Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Efficient learning of robust multigait quadruped locomotion for minimizing the cost of transport*#

Zhicheng WANG†[1,2], Xin ZHAO†‡[3], Meng Yee (Michael) CHUAH[2], Zhibin LI[4], Jun WU[1,5], Qiuguo ZHU[1,5]

[1]*Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China*
[2]*Institute for Infocomm Research, A*STAR, Singapore 138632, Singapore*
[3]*Chinese Scholartree Ridge State Key Laboratory, China North Vehicle Research Institute, Beijing 100072, China*
[4]*Department of Computer Science, University College London, London WC1E, UK*
[5]*State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China*
†E-mail: willw317@zju.edu.cn; pekingbit@163.com

Received Dec. 16, 2024; Revision accepted May 12, 2025; Crosschecked Aug. 29, 2025

Abstract: Quadruped robots are able to exhibit a range of gaits, each with its own traversability and energy efficiency characteristics. By actively coordinating between gaits in different scenarios, energy-efficient and adaptive locomotion can be achieved. This study investigates the performances of learned energy-efficient policies for quadrupedal gaits under different commands. We propose a training–synthesizing framework that integrates learned gait-conditioned locomotion policies into an efficient multiskill locomotion policy. The resulting control policy achieves low-cost smooth switching and controllable gaits. Our results of the learned multiskill policy demonstrate seamless gait transitions while maintaining energy optimality across all commands.

Key words: Reinforcement learning; Locomotion; Motor learning; Energy efficiency
https://doi.org/10.1631/FITEE.2401070 **CLC number:** TP242.6; TP18

1 Introduction

Quadruped robots have successfully navigated complex environments using various control approaches, but their adaptability and efficiency still fall short compared to biological animals. In addition to differences in body structures, a significant reason is that animals can easily adopt the most suitable gait pattern and frequency and switch between gaits rapidly, smoothly, and robustly. Different gaits enable animals to effectively handle diverse terrain conditions at different speeds while maintaining good energy efficiency (Hildebrand, 1965; Hoyt and Taylor, 1981). As quadruped robots continue to be deployed in complex and unstructured environments, they will inevitably encounter new challenges that require emulating the strategies used by their natural counterparts.

Controllable gaits and active switching capabilities offer significant advantages for quadruped robot control (Haynes and Rizzi, 2006; Hsiao-Wecksler et al., 2010). By incorporating higher-level decision inputs, robots can activate different gaits in real time (Xi et al., 2016). Moreover, these capabilities enable robots to go beyond locomotion and perform tasks

‡ Corresponding author
* Project supported by the "Leading Goose" R&D Program of Zhejiang Province (No. 2023C01177), the Research Project on the Motion Control of Bipedal Maneuver (No. 8KD006(2024)-2), the National Key R&D Program of China (No. 2022YFB4701502), and the 2035 Key Technological Innovation Program of Ningbo City (No. 2024Z300)
# Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2401070) contains supplementary materials, which are available to authorized users
ORCID: Zhicheng WANG, https://orcid.org/0000-0002-2657-7591

such as dancing and leaping using specially designed gait controllers (Margolis and Agrawal, 2022).

However, the integration of multiple gaits to improve adaptability has been largely unexplored in existing research. Most existing frameworks either restrict locomotion to a single predefined gait type or disregard gait controllability altogether by leaving gait selection to the policy. Although this approach prioritizes simplicity and ease of tuning, it sacrifices optimality and controllability. As manual fine-tuning is typically required for multigait integration, many researchers opt not to explore this avenue.

Methods that emphasize gait-conditioned locomotion controllers and gait transitions, such as those utilizing central pattern generators, often rely on heuristics-based reference trajectories and constructed oscillators (Iscen et al., 2018; Shao et al., 2022; Tan DCH et al., 2023). However, these approaches often introduce additional parameters that require time-consuming manual tuning, which imposes limitations on their performance.

As shown in Fig. 1, in this study, we investigate learned gait-conditioned locomotion policies and present a framework that addresses the considerations of energy efficiency, robustness, and robot hardware safety. Our goal is to develop an efficient multiskill locomotion policy that seamlessly integrates these different gait policies while maintaining

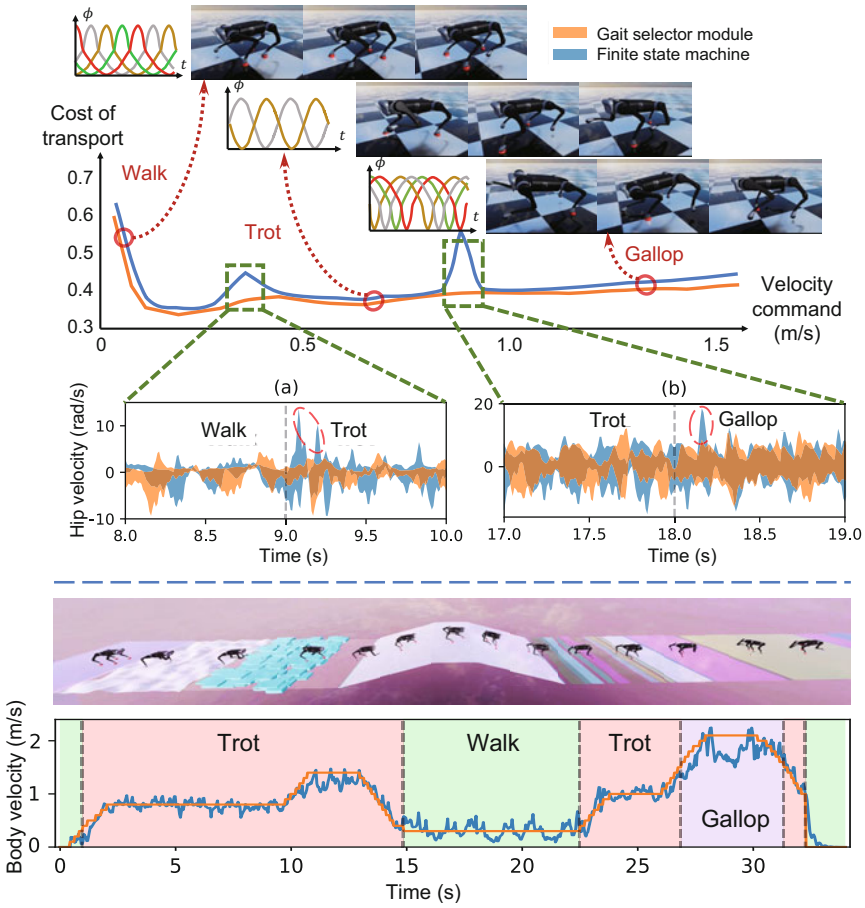

Fig. 1 Efficient gait-conditioned locomotion learning framework. Top: single skill learning, analysis, and integration. Single-gait policies are trained to follow corresponding phase command signals and velocity commands. Through the use of reinforcement learning, a gait selector module can be trained. The performance of the proposed policy (orange plots) demonstrates a smoother transition than the state machine (blue plots) as shown in joint velocity plots (a) and (b). The vertical dotted lines indicate the gait transitions, and the undesired large spikes caused by improper switching schemes are marked by dashed red circles. Bottom: The proposed multigait policy is robust enough to traverse complex environments. The snapshot and plot demonstrate the proposed policy deployed on a simulated robot traversing different terrains with gait transitions. References to color refer to the online version of this figure

a balanced approach to the factors mentioned above. By achieving smooth transitions between the various policies, our framework aims to optimize the overall performance and effectiveness in quadruped locomotion tasks.

The main contributions of this work are as follows: (1) development of a unified framework capable of training diverse and robust gait-conditioned locomotion policies; (2) proposal of a safe and cost-effective architecture for integrating trained gait policies to form an adaptive multimodal locomotion skill with smooth transition; (3) testing of the proposed policies in simulation environments and analysis of their performance, which demonstrates their optimality.

2 Related works

2.1 Privileged learning-based locomotion

The base states of a robot, especially body velocity, are vital for achieving robust locomotion. However, estimators based on proprioceptive observations can be very noisy and inaccurate. To overcome this challenge, learning-based approaches that leverage privileged information during training have emerged as a powerful tool (Chen et al., 2019). By using precise states available in simulation, these approaches enable more effective learning.

Since Lee et al. (2020) made the first successful implementation and Kumar et al. (2021), Fu et al. (2021), and Miki et al. (2022) achieved reference-free sim-to-real transfer, privileged training has been widely adopted in learning-based locomotion research (Agarwal et al., 2022; Kumar et al., 2022; Margolis and Agrawal, 2022; Margolis et al., 2022; Loquercio et al., 2023). Motivated by these advancements, our work aims to combine privilege training with gait regulation to develop an efficient and robust multigait policy.

2.2 Gait-conditioned locomotion

Legged locomotion has been a long-standing area of research. The gait-conditioned policy can be obtained through various approaches, such as learning gait library residuals (Iscen et al., 2018; Siekmann et al., 2020; Kumar et al., 2021; Xie et al., 2021), imitation learning (Peng et al., 2020; Tsounis et al., 2020; Jin et al., 2022), and reward design. One common approach for reward formulation is penalizing contact state errors to encourage tracking a predefined contact reference (Tan J et al., 2018; Acero et al., 2022; Shao et al., 2022). Bio-inspired heuristic policies can be used to generate smoother transitions between different gaits (Iscen et al., 2018; Shao et al., 2022). Siekmann et al. (2021b) introduced phase-dependent rewards that penalize contact force and foot velocity to ensure proper swing–stance cycles, leading to the development of gait-conditioned policies for legged robots (Siekmann et al., 2020, 2021a; Shao et al., 2022). Fu et al. (2021) demonstrated that indirect energy rewards can also lead to the emergence of gaits, although they may not strictly follow explicit gait patterns.

However, the energy efficiency of these learned gaits has not been extensively explored in existing works. In this study, we focus on investigating the energy efficiency of different learned gaits and identifying the optimal gait under various velocity commands. We aim to provide insights into the energy characteristics of these gaits and determine the most suitable gait for different locomotion requirements.

2.3 Motor skill integration

Skill synthesis plays a critical role in achieving multigait locomotion. One common approach is to design finite state machines (FSMs) and activate the desired controller based on specific conditions (Lee et al., 2019). However, this approach can result in brittle behaviors during switching. Alternative synthesizing methods, including integrating motion reference datasets into neural policies using techniques such as multiplicative compositional policies (Peng et al., 2019), and latent adversarial methods (Luo et al., 2020; Peng et al., 2021, 2022) have also been proposed. Other approaches involve fusing skills into a single policy, such as policy distillation (Hinton et al., 2015; Fu et al., 2021), mixture of experts (Jacobs et al., 1991), and expert policy mixture (Zhang et al., 2018; Yang et al., 2020).

Our study uses a trained gait selector module that uses velocity commands and terrain information to decide gait parameters. We have observed that parameter-level switching is sufficient to provide robust and adaptive integrating behavior, without having to distill multiple policies with divergent behavior patterns.

3 Methodology

3.1 Control architecture

In our framework, the control system includes a group of learning-based single-gait policies, a trained gait selector module, and a gait phase generator, as depicted in Fig. 2. The details are explained in Section 3.2.

To enable efficient training, we refer to the architecture proposed in Nahrendra et al. (2023) when training the single-gait policy. A single-gait policy with gait type j, denoted as σ_j, consists of a variational autoencoder (μ_j, η_j) to predict the proprioceptive observation, compress the proprioceptive observation history into latent encodings $\boldsymbol{z}$, and explicitly estimate the robot's linear velocity $\hat{\boldsymbol{v}}$; a backbone policy ρ conditions joint-level position targets on the latest proprioceptive states, estimates, and encodings.

The input of a single-gait policy is composed of the linear velocity command $\boldsymbol{v}_t^*$, gait phase vector $\boldsymbol{g}(t)$, an h-step proprioceptive state history $\boldsymbol{s}_{t:t-h+1}$, and the previous action $\boldsymbol{a}_{t-1}$. The proprioceptive state $\boldsymbol{s}$ refers to the available sensor states of real-world robots, including gravity vector $\boldsymbol{k}$, body angular velocity $\boldsymbol{\omega}$, joint space position $\boldsymbol{q}$, and velocity $\dot{\boldsymbol{q}}$. The action space of a single-gait policy consists of a 12-dimensional joint position command. The single-gait policies run at 50 Hz. A low-stiffness 1 kHz proportional–derivative (PD) controller is used to track these joint position commands while maintaining compliance.

To achieve automatic gait switching, a gait selector module ψ is trained with reinforcement learning (RL). During training, the single-gait policies are frozen. The gait selector module computes gait parameters $\boldsymbol{\theta}_t$ and a one-hot vector $\boldsymbol{w}$. Its input is composed of the height map around the robot $\boldsymbol{H}$, the linear velocity command $\boldsymbol{v}_t^*$, the proprioceptive history $\boldsymbol{s}_{t:t-h+1}$, and the previous action $\boldsymbol{\theta}_{t-1}$. The gait selector module runs at 5 Hz.

The formulation of the control system can be described by the following equations:

$$[\boldsymbol{w}, \boldsymbol{\theta}] = \psi(\boldsymbol{v}^*, \boldsymbol{H}, \boldsymbol{s}_{t:t-h}, \boldsymbol{\theta}_{t-1}), \tag{1}$$

$$[\boldsymbol{z}, \hat{\boldsymbol{v}}] = \mu_j(\boldsymbol{s}_{t-1:t-h}), \tag{2}$$

$$\hat{\boldsymbol{s}}_{t+1} = \eta_j(\boldsymbol{s}_{t-1:t-h}), \tag{3}$$

$$\boldsymbol{a}_{j,t} = \rho_j(\boldsymbol{z}, \hat{\boldsymbol{v}}, \boldsymbol{v}^*, \boldsymbol{g}_{\boldsymbol{\theta}(t)}, \boldsymbol{s}_t, \boldsymbol{a}_{j,t-1}), \tag{4}$$

$$\boldsymbol{a}_t = \sum_{j=1}^{N} w_j \cdot \boldsymbol{a}_{j,t}. \tag{5}$$

3.2 Gait formulation

The choice of gaits in legged locomotion involves a trade-off between energy consumption and traversability, as different gaits prioritize different performance aspects. In general, a longer flight phase within a given period allows the policy to achieve a higher velocity but at the cost of an increased energy consumption.

To achieve controllable gait patterns within a unified framework, we design a phase generator to generate the gait phase vector as an input component of the policies. Inspired by central-pattern-generator-based works (Ijspeert, 2008; Iscen et al., 2018), we adopt periodic signals based on sine waves to specify gait phase commands. Fig. 3 illustrates the contact sequences based on gait phase signals for

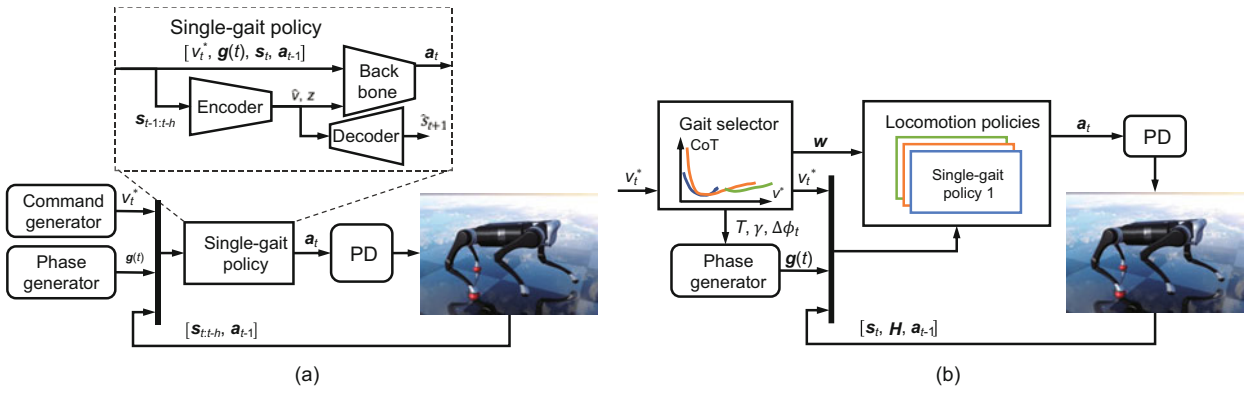

Fig. 2 Main control architecture: (a) training the single-gait policy; (b) formulation of the multiskill locomotion policy. The notations are explained in Section 3.1

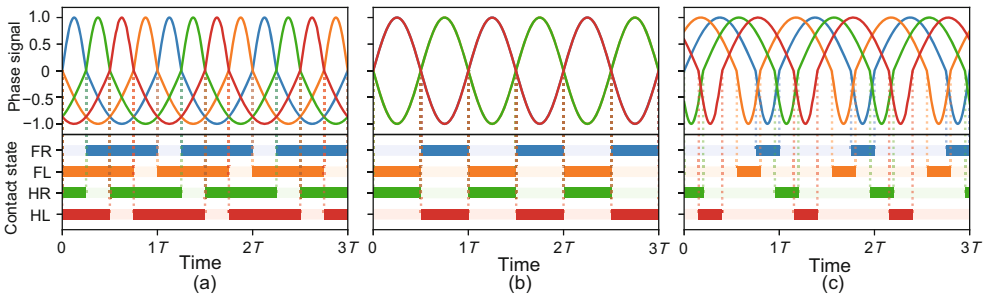

Fig. 3 Gait phase signals and foot contact sequence of different gaits: (a) quasi-static walk; (b) trot; (c) gallop

typical gaits. By modulating the period and offset of every leg, the robot can learn to perform any desired gait. This approach provides flexibility in designing gaits for other performance considerations, as energy consumption may not be the only consideration in challenging environmental conditions.

The phase generator adopted in our implementation can be formulated as

$$\boldsymbol{g}_{\boldsymbol{\theta}(t)} = [\boldsymbol{g}_0, \boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3]^{\mathrm{T}}, \tag{6}$$

$$\boldsymbol{\theta} = [\Delta\phi_0, \cdots, \Delta\phi_3, \gamma_0, \cdots, \gamma_3, T_0, \cdots, T_3], \tag{7}$$

$$\boldsymbol{g}_i = [\sin\phi_i, \cos\phi_i]^{\mathrm{T}}, i = 0, 1, 2, 3, \tag{8}$$

$$\phi_i = \begin{cases} \frac{\pi}{\gamma_i T_i} t + \Delta\phi_i, & 0 \le t < \gamma_i, \\ \frac{\pi t}{(1-\gamma_i)T_i} + \frac{1-2\gamma_i}{1-\gamma_i}\pi + \Delta\phi_i, & \gamma_i \le t < T_i, \end{cases} \tag{9}$$

where $\boldsymbol{g}_{\boldsymbol{\theta}(t)}$ is the periodic phase command generated by the generator, a function of time in a period t and gait parameter $\boldsymbol{\theta}$, T_i is the period of the current gait, γ_i is the duty cycle of the specified gait, representing the proportion of the flight phase within a period, $\Delta\phi_i$ is the phase offset of the corresponding gait, ϕ_i is the linear phase indicator for foot i, and $\boldsymbol{g}_i$ is the trigonometric phase signal for foot i. Two state channels per foot are designed to guarantee that every time point can correspond to only one phase state within a given period. Note that π here is the mathematical constant instead of the policy.

To demonstrate the multigait capability of our proposed framework, we select quasi-static walk, dynamic trot, and gallop. These gaits possess diverse features that can address most basic locomotion demands.

3.3 Policy training

3.3.1 RL approach

The state of the robot at a specific time is constrained by the previous state and the action taken by the robot; therefore, both the gait-conditioned locomotion control and the gait decision can be described by a Markov decision process, which is suitable for RL. The action performed by the robot using its policy influences the probability distribution of the state transition, and the result of the transition leads to a corresponding reward value, which indicates how successful the state is. Hence, optimizing the performance of the controller is equivalent to maximizing the total reward over an infinite horizon. In our implementation, the single-gait policy and the gait selector module are trained with proximal policy optimization (PPO) (Schulman et al., 2017). Parameters related to PPO and the policy architecture can be found in Table 1.

Table 1 Training parameters

Parameter	Value
Parallel number	4096
Discount factor	0.995
Episode length (s)	8
Maximum payload	[–2.0, 4.0]
Friction range	[0.2, 1.0]
Motor strength	[0.8, 1.2]
Number of backbone hidden layers	256, 128
Number of encoder hidden layers	512, 128
Number of decoder hidden layers	128, 64
Number of gait selector hidden layers	256, 256, 64
Observation history length	25

3.3.2 Single-gait reward design

Our reward terms, which focus more on periodic gait behavior and energy efficiency, can be categorized into three distinct classes: task-oriented, gait-conditioned, and efficiency-related rewards. The gait-related reward design is based on Siekmann et al. (2021b). We use the cost of transport (CoT) as a representation of energy consumption. The reward

function is formulated as follows:

$$R_{\text{velocity}} = F_{\alpha_1,\beta_1}(||\boldsymbol{v}^* - \boldsymbol{v}||), \tag{10}$$

$$R_{\text{orientation}} = F_{\alpha_2,\beta_2}(||\boldsymbol{k}^* - \boldsymbol{k}||), \tag{11}$$

$$R_{\text{height}} = F_{\alpha_3,\beta_3}(|p_z^* - p_z|), \tag{12}$$

$$R_{\text{swing}} = F_{\alpha_4,\beta_4}\Big(\sum_i C_i||\boldsymbol{F}_i||\Big), \tag{13}$$

$$R_{\text{stance}} = F_{\alpha_5,\beta_5}\Big(\sum_i (1-C_i)||\boldsymbol{v}_{\text{f},i}||\Big), \tag{14}$$

$$R_{\text{energy}} = F_{\alpha_6,\beta_6}\left(\frac{|\dot{\boldsymbol{q}}^{\text{T}}\boldsymbol{\tau}|}{mg||\boldsymbol{v}||}\right), \tag{15}$$

$$R_{\text{smooth}} = F_{\alpha_7,\beta_7}(||\boldsymbol{\tau} - \boldsymbol{\tau}_{\text{prev}}||), \tag{16}$$

$$R_{\text{jointvel}} = F_{\alpha_8,\beta_8}(||\dot{\boldsymbol{q}}||), \tag{17}$$

$$F_{\alpha_j,\beta_j}(x) = \alpha_j \text{e}^{-\beta_j x^2},\ j = 1, 2, ..., 8, \tag{18}$$

$$C_i = \begin{cases} 1, \phi_i > 0, \\ 0, \phi_i \le 0, \end{cases} \tag{19}$$

where $i \in \{\text{FR}, \text{FL}, \text{HR}, \text{HL}\}$ is the foot index, C_i is the flight phase coefficient indicating whether the corresponding foot i should be in the swing phase, $\boldsymbol{F}_i$ is the contact force on foot i, $\boldsymbol{v}_{\text{f},i}$ is the world-frame velocity of foot i, $\boldsymbol{\tau}$ is the torque command applied on joints, and $F_{\alpha_j,\beta_j}(x)$ is a Gaussian kernel applied to each term to ensure reward survival. The coefficient α_j represents the importance coefficient related to the weight of the term in the final reward composition, and β_j is the scaling coefficient, which is related to the physical meaning of the error term. CoT is computed by summing the absolute joint powers to represent the total energy consumption.

3.3.3 Gait selector reward design

Aimed at selecting the optimal gait policy and parameters, while preventing the gait selection from oscillating too much, the reward function is defined as follows:

$$R_{\text{tracking}} = F_{\alpha_9,\beta_9}(||\boldsymbol{v}^* - \boldsymbol{v}||), \tag{20}$$

$$R_{\text{survival}} = -\alpha_{10} K_{\text{fail}}, \tag{21}$$

$$R_{\text{decision}} = -\alpha_{11}||\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{prev}}||, \tag{22}$$

$$R_{\text{safety}} = -\alpha_{12}\max||\dot{\boldsymbol{q}}||, \tag{23}$$

$$R_{\text{CoT}} = -\alpha_{13}\frac{|\dot{\boldsymbol{q}}^{\text{T}}\boldsymbol{\tau}|}{mg||\boldsymbol{v}||}, \tag{24}$$

where K_{fail} is the number of failing times between two timesteps, and $\boldsymbol{\theta}_{\text{prev}}$ represents the gait parameters given by the gait selector at the last timestep.

3.3.4 Domain randomization

To facilitate adaptation to a wide range of tasks, we adopt natural constraints and randomization, including terrain friction, motor strength, and payload, for better sim-to-real transfer, with reference to Kumar et al. (2021). The detailed ranges are given in Table 2.

Table 2 Domain randomization parameters

Parameter	Range
Payload mass (kg)	[–2.0, 4.0]
Friction coefficient	[0.3, 1.7]
Motor strength	[0.8, 1.2]
PD factor	[0.9, 1.1]
Latency (ms)	[0, 20]

3.3.5 Curriculum training

A game-inspired curriculum training approach is used to ensure the learning of gait-conditioned policies. The same curriculum terrain is applied to both single-gait training and gait selector training. Detailed parameters can be found in Table 3.

Table 3 Challenge course configuration

Parameter	Value		
	Walk	Trot	Gallop
Start padding length (m)	1	1	2
Length per region (m)	4	5	6
Fractal terrain height (m)	0.15	0.12	0.08
Platform maximum height (m)	0.08	0.08	0.04
Platform density (m^{-2})	40	40	40
Platform size (m)	0.15	0.15	0.15
Slope inclination (deg)	20	15	10
Gap maximum width (m)	0.12	0.12	0.12
Gap minimum interval (m)	0.12	0.12	0.12
Maximum mass payload (kg)	4.0	4.0	4.0
Friction range	[0.2, 1.0]	[0.2, 1.0]	[0.2, 1.0]
Projectile mass (kg)	0.1	0.1	0.1
Projectile velocity (m/s)	40	40	40

4 Experimental results

4.1 Experimental setup

In our work, we use the Jueying Lite3 and Unitree A1, both small-size 12-degree-of-freedom quadruped robots. All processes are run on a laptop equipped with an Intel Core i7-11800H and an NVIDIA GeForce RTX 3070. The training environments are implemented in Isaac Gym (Makoviychuk

et al., 2021), simulations are performed in a realistic simulator, RaiSim (Hwangbo et al., 2018), and the policy architecture is built with PyTorch (Paszke et al., 2019). To ensure reproducibility, the random seeds are set based on the system time upon process initialization.

To evaluate the performance of the trained policy and the proposed gait selector module, we set up comparison groups against the proposed method.

Single-gait policies: To demonstrate the performance of single-gait policies, we apply two typical gait frequencies to every policy for evaluation. All policies are trained with the same parameter configuration, as listed in Table 4. To guarantee reproducibility, we conduct 20 independent training runs per gait, each initialized with a different random seed. All reported metrics are obtained by averaging over these 20 trials.

Table 4 Typical gait parameters

Gait type	$\Delta\phi$	Duty cycle	Nominal frequency (Hz)
Walk	$[0, 0.5\pi, 1.5\pi, \pi]^{\mathrm{T}}$	0.25	0.8
Trot	$[0, \pi, \pi, 0]^{\mathrm{T}}$	0.5	1.8
Gallop	$[0, 1.6\pi, 0.8\pi, 0.4\pi]^{\mathrm{T}}$	0.75	2.8

Finite state machine (FSM): Given a user command v^*, FSM chooses the policy with the best energy efficiency. The gait frequency is fixed to a medium frequency. The FSM switching logic is listed in Table 5.

Table 5 FSM gait switching scheme

Gait type	Frequency (Hz)	Velocity range (m/s)
Walk	0.8	[0, 0.38)
Trot	1.2	[0.38, 0.85)
Gallop	2.4	[0.85, 2.0)
Gallop	3.2	[2.0, 3.0)

Meanwhile, to compare the energy efficiency with that of existing methods, the proposed method is compared with the following methods:

Emergent gait (EG) distillation (Fu et al., 2021): Instead of using explicit gait signal commands, this method uses energy reward to train policies with different gaits.

Convex model predictive control (MPC) (di Carlo et al., 2018): This method is a model-based convex MPC controller. It does not explicitly optimize energy terms.

4.2 Evaluation metrics

4.2.1 RMSE

To evaluate the task performance, we measure the average velocity tracking the root mean squared errors (RMSEs) under typical velocity commands on flat terrain over a duration of 40 s. The velocity commands, chosen based on related works (di Carlo et al., 2018; Fu et al., 2021), include 0.375, 0.9, and 1.5 m/s. The corresponding velocity tracking errors are represented as RMSE(Δv_{L}), RMSE(Δv_{M}), and RMSE(Δv_{H}), respectively.

4.2.2 Energy efficiency

Energy efficiency is assessed using the average CoT, denoted as $\mathrm{CoT}_{\mathrm{avg}}$, under feasible velocity commands. Feasibility is defined as the ability of the policy to operate for >40 s without a failure on flat ground. Velocity commands are sampled at an interval of 0.02 m/s, and each trial is conducted for a duration of 40 s. To simulate real-world conditions, a Gaussian noise with a standard deviation of 0.1 m/s is added to the constant velocity command in each trial.

4.2.3 Robustness

Robustness is determined by the traversing success rate, denoted as TSR, over 20 trials on terrains of 10 m length representing common challenges. As demonstrated by Fig. 4, these terrains include fractal terrain with a height standard deviation of 0.08 m, gaps with a maximum length of 0.1 m, discrete terrain with a maximum height difference of 0.1 m, slopes of 20°, and projectile impacts, where a 0.1 kg sphere is shot at 40 m/s from a random direction every 0.5 s. In our implementation, failure is defined as the inability to cross the terrain within 60 s. The velocity commands for each trial are randomly sampled from the feasible range, ensuring diverse testing scenarios.

4.2.4 Safety

Safety is evaluated based on the ratio between the maximum joint velocity during the first second after switching and that during the next second. A larger maximum joint velocity ratio suggests that the transition behavior is more violent than the normal limit cycle. We collect data from 2000 switchings

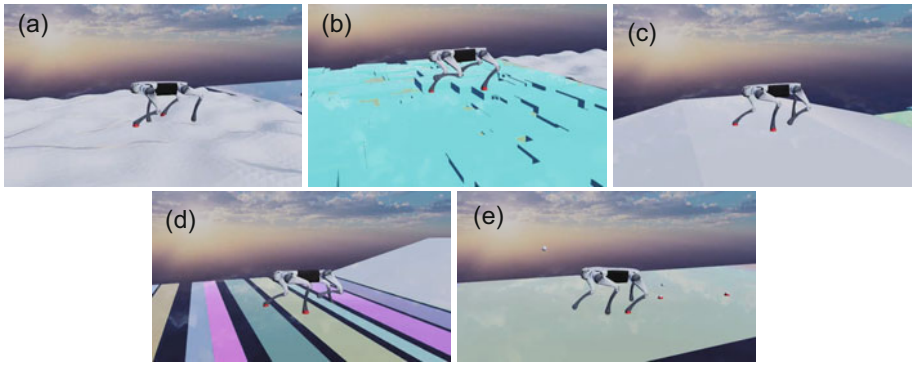

Fig. 4 Snapshots of different regions in the testing course: (a) fractal terrain; (b) random platforms; (c) sloped ground; (d) region with gaps; (e) projectile impact region

happening at random times. This test applies only to the policies with explicit gait transition, including the proposed method and FSM.

4.3 Gait-conditioned policies

4.3.1 Feasibility region evaluation for single-skill policies

The performance of the single-gait policies is summarized in Table 6 and Fig. 5. These results demonstrate that all single-gait policies are capable of effectively tracking velocity and gait command signals, highlighting the robustness of our training framework in enabling the policies to master locomotion skills.

Each single-gait policy exhibits trade-off across different metrics. Walk policies excel in energy efficiency under low-velocity commands and demonstrate superior robustness, but they cannot track high-velocity commands. Conversely, gallop policies achieve high-speed locomotion but exhibit reduced robustness and increased energy consumption. These trade-offs underline the inherent limitations of single-gait strategies and the potential benefits of combining them in multigait policies.

To ensure that the policies are applied under appropriate circumstances, we need to determine the feasible working velocity command range of each policy. We consider the policy to be reliable when the failure rate, mean velocity command tracking error, and mean contact state error are all below manually set thresholds. These results are also affected by the gait frequency and command velocities.

Figs. 6–8 show the performances of learned policies under different conditions. The feasible command ranges of all policies under different frequencies can be summarized from them. Generally, we can conclude that higher gait frequencies allow the policies to achieve a wider velocity range, and that the trot gait policies have the widest feasible range compared with other gait policies. Specifically, in Fig. 8, we disqualify low-speed gallop because it becomes grounded and inconsistent with the command phase signal for lower-velocity targets.

Table 6 Performance summary under different conditions

Parameter	Walk		Trot		Gallop		FSM	Ours
	0.8 Hz	1.6 Hz	1.2 Hz	2.4 Hz	2.0 Hz	3.2 Hz		
RMSE(Δv_{L})	0.1034	**0.0920**	0.0954	0.0982	0.0966	0.0941	0.1125	0.0929
RMSE(Δv_{M})	0.3768	0.1430	0.1365	0.0860	0.0621	0.0625	0.0827	**0.0620**
RMSE(Δv_{H})	0.8916	0.4997	0.3042	0.0880	0.0760	**0.0500**	0.0650	0.0522
$\mathrm{CoT}_{\mathrm{avg}}$	0.7241	0.4733	0.6886	0.6041	0.5675	0.4514	0.4531	**0.4306**
$\mathrm{TSR}_{\mathrm{fractal}}$	0.90	0.95	**1.00**	**1.00**	0.80	0.85	**1.00**	**1.00**
$\mathrm{TSR}_{\mathrm{slope}}$	0.85	0.85	**1.00**	**1.00**	0.65	0.70	**1.00**	**1.00**
$\mathrm{TSR}_{\mathrm{discrete}}$	0.80	0.95	**1.00**	0.95	0.70	0.60	0.75	**1.00**
$\mathrm{TSR}_{\mathrm{gaps}}$	0.80	0.80	0.90	**1.00**	0.80	0.80	**1.00**	0.95
$\mathrm{TSR}_{\mathrm{projectile}}$	0.95	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

The best results are in bold

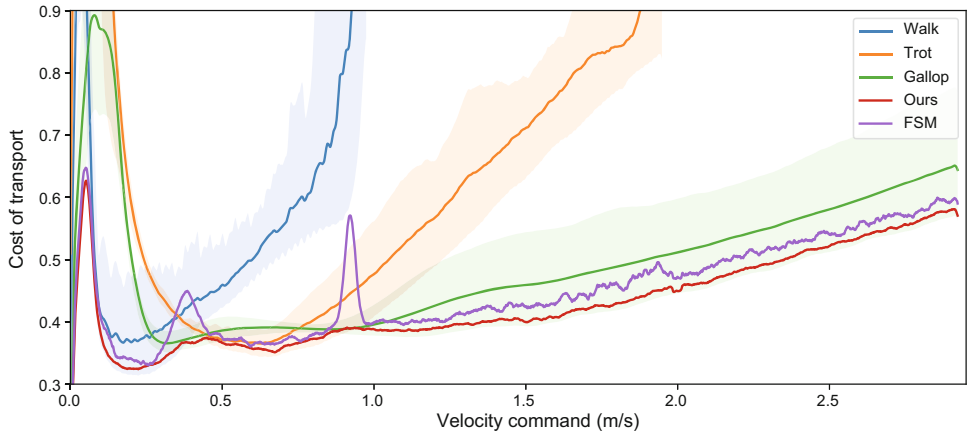

Fig. 5 Comparison of cost of transport between multigait policies and single-gait policies. References to color refer to the online version of this figure

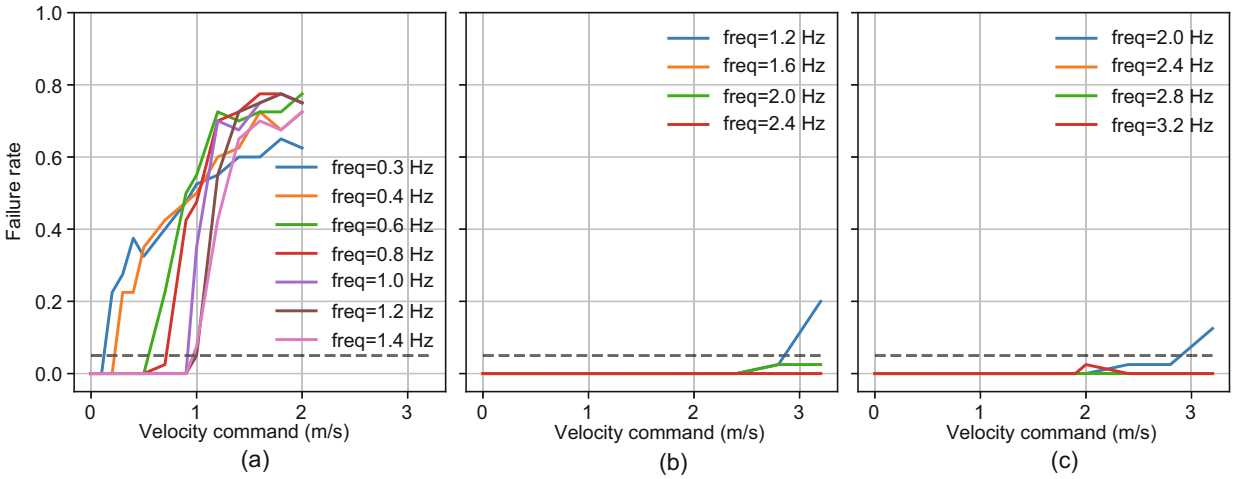

Fig. 6 Failure rates of trained single-gait policies under various velocity commands: (a) walk; (b) trot; (c) gallop. Failure rate is the total number of failures in a time period divided by the length of the episode. To ensure high robustness for real robot transfer, we set the failure threshold at 0.05. References to color refer to the online version of this figure

4.3.2 Gait transition

The performance of the multigait policies is presented in Table 6. Both the proposed method and FSM effectively leverage the advantages of all single-gait policies. The velocity tracking accuracy of both groups is comparable to that of the best-performing single-gait policies, with the proposed method slightly outperforming FSM. Additionally, the proposed policy demonstrates superior adaptability to challenging terrains compared to FSM. Notably, the gait selector tends to favor the robust trot policy when traversing complex terrain conditions, such as discrete regions and slopes.

A detailed plot of energy efficiency is shown in Fig. 5. The proposed policy consistently achieves the highest energy efficiency across all velocity commands. While FSM is also efficient for most velocity commands, two notable spikes in energy consumption occur around gait-switching points. These spikes are attributed to oscillations between policies caused by noisy command velocities, which result in discrete fluctuations in phase vectors. In contrast, the gait selector activates gait transitions at more optimal timing, minimizing such oscillations and ensuring smoother transitions.

Regarding safety, Fig. 9 illustrates the joint velocity ratio distribution. FSM exhibits higher peak joint velocities during gait transitions, indicating

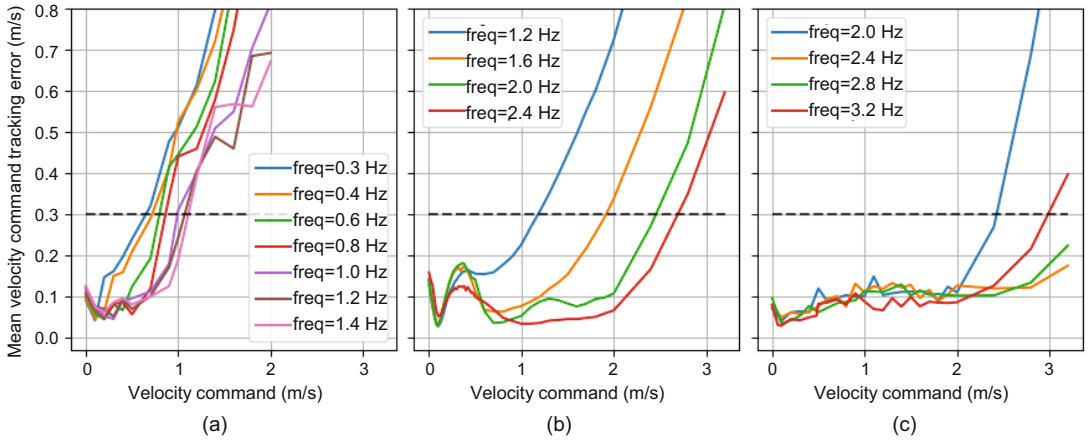

Fig. 7 Velocity tracking performance over a range of velocity commands: (a) walk; (b) trot; (c) gallop. Mean velocity command tracking error is the RMSE between world-frame velocities and commands in one test episode. According to our experience of using a real quadruped robot, we set the threshold to 0.3 m/s. References to color refer to the online version of this figure

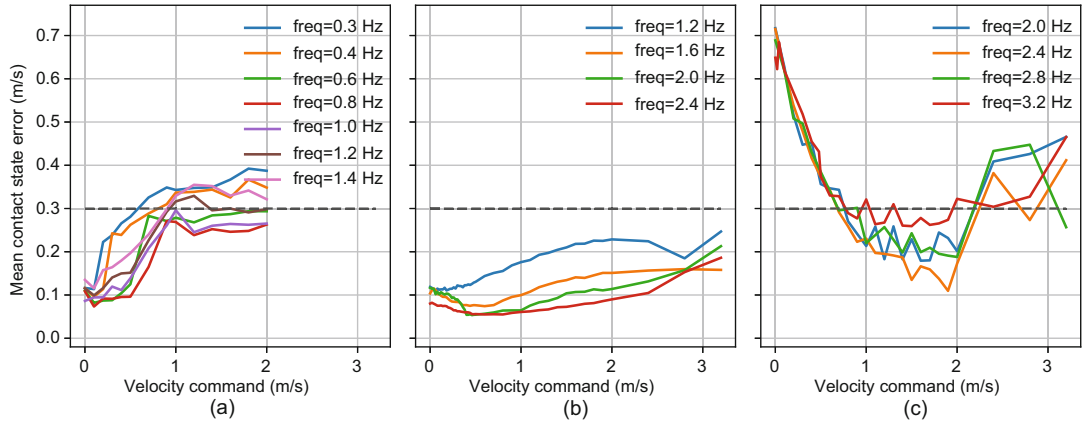

Fig. 8 Mean contact state error over a range of velocity commands: (a) walk; (b) trot; (c) gallop. Mean Contact state error is the average absolute error between binary contact states and expected contact states in one test episode. To make sure that the desired gait is performed while leaving a flexible margin for flight phases, the threshold is set to 0.3 m/s. References to color refer to the online version of this figure

more pronounced fluctuations. In comparison, the gait selector produces smoother transitions with fewer fluctuations, aligning with the energy efficiency findings. A video demonstrating the gait transition behavior is included in the supplementary materials.

4.3.3 Performance summary

A summary of performance under different conditions is provided in Table 6, and a comparison of energy efficiency between our method and two other methods at different velocity commands is presented in Table 7.

Table 7 Comparison of energy efficiency between our method and two other methods at different velocity commands

v^* (m/s)	Cost of transport		
	Ours	EG	Convex MPC
0.375	**0.3551**	0.8059	2.2390
0.9	**0.3803**	0.3936	0.7506
1.5	**0.3937**	0.5841	0.5871

The best results are in bold

5 Conclusions and discussion

This work presents a learning framework for training and organizing robust gait-conditioned policies, revealing the relationship between velocity command and energy efficiency across different gaits.

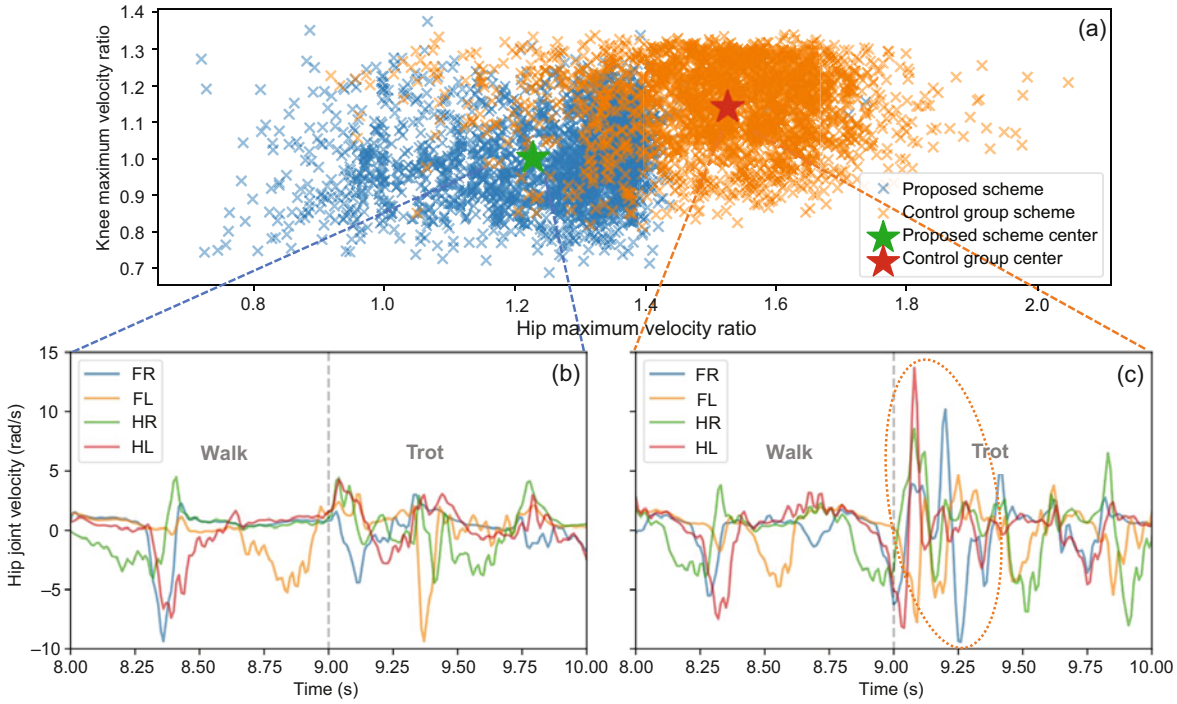

Fig. 9 Joint velocity comparison: (a) joint velocity ratio distribution, where the FSM cluster has more intense hip joint movements; (b–c) typical joint velocity plots during the time of switching, where (b) and (c) are results of the proposed optimal scheme and control group scheme, respectively. The undesired joint velocity spikes are marked by a dotted circle. References to color refer to the online version of this figure

Based on these findings, an adaptive multimodal locomotion controller is developed, and it outperforms single-gait policies in terms of velocity tracking accuracy, CoT, and robustness.

In the presented work, the single-gait policies switch discretely, and the constraints are expressed in rewards when training the gait selector. Future work can focus on developing safer and more sophisticated switching strategies that take into account strict constraints.

The energy consumption analysis reveals that while the proposed policy exhibits advantages over existing policies, it still has lower energy efficiency compared to animals. This can be attributed to the fact that animals have elastic body structures, such as tendons, which allow them to store and release energy (Seok et al., 2013). Future work could incorporate elastic structures into a robot to achieve better energy performance.

The presented framework allows for the use of other motor skills by higher-level controllers. Future research could focus on developing advanced motor skills that integrate perception information to enhance the capability and adaptability of the robot.

Contributors

Funding acquisition was done by Qiuguo ZHU, Jun WU, and Xin ZHAO. Qiuguo ZHU and Zhibin LI were in charge of supervision and project administration. Meng Yee (Michael) CHUAH conceptualized and designed the research. Zhicheng WANG coded the robot learning and controlling system and processed the data. Zhicheng WANG drafted the paper. Meng Yee (Michael) CHUAH helped organize the paper. Zhicheng WANG and Meng Yee (Michael) CHUAH revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Acero F, Yuan K, Li ZB, 2022. Learning perceptual locomotion on uneven terrains using sparse visual observations. *IEEE Robot Autom Lett*, 7(4):8611-8618. https://doi.org/10.1109/LRA.2022.3188108

Agarwal A, Kumar A, Malik J, et al., 2022. Legged locomotion in challenging terrains using egocentric vision. Proc 6th Annual Conf on Robot Learning, p.403-415.

Chen D, Zhou B, Koltun V, et al., 2019. Learning by cheating. Proc 3rd Annual Conf on Robot Learning, p.66-75.

di Carlo J, Wensing PM, Katz B, et al., 2018. Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control. Proc IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.1-9. https://doi.org/10.1109/IROS.2018.8594448

Fu ZP, Kumar A, Malik J, et al., 2021. Minimizing energy consumption leads to the emergence of gaits in legged robots. Proc 5th Annual Conf on Robot Learning, p.928-937.

Haynes GC, Rizzi AA, 2006. Gaits and gait transitions for legged robots. Proc IEEE Int Conf on Robotics and Automation, p.1117-1122. https://doi.org/10.1109/ROBOT.2006.1641859

Hildebrand M, 1965. Symmetrical gaits of horses. *Science*, 150(3697):701-708. https://doi.org/10.1126/science.150.3697.701

Hinton G, Vinyals O, Dean J, 2015. Distilling the knowledge in a neural network. https://arxiv.org/abs/1503.02531

Hoyt DF, Taylor CR, 1981. Gait and the energetics of locomotion in horses. *Nature*, 292(5820):239-240. https://doi.org/10.1038/292239a0

Hsiao-Wecksler ET, Polk JD, Rosengren KS, et al., 2010. A review of new analytic techniques for quantifying symmetry in locomotion. *Symmetry*, 2(2):1135-1155. https://doi.org/10.3390/sym2021135

Hwangbo J, Lee J, Hutter M, 2018. Per-contact iteration method for solving contact dynamics. *IEEE Robot Autom Lett*, 3(2):895-902. https://doi.org/10.1109/LRA.2018.2792536

Ijspeert AJ, 2008. Central pattern generators for locomotion control in animals and robots: a review. *Neur Netw*, 21(4):642-653. https://doi.org/10.1016/j.neunet.2008.03.014

Iscen A, Caluwaerts K, Tan J, et al., 2018. Policies modulating trajectory generators. Proc 2nd Annual Conf on Robot Learning, p.916-926.

Jacobs RA, Jordan MI, Nowlan SJ, et al., 1991. Adaptive mixtures of local experts. *Neur Comput*, 3(1):79-87. https://doi.org/10.1162/neco.1991.3.1.79

Jin YB, Liu XW, Shao YC, et al., 2022. High-speed quadrupedal locomotion by imitation-relaxation reinforcement learning. *Nat Mach Intell*, 4(12):1198-1208. https://doi.org/10.1038/s42256-022-00576-3

Kumar A, Fu ZP, Pathak D, et al., 2021. RMA: rapid motor adaptation for legged robots. Proc 17th Robotics: Science and Systems, p.1-9. https://doi.org/10.15607/rss.2021.xvii.011

Kumar A, Li Z, Zeng J, et al., 2022. Adapting rapid motor adaptation for bipedal robots. Proc IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.1161-1168. https://doi.org/10.1109/IROS47612.2022.9981091

Lee J, Hwangbo J, Hutter M, 2019. Robust recovery controller for a quadrupedal robot using deep reinforcement learning. https://arxiv.org/abs/1901.07517

Lee J, Hwangbo J, Wellhausen L, et al., 2020. Learning quadrupedal locomotion over challenging terrain. *Sci Robot*, 5(47):eabc5986. https://doi.org/10.1126/scirobotics.abc5986

Loquercio A, Kumar A, Malik J, 2023. Learning visual locomotion with cross-modal supervision. Proc Int Conf on Robotics and Automation, p.7295-7302. https://doi.org/10.1109/ICRA48891.2023.10160760

Luo YS, Soeseno JH, Chen TPC, et al., 2020. CARL: controllable agent with reinforcement learning for quadruped locomotion. *ACM Trans Graph*, 39(4):38. https://doi.org/10.1145/3386569.3392433

Makoviychuk V, Wawrzyniak L, Guo YR, et al., 2021. Isaac Gym: high performance GPU based physics simulation for robot learning. Proc 35th Conf on Neural Information Processing Systems, p.1-12.

Margolis GB, Agrawal P, 2022. Walk these ways: tuning robot control for generalization with multiplicity of behavior. Proc 6th Annual Conf on Robot Learning, p.1-9.

Margolis GB, Yang G, Paigwar K, et al., 2022. Rapid locomotion via reinforcement learning. Proc 18th Robotics: Science and Systems, p.1-9.

Miki T, Lee J, Hwangbo J, et al., 2022. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci Robot*, 7(62):eabk2822. https://doi.org/10.1126/scirobotics.abk2822

Nahrendra IMA, Yu B, Myung H, 2023. DreamWaq: learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. Proc IEEE Int Conf on Robotics and Automation, p.5078-5084. https://doi.org/10.1109/ICRA48891.2023.10161144

Paszke A, Gross S, Massa F, et al., 2019. PyTorch: an imperative style, high-performance deep learning library. Proc 33rd Int Conf on Neural Information Processing Systems, p.8024-8035.

Peng XB, Chang M, Zhang G, et al., 2019. MCP: learning composable hierarchical control with multiplicative compositional policies. Proc 33rd Int Conf on Neural Information Processing Systems, Article 331.

Peng XB, Coumans E, Zhang TN, et al., 2020. Learning agile robotic locomotion skills by imitating animals. Proc 16th Robotics: Science and Systems, p.1-9.

Peng XB, Ma Z, Abbeel P, et al., 2021. AMP: adversarial motion priors for stylized physics-based character control. *ACM Trans Graph*, 40(4):144. https://doi.org/10.1145/3450626.3459670

Peng XB, Guo YR, Halper L, et al., 2022. ASE: large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Trans Graph*, 41(4):94. https://doi.org/10.1145/3528223.3530110

Schulman J, Wolski F, Dhariwal P, et al., 2017. Proximal policy optimization algorithms. https://arxiv.org/abs/1707.06347

Seok S, Wang A, Chuah MY, et al., 2013. Design principles for highly efficient quadrupeds and implementation on the MIT Cheetah robot. IEEE Int Conf on Robotics and Automation, p.3307-3312. https://doi.org/10.1109/ICRA.2013.6631038

Shao YC, Jin YB, Liu XW, et al., 2022. Learning free gait transition for quadruped robots via phase-guided controller. *IEEE Robot Autom Lett*, 7(2):1230-1237. https://doi.org/10.1109/LRA.2021.3136645

Siekmann J, Valluri S, Dao J, et al., 2020. Learning memory-based control for human-scale bipedal locomotion. Proc 16th Robotics: Science and Systems, p.1-8. https://doi.org/10.15607/rss.2020.xvi.031

Siekmann J, Green K, Warila J, et al., 2021a. Blind bipedal stair traversal via sim-to-real reinforcement learning. Proc 17th Robotics: Science and Systems, p.1-9. https://doi.org/10.15607/rss.2021.xvii.061

Siekmann J, Godse Y, Fern A, et al., 2021b. Sim-to-real learning of all common bipedal gaits via periodic reward composition. Proc IEEE Int Conf on Robotics and Automation, p.7309-7315. https://doi.org/10.1109/ICRA48506.2021.9561814

Tan DCH, Zhang J, Chuah M, et al., 2023. Perceptive locomotion with controllable pace and natural gait transitions over uneven terrains. https://arxiv.org/abs/2301.10894

Tan J, Zhang TN, Coumans E, et al., 2018. Sim-to-real: learning agile locomotion for quadruped robots. Proc 14th Robotics: Science and Systems, p.1-9. https://doi.org/10.15607/RSS.2018.XIV.010

Tsounis V, Alge M, Lee J, et al., 2020. DeepGait: planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robot Autom Lett*, 5(2):3699-3706. https://doi.org/10.1109/LRA.2020.2979660

Xi WT, Yesilevskiy Y, Remy CD, 2016. Selecting gaits for economical locomotion of legged robots. *Int J Robot Res*, 35(9):1140-1154. https://doi.org/10.1177/0278364915612572

Xie ZM, Da XY, van de Panne M, et al., 2021. Dynamics randomization revisited: a case study for quadrupedal locomotion. Proc IEEE Int Conf on Robotics and Automation, p.4955-4961. https://doi.org/10.1109/ICRA48506.2021.9560837

Yang CY, Yuan K, Zhu QG, et al., 2020. Multi-expert learning of adaptive legged locomotion. *Sci Robot*, 5(49):eabb2174. https://doi.org/10.1126/scirobotics.abb2174

Zhang H, Starke S, Komura T, et al., 2018. Mode-adaptive neural networks for quadruped motion control. *ACM Trans Graph*, 37(4):145. https://doi.org/10.1145/3197517.3201366

List of supplementary materials

Quadruped gait transition demonstration video