study on the humanoid," in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 5094–5100.

[5] K. Araki, T. Miwa, H. Shigemune, S. Hashimoto, and H. Sawada, "Standing-up control of a fallen humanoid robot based on the ground-contacting state of the body," in IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2018, pp. 3292–3297.

[6] A. Radulescu, I. Havoutis, D. G. Caldwell, and C. Semini, "Whole-body trajectory optimization for non-periodic dynamic motions on quadrupedal systems," in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 5302–5307.

[7] M. Al Borno, M. De Lasa, and A. Hertzmann, "Trajectory optimization for full-body movements with complex contacts," IEEE transactions on visualization and computer graphics, vol. 19, no. 8, pp. 1405–1414, 2012.

[8] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," ACM Transactions on Graphics (TOG), vol. 31, no. 4, pp. 1–8, 2012.

[9] Z. Manchester and S. Kuindersma, "Variational contact-implicit trajectory optimization," in Robotics Research. Springer, 2020, pp. 985–1000.

[10] A. Patel, S. L. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler, "Contact-implicit trajectory optimization using orthogonal collocation," IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 2242–2249, 2019.

[11] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," The International Journal of Robotics Research, vol. 33, no. 1, pp. 69–81, 2014.

[12] T. A. Howell, S. Le Cleac'h, S. Singh, P. Florence, Z. Manchester, and V. Sindhwani, "Trajectory optimization with optimization-based dynamics," IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 6750–6757, 2022.

[13] 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.

[14] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters," ACM Transactions On Graphics (TOG), vol. 41, no. 4, pp. 1–17, 2022.

[15] J. A. Castano, C. Zhou, and N. Tsagarakis, "Design a fall recovery strategy for a wheel-legged quadruped robot using stability feature space," in 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2019.

[16] J. Siekmann, Y. Godse, A. Fern, and J. Hurst, "Sim-to-real learning of all common bipedal gaits via periodic reward composition," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 7309–7315.

[17] C. Zhang, W. Yu, and Z. Li, "Accessibility-based clustering for efficient learning of locomotion skills," International Conference on Robotics and Automation (ICRA), 2022.

[18] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in Robotics: Science and Systems, 07 2020.

[19] J. Lee, J. Hwangbo, and M. Hutter, "Robust recovery controller for a quadrupedal robot using deep reinforcement learning," arXiv preprint arXiv:1901.07517, 2019.

[20] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li, "Multi-expert learning of adaptive legged locomotion," Science Robotics, vol. 5, no. 49, p. eabb2174, 2020.

[21] J. Borràs, C. Mandery, and T. Asfour, "A whole-body support pose taxonomy for multi-contact humanoid robot motions," Science Robotics, vol. 2, no. 13, 2017.

[22] C. Yang, K. Yuan, S. Heng, T. Komura, and Z. Li, "Learning natural locomotion behaviors for humanoid robots using human bias," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2610–2617, 2020.

[23] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in International conference on machine learning. PMLR, 2018, pp. 1861–1870.

[24] K. Bergamin, S. Clavet, D. Holden, and J. R. Forbes, "Drecon: data-driven responsive control of physics-based characters," ACM Transactions On Graphics (TOG), vol. 38, no. 6, pp. 1–11, 2019.

[25] D. Rodriguez and S. Behnke, "Deepwalk: Omnidirectional bipedal gait by deep reinforcement learning," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 3033–3039.

[26] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[27] Q. Rouxel, G. Passault, L. Hofer, S. N'Guyen, and O. Ly, "Rhoban hardware and software open source contributions for robocup humanoids," in 10th Workshop on Humanoid Soccer Robots, IEEE-RAS Int. Conference on Humanoid Robots, Seoul, Korea, 2015.