Design and Learning-based Control of Agile-PF: an Agile and Easy-to-Reproduce Point-Foot Biped Robot

Yasen Jia[a], Chenxiang Zhang[a] and Yan Huang[a,b,*]

[a]School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, 100081, China
[b]Key Laboratory of Biomimetic Robots and Systems, Ministry of Education, Beijing, 100081, China

ARTICLE INFO

Keywords:
legged robot
bipedal locomotion
reinforcement learning

ABSTRACT

Latest progress in Reinforcement Learning (RL) has spurred growing interest in the development of biped robots. Multiple human-size and small-scale biped robots have demonstrated outstanding locomotion performance. However, they mostly employ custom control boards and actuators or possess complicated architectures, raising the threshold of reproduction for beginner-level enthusiasts in the field of robotics. Biped robots that are easy to reproduce can help accelerate related research and provide accessible platforms for education and benchmarking. In this paper, we develop a small-scale Agile Point-Foot biped robot (Agile-PF) with strong reproducibility. The design emphasizes low-cost and commercially available components and thus reduces the barrier to robot system development. Additionally, we propose an RL-based training framework that incorporates hybrid dynamics representation and sim-to-real alignment techniques. With policies trained via this framework, the robot demonstrates remarkable capabilities, including high-speed running, traversing diverse terrains, pushing recovery, and load-bearing performance.

1. Introduction

Biped robots have long been a focus of research due to their biomimetic morphology and locomotion capabilities. Recently, robots of this kind has gained extensive attention due to the advance of artificial intelligence and robotics[1–6].

Human-size biped robots has shown outstanding performance in terms of human-like motion and robust locomotion over complex terrains[1–3]. Liao *et al.*[1] proposed a control method BeyondMimic for humanoid robots to learn versatile and naturalistic movements from human motions via guided diffusion. Policies trained with BeyondMimic can accomplish challenging skills like jumping spins, sprinting and cartwheel, demonstrating extraordinary dynamic performance. Zhuang *et al.*[2] presented a method to make the robot learn to conduct agile parkour motions with teacher-student distillation and depth images. Though these latest results have been shown on human-size robots like Unitree G1 and H1, these commercial robot platforms are complicated and hard to reproduce. On the other side, small-scale biped robots possess simpler architectures but usually consist of custom parts, adding the difficulty to reproduce[4–6]. Zhu *et al.*[4] developed BRAVER, a 0.51m tall point-foot biped robot actuated via proprioceptive electric motors. It utilized custom actuators and control boards to build the robot system. Using RL-based controllers, BRAVER can run up to 2m/s, demonstrating remarkable motion performance. Bolt[5] was a 0.45m tall open-source biped robot whose configuration was similar to BRAVER. It also utilized custom actuators and control boards. Bolt featured light weight but fell short of running and load capacity. Chi *et al.*[6] proposed Berkeley Humanoid Lite, an open-source and customizable humanoid robot with a height of 0.55m. Employing customized actuators and 3D-printed mechanical structures, the Berkeley Humanoid Lite robot only cost $5000 to build from scratch. It can perform bipedal walking, teleoperated manipulation and can also be used in education and animatronics. However, the athletic performance it presented was limited to low-speed walking. These small-scale biped robots often exploit custom control boards and actuators, raising the threshold of developing a small-scale biped robot for beginner-level enthusiasts in the field of robotics. Currently, there is a lack of an easily replicable biped robot with concise architecture and high performance. With better replicability, this robot platform can help robotics enthusiasts understand essential knowledge of a robot system swiftly. Moreover, this easy-to-reproduce robot can be used for secondary development, robotics research and technology education.

In terms of leg design, biped robots can be broadly classified into two types. The first type directly couples actuators to joints, as seen in robots such as Lola [7] and COMAN [8]. This arrangement eliminates the need for complex transmission systems and provides a large workspace, but often results in higher leg mass and restricted agility. In contrast, the second category employs closed-chain linkages to substantially reduce leg inertia, exemplified by point-foot robots like BRAVER[4] and Bolt[5]. Such mechanisms facilitate highly dynamic maneuvers, including running, jumping, and back-flipping. Furthermore, the point-foot design can get rid of the inertia of ankle actuators, fostering higher dynamic performance. To this end, to make an easy-to-reproduce biped robot with simplified architecture and high performance, point-foot biped robots with closed-chain linkages are ideal.

This paper develops Agile-PF (Agile Point-Foot biped robot), a high-performance bipedal platform characterized

*Corresponding author
jason_1120202397@163.com (Y. Jia); yanhuang@bit.edu.cn (Y. Huang)
ORCID(s): 0000-0001-6235-6780 (Y. Jia)

by a simplified architecture and strong reproducibility. The robot is 0.53m tall and weighs 13kg. Each leg of it integrates three actuators mounted at the hip, with knee motion realized through a closed-chain linkage. The objective of this work is to establish an highly reproducible bipedal robot platform with high performance. In order to achieve this goal, we utilize low-cost, commercially available components to construct the robot. All electronic parts are standard commercial products. Mechanical parts can be easily accessed through 3D print or Computer Numerical Control (CNC) manufacturing. Compared with BRAVER, Berkeley Humanoid Lite and Bolt which employ custom actuators and control boards, our robot offers enhanced replicability. In addition, the total cost of Agile-PF is about \$6400/\$5000 (using aluminium alloy/3D-printed mechanical parts), as shown in Table 1. Furthermore, the performance of Agile-PF is not compromised with better replicability and low cost. Based on reinforcement learning, we successfully train a unified locomotion policy that enables agile high-speed running (up to 3m/s), stable traversal across varied terrains (including 30° slopes and 15cm stairs), significant load-bearing capacity (balancing in place with 10kg load), and robust push recovery. With improved athletic capability, this platform is well-suited for research, education, entertainment and many other fields. The primary contributions of this work are summarized as follows:

1. An easy-to-reproduce point-foot biped robot, Agile-PF, designed with low leg inertia via proximal actuator placement at the hip. The exclusive use of commercially available electronic components and simplified architecture ensures straightforward replicability and reduced development overhead.
2. A reinforcement learning framework that combines hybrid dynamic representations with sim-to-real alignment techniques, effectively training control policies capable of high-speed locomotion and robust terrain adaptation.
3. Comprehensive real-world experiments conducted in both indoor and outdoor environments, validating the robot's performance in high-speed movement, multi-terrain walking, recovery from external perturbations, and loaded operation.

2. Robot Design

In this section, we provide a detailed description of the robot design, including its leg and foot design, control and sensor system. The total cost of the system is approximately \$6400, with a detailed cost breakdown provided in Table 1. All mechanical parts are made of aluminium alloy to ensure reliability. By replacing aluminum alloy parts with 3D printed ones, costs can be further reduced to about \$5000. Files about mechanical structure, program codes can be seen in supplemented materials. These materials will be open source afterwards.

Table 1
Costs of Components of Agile-PF

Name	Cost (\$)
EC-A8112-P1-18	3257.1 (6 actuators)
Upper Board RDKX5	71.4
Lower Board DM-MC02	28.5
Joystick	2.8
IMU HI14M0	92.8
DC/DC Converter	5.7
Battery (48V, 10AH)	42.8
Power Distribution Board	19.2
Connectors	10.7
Cables	7.0
Mechanical Part Manufacturing	2857.1
Total Cost	6395.1

Table 2
Agile-PF Specifications

Robot Specifications	Values
Dimension (L × W × H (mm))	400 × 230 × 530
Upper leg length (mm)	200
Lower leg length (mm)	200
Total weight (kg)	13
Actuators Specifications	**Values**
Nominal Torque (Nm)	30
Maximum Torque (Nm)	90
Maximum Speed (rad/s)	15
Mass (kg)	0.84

2.1. Mechanical Design of Foot and Leg

Fig. 1 (a)-(d) illustrates the CAD model of Agile-PF with its key dimensions, and Fig. 1 (e) presents a snapshot of the real robot. Table 2 lists the main specifications of the robot and its actuators. The robot features a modular design, consisting of three modules: a battery compartment, an electronics enclosure, and a lower limb assembly. The battery compartment contains a 48 V, 10 Ah battery to supply power to the robot. The electronics enclosure houses and protects the robot's electronic components. The lower limb possesses six degrees of freedom (DoF), distributed equally between both legs. The modular design simplifies the architecture and facilitates both reconfiguration and repair of the robot. The robot has a height of 530 mm in its normal standing configuration, extending to 650 mm with fully extended legs.

Fig. 2 illustrates the detailed design of the robot's lower limb assembly. Each leg of the robot has three DoFs: hip abduction-adduction with a range of ±30°, hip flexion-extension with a range of ±122°, and knee flexion-extension with a range of 0° to 146°. Every DoF is independently actuated by an integrated joint module (EC-A8112-P1-18, ENCOS). The joint module provides good output performance, enabling the robot to execute highly dynamic movements. By mounting all three joint modules at the hip, the inertia of the leg linkages is significantly reduced. As a result, the combined mass of the linkages is approximately 1 kg (7.7% of the 13 kg total mass), which allows the robot to be effectively modeled as an inverted pendulum system [9]. To enhance stability in the nominal standing posture, the leg linkage is designed to vertically align the Center of Mass (CoM) over the foot, thereby achieving static equilibrium. Furthermore, the knee joint module is positioned at the midpoint of its operational range to balance the range of motion for the leg; additionally, the knee's four-bar linkage is arranged in a rectangular configuration, thereby maximizing torque transmission efficiency. To reduce the impact upon ground contact during each foot strike, we designed a curved foot made of rubber. The grid-patterned texture of the sole ensures effective friction between the sole and the ground in all directions.

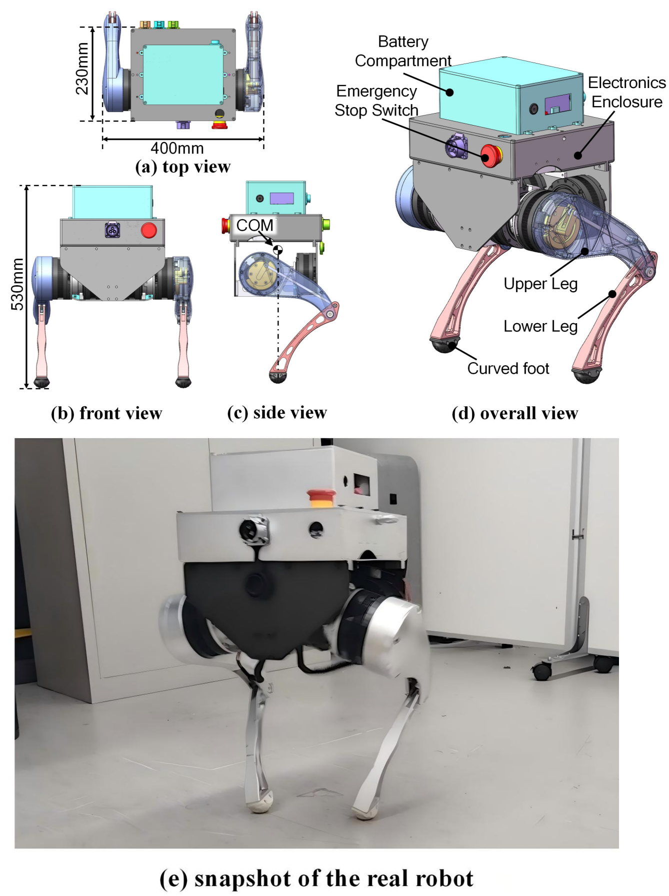


Fig. 1: The CAD drawings with key dimensions of Agile-PF (a-d) and a snapshot of the real robot (e).

2.2. Control and Sensor System

Fig. 3 illustrates the electrical system architecture of Agile-PF. The whole electrical system can be separated into four parts: power supply, control, actuation and sensing, with components of each part marked as green, blue, red and yellow in Fig. 3. The primary computing unit RDKX5, manages sensor data acquisition, executes neural network inference, and sends control commands to the subordinate board via a Universal Serial Bus (USB). The secondary controller DM-MC02, receives these directives and communicates with actuators using a Controller Area Network (CAN) bus. Base

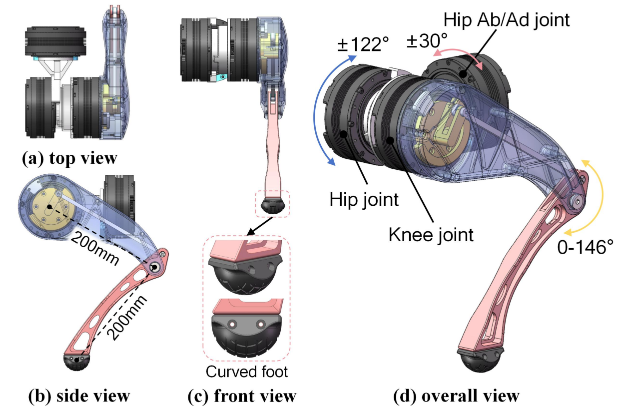


Fig. 2: Actuators and joint configurations of one leg.

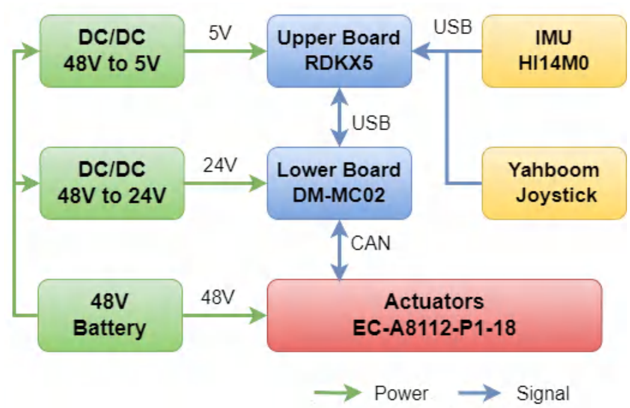


Fig. 3: Overview of the electrical system of Agile-PF

orientation measurements are collected from a HI14M0 Inertial Measurement Unit (IMU), and user inputs are obtained from a Yahboom wireless joystick; both devices transmit data over the USB interface, which operates at 500 Hz. To meet diverse voltage demands across components, two DC/DC converters allocate power from the battery. Finally, actuators (EC-A8112-P1-18, ENCOS) receive commands to perform rotary motion and feeds joint angles and angular velocities back to DM-MC02. All of these electrical components are accessible commercial products. The physical arrangement of the electrical components of the platform is shown in Fig. 4. Essentially, off-the-shelf commercial products have the same interfaces and functions as those custom control boards and actuators. Meanwhile, numerous validated instructions and open-source examples associated with these commercial products are available online, making it easier to reproduce our robot.

The program in RDKX5 is responsible for both real-time communication and neural network inference. We implemented a framework based on multi-thread programming of C++ to ensure the real-time capability of communication tasks. For DM-MC02, we utilized interrupt callbacks to manage different tasks, realizing high-frequency communication with both RDKX5 and actuators.

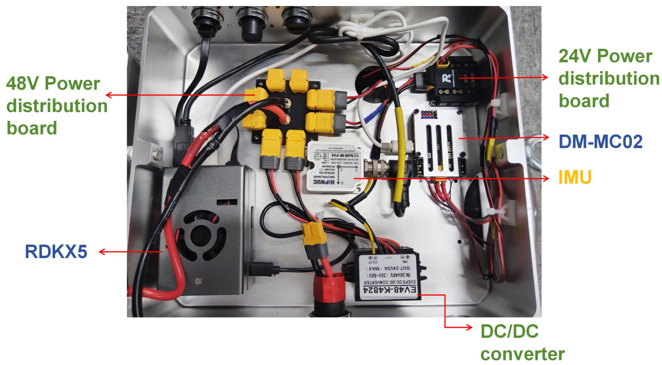


Fig. 4: Layout of the electrical system of Agile-PF

3. Robot Control

The diagram of our training framework is shown in Fig. 5, where two main components are: dynamics representation based on hybrid information and real-to-sim alignment. These two components are explained in 3.2 and 3.3 respectively.

3.1. Problem Formulation

We formulate the locomotion control problem as an infinite-horizon Partially Observable Markov Decision Process (POMDP) defined by $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{O}$ is the observation space, $\mathcal{A}$ is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is the transition function, $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function and γ is the discount factor. The full state, partial observation and action are continuous and defined by $\boldsymbol{s} \in \mathcal{S}$, $\boldsymbol{o} \in \mathcal{O}$, and $\boldsymbol{a} \in \mathcal{A}$ respectively. We define a temporal observation at time t over the past H measurements as $\boldsymbol{o}_t^H = [\boldsymbol{o}_t\ \boldsymbol{o}_{t-1}\ ...\ \boldsymbol{o}_{t-H}]^T$. The objective of RL is to train a policy π^* that maximizes the discounted accumulative reward $\pi^* = \arg\max_\pi \mathbb{E}_{a_t \sim \pi(\cdot|s_t)}[\sum_{t \geq 0} \gamma^t r(s_t, a_t)]$. We adopt Proximal Policy Optimization (PPO) [10], a state-of-the-art RL algorithm to solve the POMDP. PPO jointly trains a policy network $\pi_\theta(a|s)$ and a critic function V_ϕ.

3.2. Robust Locomotion based on Hybrid Information

In Teacher-Student[11] architecture, the proposal of latent vector as a representation of inherent dynamics strongly improves the robustness of the policy. Besides, the results of EstimatorNet[12] highlights the importance of explicit physical information. Inspired by DreamWaQ[13], we propose a training framework utilizing both explicit and implicit information. We utilize an autoencoder [14] architecture to extract essential features from proprioceptive observation.

The hybrid encoder encodes history observation $\boldsymbol{o}_t^H$ into explicit information $\hat{\boldsymbol{e}}_t$ and implicit information $\boldsymbol{i}_t$. The estimated explicit information is guided to approximate ground-truth physical values $\boldsymbol{e}_t$ in simulation through supervised learning. The mean-squared-error (MSE) loss of explicit estimation is as follows:

$$\mathcal{L}_{est} = MSE(\hat{\boldsymbol{e}}_t, \boldsymbol{e}_t) \tag{1}$$

Besides, to achieve one-stage learning and make hybrid information represent the underlying dynamics, the output of the decoder $\hat{\boldsymbol{o}}_{t+1}$ is guided to approximate true value of next observation $\boldsymbol{o}_{t+1}$. The reconstruction loss is defined as follows:

$$\mathcal{L}_{recon} = MSE(\hat{\boldsymbol{o}}_{t+1}, \boldsymbol{o}_{t+1}) \tag{2}$$

Therefore, the total loss for the hybrid encoder is defined as follows:

$$\mathcal{L}_{hybrid} = \mathcal{L}_{est} + \mathcal{L}_{recon} \tag{3}$$

The hybrid encoder essentially compresses history observation into low-dimensional explicit and implicit information. Then the explicit and implicit information, as the representation of underlying dynamics, are decoded to reconstruct next observation.

3.3. Zero-shot Sim-to-Real Transfer through Real-to-Sim Alignment

We identify the most important factor for successful sim-to-real transfer of our policy to be motor response. In simulators like IsaacGym[15], the motor response by default is very ideal without much latency. However, the motor response in the real robot is affected by many aspects like communication latency, inertia of the rotor and so on. It's necessary to align the motor response between simulation and reality. Inspired by [16], we propose an alignment method that utilizes real-world data to optimize parameters in simulation. First, we generated sine wave trajectories of frequencies from 1 Hz to 3 Hz as desired joint angles. The desired joint angles would then be applied to actuators of the real robot, with desired joint angles and real joint angles data collected at 200 Hz for 1 minute. Next, we ran 4096 environments in simulation that sampled joint friction, joint damping and joint armature from a specified range. When applying the desired joint angle in collected data, we calculated the cumulative norm e_{joint} between joint angles in simulation and those in reality. Finally, after the whole process we picked the environment with lowest e_{joint} and got the corresponding joint friction, joint damping and joint armature values. To ensure the generalization ability, we selected a small range for each term as a domain randomization term. In this way, we can align the simulation with reality effectively using the real-world data.

Besides, we identified a 20 ms action delay in the real robot system through applying action delay in simulation. All domain randomization terms are listed in Table 3.

To demonstrate the effectiveness of the proposed real-to-sim alignment, we present the motor response trajectory comparison in Fig. 6. We can see that without real-to-sim alignment, the motor response in simulation is very swift

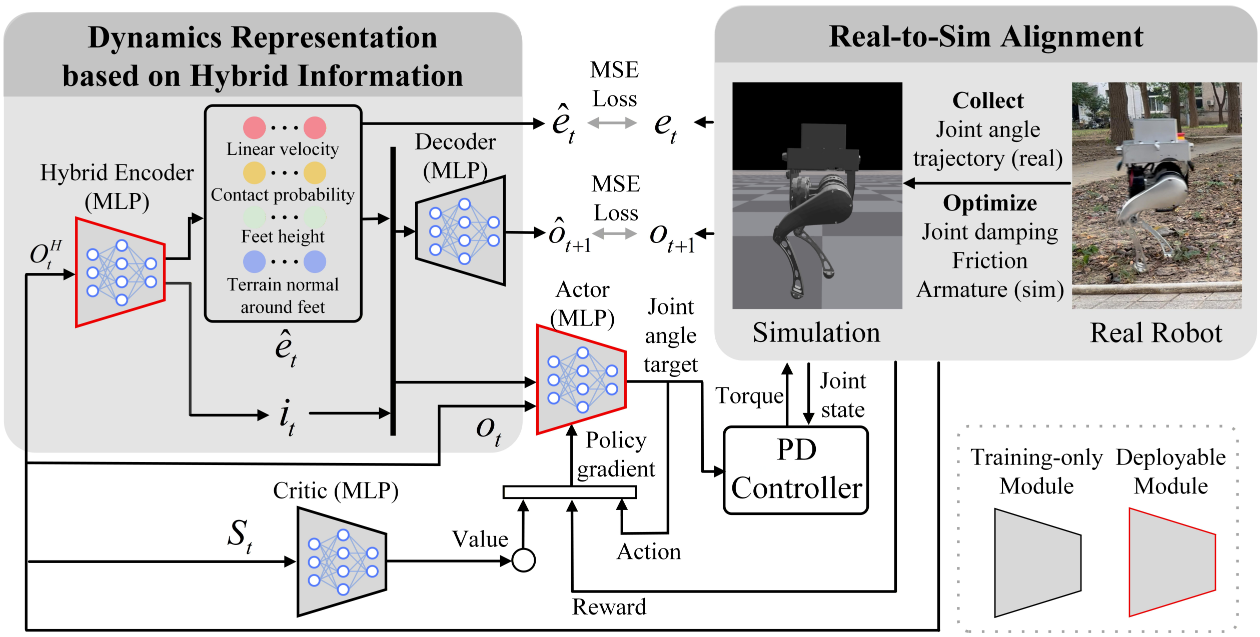


Fig. 5: Diagram of proposed training framework based on RL. There are two essential components of our framework: dynamics representation based on hybrid information and real-to-sim alignment. We integrates enhanced explicit information $\hat{\boldsymbol{e}}_t$ with latent information $\boldsymbol{i}_t$ to represent the underlying dynamics through fitting the future observation. Besides, to enable zero-shot sim-to-real transfer, we optimize joint parameters in simulation using data collected in real-world robot.

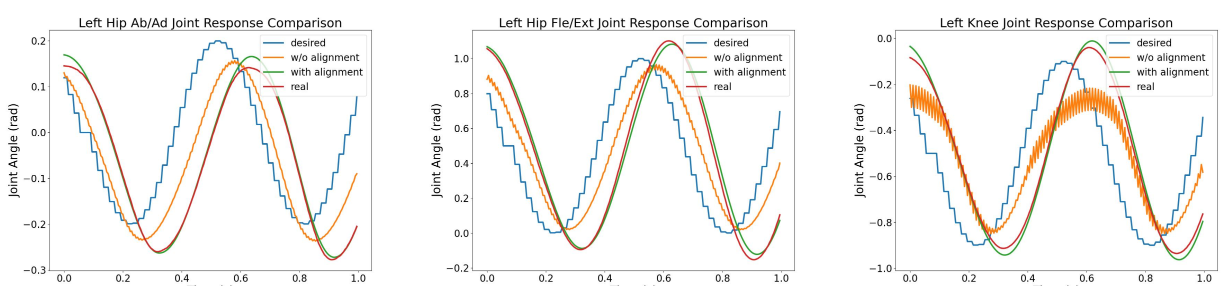


Fig. 6: Comparison of motor response for left joints. Three subfigures correspond to left hip abduction-adduction joint, left hip flexion-extension joint and left knee joint from left to right. Desired angles, simulation joint angles without alignment, simulation joint angles with alignment and real-robot joint angles are plotted to demonstrate the effectiveness of real-to-sim alignment.

Table 3
Domain Randomization

Randomization Term	Range	Unit
Friction	[0.0, 1.6]	-
Payload mass	[-0.5, 2.0]	kg
Pushing robot	[-0.6, 0.6]	m/s
Center of Mass (CoM) bias	[-0.03, 0.03]	m
Joint damping	[0.05, 0.15]	-
Joint friction	[0.015, 0.025]	-
Joint armature	[0.065, 0.075]	-
Joint p gains	[0.8, 1.2] × nominal value	N· rad
Joint d gains	[0.8, 1.2] × nominal value	N· rad/s
Action delay	[0, 20]	ms

without much latency, which has a great gap with that in reality. After this alignment process, the motor response in simulation is effectively aligned with that in reality.

3.4. Implementation Details

3.4.1. Observation and Action

The observation for one time step $\boldsymbol{o}_t$ includes base angular velocity, projected gravity, joint angles, joint angular velocities and last actions. The observation history length H is set to be 5. The full state $\boldsymbol{s}_t$ given to the critic network consists of $\boldsymbol{o}_t$, $\boldsymbol{e}_t$, domain randomization parameters and height point samples around the base link. This setup forms an asymmetric actor critic (A2C)[17] framework, fostering the estimation of value for the critic network. The feet height and terrain normal vectors around feet are calculated using the height point samples around the robot base.

The input of the actor (policy) is the concatenation of current observation $\boldsymbol{o}_t$ and the output of hybrid encoder $[\hat{\boldsymbol{e}}_t, \boldsymbol{i}_t]$. Actions are converted into torques as follows:

$$\boldsymbol{\tau} = K_p(k_a \boldsymbol{a}_t + \boldsymbol{q}_{stand} - \boldsymbol{q}_t) - K_d \dot{\boldsymbol{q}}_t \tag{4}$$

Table 4
Reward Terms with Expressions and Weights

Reward Term	Expression	Weight
	Task	
Tracking forward velocity	$\exp(-4\|\|v_x^* - v_x\|\|_2^2)$	1.0
Tracking lateral velocity	$\exp(-4\|\|v_y^* - v_y\|\|_2^2)$	0.5
Tracking angular yaw velocity	$\exp(-4\|\|\omega_z^* - \omega_z\|\|_2^2)$	0.5
Tracking base height	$\exp(-100\|\|h^* - h\|\|_2^2)$	0.6
Tracking foot clearance	$\exp(-100\ error_{clearance})$	0.6
	Smoothness	
Base vertical velocity	v_z^2	−0.5
Base angular velocity (roll and pitch)	$\|\|\boldsymbol{\omega}_{xy}\|\|_2^2$	−0.05
Base orientation	$\|\|\boldsymbol{g}_{xy}\|\|_2^2$	−3.0
Joint acceleration	$\|\|\ddot{\boldsymbol{q}}\|\|_2^2$	$-2.5e^{-7}$
Action rate	$\|\|\boldsymbol{a}_t - \boldsymbol{a}_{t-1}\|\|_2^2$	$-1e^{-2}$
Action smoothness	$\|\|\boldsymbol{a}_t - 2\boldsymbol{a}_{t-1} - \boldsymbol{a}_{t-2}\|\|_2^2$	$-1e^{-2}$
Torque	$\|\|\boldsymbol{\tau}\|\|_2^2$	$-2e^{-5}$
Joint angle limitation	$1(\boldsymbol{q}_t \notin [\boldsymbol{q}_{min}, \boldsymbol{q}_{max}])$	−1
Hip ab/ad angular velocities	$\|\|\dot{\boldsymbol{q}}_{ab/ad}\|\|_2^2$	$-1e^{-3}$
	Regularization	
Collision	$1_{collision}$	−1
Feet distance	$\max\ (0, 0.18 - \|\|\boldsymbol{p}_{xy}^{left} - \boldsymbol{p}_{xy}^{right}\|\|_2)$	-100.0
Periodic gait	$r_{periodic\ gait}$	1.0

where $\boldsymbol{\tau}$ is the joint torque, K_p is the proportional gain of the PD controller, $\boldsymbol{q}_{stand}$ is the default joint angles, $\boldsymbol{q}_t$ is the joint angles at time step t and K_d is the derivative gain of the PD controller.

In this paper, we set $K_p = 30.0$, $K_d = 1.5$ (larger values compared with $K_p = 25.0$ and $K_d = 0.5$ of BRAVER[18]). In our practice, we find that specifying a large value to K_p can avoid the kneeling behavior of the robot. The leg-foot reset condition mentioned in [18] can be substituted by specifying large K_p values.

3.4.2. Reward Design

Our reward functions are designed based on some previous works[19–21] and can be separated into three parts: task rewards, smoothness rewards and regularization rewards. Task rewards consist of reward functions that encourage the policy to complete specified tasks, including tracking forward velocity, tracking lateral velocity, tracking angular yaw velocity, tracking foot clearance and tracking base height. Smoothness rewards are designed to penalize radical actions of the policy. They comprise base vertical velocity reward, base angular velocity (roll and pitch) reward, base orientation reward, joint acceleration reward, action rate reward, action smoothness reward, torque reward, joint angle limitation reward and hip ab/ad angular velocities reward. Regularization rewards are meant to regularize the way a policy completes tasks. Rewards in this part include collision reward, feet distance reward and periodic gait reward.

The periodic gait reward is constructed based on the same principle introduced in [20], and is defined as follows:

$$r_{periodic\ gait} = \exp(\mathbb{E}[R_{bipedal}(\boldsymbol{s}, \boldsymbol{\Phi})]) \tag{5}$$

$$\mathbb{E}[R_{bipedal}(\boldsymbol{s}, \boldsymbol{\Phi})] = \sum_{feet} (\mathbb{E}[C_{frc}(\boldsymbol{\Phi} + \theta_i)] \cdot n_{i\ frc}(s) + \mathbb{E}[C_{spd}(\boldsymbol{\Phi} + \theta_i)] \cdot n_{i\ spd}(s))$$

where $R_{bipedal}(\boldsymbol{s}, \boldsymbol{\Phi})$ has the same definition as [20] but we substitute $q_{i\ frc}$ and $q_{i\ spd}$ in [20] with $n_{i\ frc}$ and $n_{i\ spd}$ to avoid confusion of symbols. Instead of Von Mises distribution, we utilized step functions to generate contact indicators, which did not lead to performance degradation and facilitated faster training.

Expressions and weights of all reward functions are listed in Table 4, where $\boldsymbol{v}^*$ and $\boldsymbol{v}$ are desired and real linear velocity of the robot base, $\boldsymbol{\omega}^*$ and $\boldsymbol{\omega}$ are desired and real angular velocity of the robot base, h^* and h are desired and real base height, $\boldsymbol{g}$ is the gravity vector projected into base frame and $\boldsymbol{p}$ is the position of the robot's feet in the world frame.

For the tracking foot clearance reward, the error of foot clearance is defined as follows:

$$error_{clearance} = \sum_{feet} (p_i - (max(H_{sample,i}) + p_{des}))^2 ||\boldsymbol{v}_{f,xy,i}||_2$$

where $\boldsymbol{v}_f$ is the feet velocity in the world frame, H_{sample} is the height value of sample points around feet and p_{des} is the desired foot clearance above ground.

3.4.3. Environment Settings

We utilize IsaacGym[15] as the simulator, and implement our framework based on legged gym[19]. The hidden layers are $[256, 256, 128]$ for the actor network, $[512, 256, 128]$ for the critic network, $[256, 128]$ for the hybrid encoder and $[128, 256]$ for the decoder network. Some hyper parameters for training are listed in Table 5. We train 4096 parallel environments on a terrain curriculum including slope, stairs and discrete obstacles, and it takes 5000 iterations for the policy to obtain satisfying performance, which corresponds to 85 minutes of wall clock time using a Nvidia RTX 3080 for training. We set the max episode length to 20 seconds, corresponding to 1000 time steps for a control policy of 50 Hz. Episodes are terminated once the maximum episode length is reached or collision between the robot base and terrains happens.

Table 5
Hyper Parameters for Training

PPO	
Batch size	4096 × 24
Mini-batch size	4096 × 6
Number of epochs	5
Clip range	0.2
Entropy coefficient	0.01
Discount factor	0.99
GAE discount factor	0.95
Desired KL-divergence	0.01
Learning rate	adaptive
Supervised Learning	
Learning rate	$5e^{-4}$
Mini-batch size	4096 × 6
Number of epochs	3

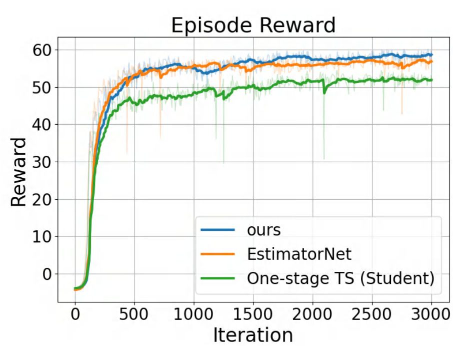
Table 6
Comparison results on different terrains in simulation. Results are averaged over 100 trajectories with different difficulties. Bold numbers indicate the best scores among methods. The velocity tracking error is defined as the sum of the Euclidean norms of the X- and Y-direction velocity errors at each control step within a 1-second window.

		Slope (0-21°)	Stair (0-10cm)	Discrete Obstacles (0-10cm)
Return ↑	ours	$\mathbf{63.54 \pm 2.56}$	$\mathbf{52.35 \pm 17.17}$	$\mathbf{58.02 \pm 12.34}$
	EstimatorNet	62.35 ± 6.44	50.74 ± 18.02	57.77 ± 12.49
	One-stage TS (Student)	58.84 ± 8.93	24.22 ± 19.06	46.68 ± 19.58
Tracking Error ↓	ours	$\mathbf{0.20 \pm 0.03}$	$\mathbf{0.21 \pm 0.07}$	$\mathbf{0.19 \pm 0.04}$
	EstimatorNet	0.21 ± 0.04	$\mathbf{0.21 \pm 0.07}$	0.20 ± 0.04
	One-stage TS (Student)	0.26 ± 0.07	0.25 ± 0.11	0.20 ± 0.09

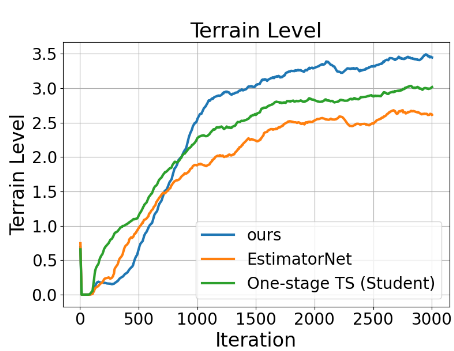
4. Experiments

The policy and hybrid encoder trained through proposed framework were deployed to RDKX5. RDKX5 processed multiple tasks through multi-thread programming written in C++, and the neural network inference was conducted through LibTorch. The inference of single time step took about 12 ms to complete. For lower board DM-MC02, tasks were executed through interrupts.

We have conducted simulation experiments, various indoor and outdoor real-world experiments to demonstrate the performance of Agile-PF.



(a)



(b)

Fig. 7: Comparison of reward values and terrain level during training.

4.1. Simulation

We compared our framework in terms of episode reward, tracking forward velocity reward and terrain level with following two methods:

- **One-stage TS**[22]: Teacher encoder and student encoder are trained in one stage by conducting reinforcement learning and supervised learning concurrently.
- **EstimatorNet**[12]: An explicit estimator is trained to estimate base linear velocity, feet contact probability, feet height and terrain normal around feet.

For a fair comparison, we compare the student policy of One-stage TS with EstimatorNet and our method, which are all deployable. The episode reward and terrain level during training are shown in Fig. 7. Our method achieves highest episode reward among three methods. Besides, our method has the highest terrain level during training, which means that our method can enable the robot to climb higher stairs and steeper slopes. This reveals the advantage of our method by exploiting both explicit and implicit information to achieve stable locomotion on rough terrains. Further more, we compare the episode return and velocity tracking error of three methods, as shown in Table. 6. Our method achieves higher return and smaller velocity tracking error on most terrains.

4.2. Real-world Experiments

In terms of real-world performance, Agile-PF outperforms BRAVER and Bolt in running speed, load capacity and slope traversal ability, as listed in Table 7. We select these two robots to compare with because they are all point-foot biped robots and are similar in size.

4.2.1. High-speed Running

To test the maximal running speed of Agile-PF, we conducted running experiments on an outdoor asphalt road. In the training, the range of linear velocity commands for forward direction is [-0.5, 2.5] m/s. To examine the generalization ability of our policy, we tried to specify larger command values than the maximum value in simulation. As a result, the robot could run at a maximal speed of 3

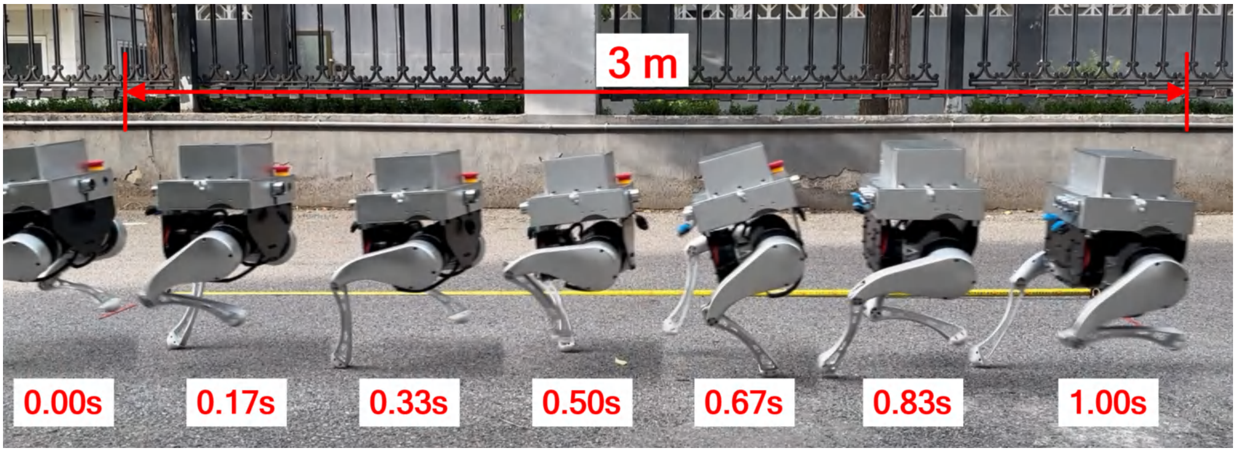


Fig. 8: Keyframes of Agile-PF running at the maximal speed of 3 m/s. It took the robot 1 second to run through a distance of 3 meters, which was a speed of 3 m/s.

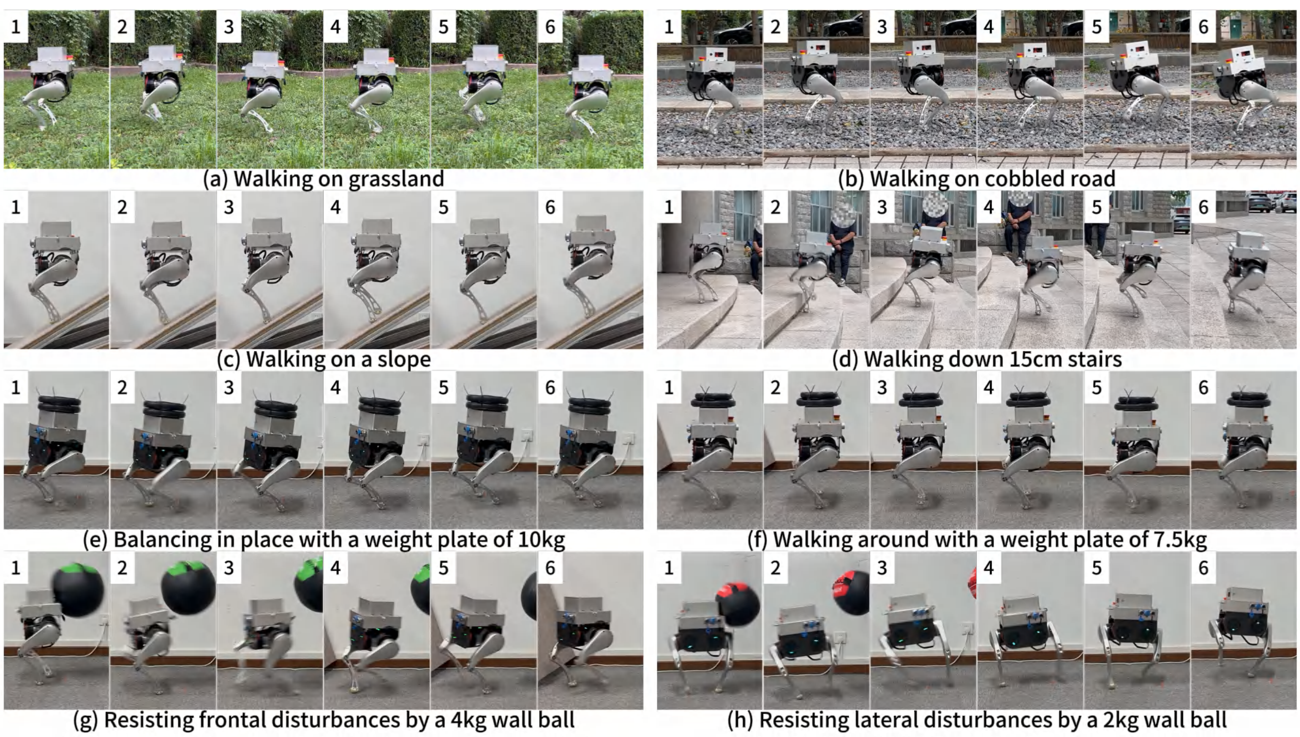


Fig. 9: Snapshots of Agile-PF walking on various terrains, walking with payloads and resisting external disturbances. (a) Walking on grassland. (b) Walking on cobbled road. (c) Walking on a slope of 30°. (d) Walking down 15 cm Stairs. (e) Balancing in place with a weight plate of 10 kg. (f) Walking around with a weight plate of 7.5 kg. (g) Resisting frontal disturbances by a 4 kg wall ball. (h) Resisting lateral disturbances by a 2 kg wall ball.

m/s, shown in Fig. 8. Considering the leg length of 0.4m, the normalized Froude number[23] $v_{max}/\sqrt{gL}$ of Agile-PF corresponds to 1.51. This value is far larger than the ever reported value of the similar platform BRAVER[18] and Bolt[5], demonstrating superior agility of Agile-PF.

4.2.2. *Walking on Various Terrains*

To test the robot's robustness on different terrains, we conducted walking experiments on grasslands, cobbled roads, stairs and slopes. Snapshots of Agile-PF walking on various terrains are shown in Fig. 9 (a)-(d). Our robot could perform stable walking on grasslands and cobbled roads. For slope climbing, the robot could stably climb a slope of 30°, which was nearly twice the value reported on BRAVER[18]. When walking down stairs of 15 cm (44% of its standing height), the robot could not tap down on the next stair using one foot and support itself using the other foot like quadruped robots or biped robots with soles, due to the limitation of its inherent point-foot morphology. However, it could still walk downstairs by jumping down swiftly.

Table 7
Comparison of parameters of Bolt, BRAVER and our robot

	Bolt[5]	BRAVER[18]	Our robot
Mass (kg)	1.34	8.6	13
DoF	6	6	6
Max Speed (m/s)	0.34	2.0	**3.0**
Leg Length (m)	0.4	0.434	0.4
$v_{max}/\sqrt{gL}$	0.54	0.97	**1.51**
Load Capacity (kg)	-	-	**10**
Maximum Slope (°)	-	17.45	**30**

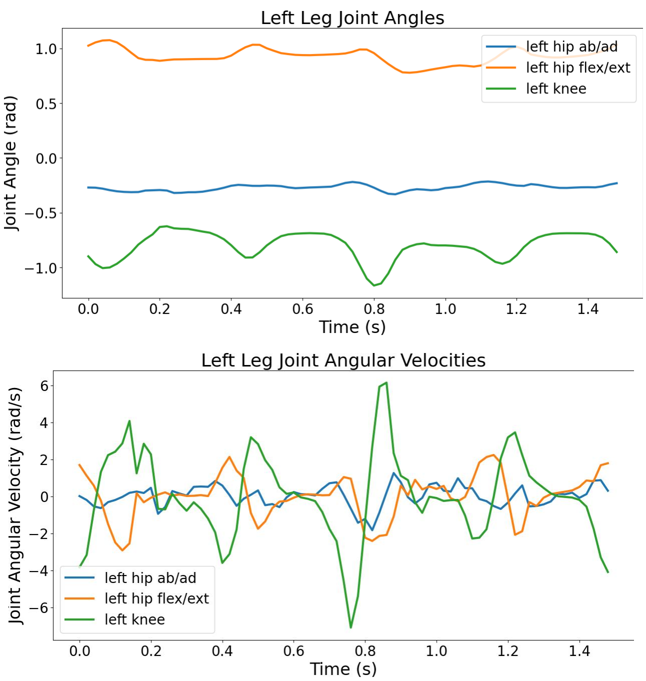


Fig. 10: Joint angles and joint angular velocities curves of the robot's left leg while walking on a cobbled road.

Besides, we present detailed joint angles and angular velocities curves of Agile-PF walking on cobbled roads, shown in Fig. 10. The cobbled road was covered with loose pebbles. As soon as the robot stepped on it, the pebbles became prone to moving, creating disturbances and placing high demands on the robot's balancing ability. Nevertheless, our robot still performed reliable locomotion on this kind of terrain, demonstrating adaptability in the face of challenging terrains.

4.2.3. Load Capacity

For load capacity, we consider static load capacity (load capacity when maintaining balance in place) and dynamic load capacity (load capacity when walking around). Initially, the robot stepped in place to keep balance. We gradually added weight plates and tested the maximal weight it could carry statically and dynamically, as shown in Fig. 9 (e) and Fig. 9 (f). The maximal static load capacity is 10 kg, and the maximal dynamic load capacity is 7.5 kg. The ratios of load to dead weight are 0.77 and 0.58 respectively. Due to the point-foot design, the robot needs to balance itself through continual stepping, therefore the base of the robot swings back and forth when carrying a load of 10 kg. However, the policy can still manage to stabilize the system.

4.2.4. Resistance against External Disturbances

We also tested the robot's capability to resist unexpected external disturbances from frontal and lateral directions. We tied a wall ball to a fixed point with a rope, then lifted the ball and let it fall freely from top to bottom. At the very bottom, the wall ball would collide with the robot, resulting in unexpected external forces. Snapshots of our robot resisting disturbances of frontal and lateral directions are shown in Fig. 9 (g) and Fig. 9 (h). As a result, our robot could resist the force generated by a 4 kg wall ball in frontal direction. In lateral direction, the maximal force the robot could resist without collapsing was generated by a 2 kg wall ball.

5. Conclusion

In this paper, we present the system design and learning-based control of an agile and easy-to-reproduce point-foot biped robot. All components of the robot are off-the-shelf commercial products, lowering the threshold of developing biped robot system. For the control of the robot, we propose a RL-based training framework that combines dynamics representation based on hybrid information with real-to-sim alignment. The dynamics representation endows the policy with robust locomotion ability and the real-to-sim alignment bridges the sim-to-real gap effectively. Using the proposed training framework, the robot demonstrates state-of-the-art ability including high-speed running, walking on diverse terrains, pushing recovery and load capacity.

Future works may include studying perceptive locomotion on the proposed bipedal platform, achieving more difficult terrain traversal by utilizing depth image or lidar information. Besides, we can equip this platform with other forms of feet and robotic arms to handle loco-manipulation tasks.

References

[1] Qiayuan Liao, Takara E. Truong, Xiaoyu Huang, Guy Tevet, Koushil Sreenath, and C. Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion, 2025.

[2] Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning, 2024.

[3] Zhen Wu, Xiaoyu Huang, Lujie Yang, Yuanhang Zhang, Koushil Sreenath, Xi Chen, Pieter Abbeel, Rocky Duan, Angjoo Kanazawa, Carmelo Sferrazza, Guanya Shi, and C. Karen Liu. Perceptive humanoid parkour: Chaining dynamic human skills via motion matching, 2026.

[4] Zhengguo Zhu, Weiliang Zhu, Guoteng Zhang, Teng Chen, Yibin Li, Xuewen Rong, Rui Song, Daoling Qin, Qiang Hua, and Shugen Ma. Design and control of braver: a bipedal robot actuated via proprioceptive electric motors. *Autonomous Robots*, 47(8):1229–1243, Dec 2023.

[5] Constant Roux, Elliot Chane-Sane, Ludovic De Matteïs, Thomas Flayols, Jérôme Manhes, Olivier Stasse, and Philippe Souères. Constrained reinforcement learning for unstable point-feet bipedal locomotion applied to the bolt robot, 2025.

[6] Yufeng Chi, Qiayuan Liao, Junfeng Long, Xiaoyu Huang, Sophia Shao, Borivoje Nikolic, Zhongyu Li, and Koushil Sreenath. Demonstrating berkeley humanoid lite: An open-source, accessible, and customizable 3d-printed humanoid robot, 2025.

[7] Arne-Christoph Hildebrandt, Simon Schwerd, Robert Wittmann, Daniel Wahrmann, Felix Sygulla, Philipp Seiwald, Daniel Rixen, and Thomas Buschmann. Kinematic optimization for bipedal robots: a framework for real-time collision avoidance. *Autonomous Robots*, 43(5):1187–1205, Jun 2019.

[8] Petar Kormushev, Barkan Ugurlu, Darwin G. Caldwell, and Nikos G. Tsagarakis. Learning to exploit passive compliance for energy-efficient gait generation on a compliant humanoid. *Autonomous Robots*, 43(1):79–95, Jan 2019.

[9] Matthew Chignoli, Donghyun Kim, Elijah Stanger-Jones, and Sangbae Kim. The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 1–8, 2021.

[10] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[11] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020.

[12] Gwanghyeon Ji, Juhyeok Mun, Hyeongjun Kim, and Jemin Hwangbo. Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robotics and Automation Letters*, 7(2):4630–4637, 2022.

[13] I Made Aswin Nahrendra, Byeongho Yu, and Hyun Myung. Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5078–5084, 2023.

[14] Kamal Berahmand, Fatemeh Daneshfar, Elaheh Sadat Salehi, Yuefeng Li, and Yue Xu. Autoencoders and their applications in machine learning: a survey. *Artificial Intelligence Review*, 57(2):28, Feb 2024.

[15] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu based physics simulation for robot learning. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[16] Yunfei Li, Jinhan Li, Wei Fu, and Yi Wu. Learning agile bipedal motions on a quadrupedal robot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9735–9742, 2024.

[17] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[18] Daoling Qin, Guoteng Zhang, Zhengguo Zhu, Teng Chen, Weiliang Zhu, Xuewen Rong, Anhuan Xie, and Yibin Li. A heuristics-based reinforcement learning method to control bipedal robots. *International Journal of Humanoid Robotics*, 21(03):2350013, 2024.

[19] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 91–100. PMLR, 08–11 Nov 2022.

[20] Jonah Siekmann, Yesh Godse, Alan Fern, and Jonathan Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7309–7315, 2021.

[21] Hongxi Wang, Haoxiang Luo, Wei Zhang, and Hua Chen. Cts: Concurrent teacher-student reinforcement learning for legged locomotion, 2024.

[22] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning, 2022.

[23] R McNeill Alexander. *Principles of animal locomotion*. Princeton university press, 2003.

Yasen Jia received the B.E. degree from Mechatronic Engineering, Beijing Institute of Technology, China. He is now working for the M.E. degree at Beijing Institute of Technology. His research work focuses mainly on deep reinforcement learning and legged locomotion.