

OPEN ACCESS

EDITED BY

Christofer J. Clemente,
University of the Sunshine Coast, Australia

REVIEWED BY

Ranjan Dasgupta,
Tata Consultancy Service Ltd., India
Di Wang,
Foxconn Assembly LLC, United States

*CORRESPONDENCE

Wanming Yu,
✉ wanming@robots.ox.ac.uk
Dimitrios Kanoulas,
✉ d.kanoulas@ucl.ac.uk

RECEIVED 01 September 2025

REVISED 15 December 2025

ACCEPTED 05 January 2026

PUBLISHED 18 February 2026

CITATION

Yu W, Acero F, Atanassov V, Yang C, Havoutis I,
Kanoulas D and Li Z (2026) Discovery of
skill-switching criteria for learning agile
quadruped locomotion.
Front. Robot. AI 13:1697159.
doi: 10.3389/frobt.2026.1697159

COPYRIGHT

© 2026 Yu, Acero, Atanassov, Yang, Havoutis,
Kanoulas and Li. This is an open-access article
distributed under the terms of the [Creative
Commons Attribution License \(CC BY\)](#). The
use, distribution or reproduction in other
forums is permitted, provided the original
author(s) and the copyright owner(s) are
credited and that the original publication in
this journal is cited, in accordance with
accepted academic practice. No use,
distribution or reproduction is permitted
which does not comply with these terms.

Discovery of skill-switching criteria for learning agile quadruped locomotion

Wanming Yu^{1*}, Fernando Acero², Vassil Atanassov¹,
Chuanyu Yang³, Ioannis Havoutis¹, Dimitrios Kanoulas^{2*} and
Zhibin Li²

¹Department of Engineering Science, Oxford Robotics Institute, University of Oxford, Oxford, United Kingdom, ²Department of Computer Science, University College London, London, United Kingdom, ³National Elite Institute of Engineering, Chongqing University, Chongqing, China

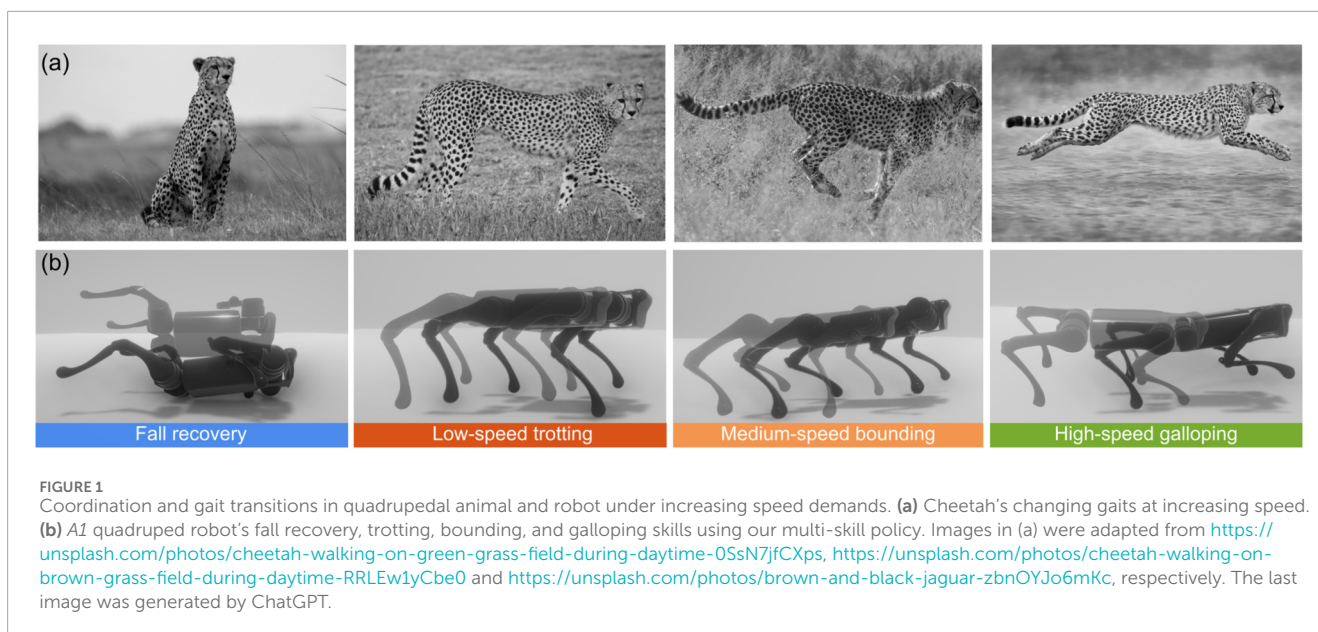
This study develops a hierarchical learning and optimization framework that can learn and achieve well-coordinated multi-skill locomotion. The learned multi-skill policy can switch between skills automatically and naturally while tracking arbitrarily positioned goals and can recover from failures promptly. The proposed framework is composed of a deep reinforcement learning process and an optimization process. First, the contact pattern is incorporated into the reward terms to learn different types of gaits as separate policies without the need for any other references. Then, a higher-level policy is learned to generate weights for individual policies to compose multi-skill locomotion in a goal-tracking task setting. Skills are automatically and naturally switched according to the distance to the goal. The appropriate distances for skill switching are incorporated into the reward calculation for learning the high-level policy and are updated by an outer optimization loop as learning progresses. We first demonstrate successful multi-skill locomotion in comprehensive tasks on a simulated Unitree A1 quadruped robot. We also deploy the learned policy in the real world, showcasing trotting, bounding, galloping, and their natural transitions as the goal position changes. Moreover, the learned policy can react to unexpected failures at any time, perform prompt recovery, and successfully resume locomotion. Compared to baselines, our proposed approach achieves all the learned agile skills with improved learning performance, enabling smoother and more continuous skill transitions.

KEYWORDS

deep reinforcement learning, gait transitions, hierarchical learning and optimization, legged locomotion, multi-skill locomotion, robot learning, skill switching

1 Introduction

Animals have evolved highly efficient movement strategies. Mimicking these can improve legged locomotion in terms of agility, stability, and adaptivity (Figure 1). In particular, animals learn to switch between motor skills swiftly according to tasks and surroundings. For instance, horses switch to different gait patterns as the speed changes (Hoyt and Taylor, 1981). However, reproducing multiple gaits and their dynamically feasible transitions on legged robots remains challenging in the robot learning and control community. In addition, the ability to recover from various failures, which is of vital interest for successful and resilient real-world deployment, is not yet well-studied in



multi-skill locomotion. A multi-skill framework has the ability to recover from failures during locomotion (Yang et al., 2020; Yuan and Li, 2022); however, it does not show more dynamic gaits beyond trotting. Although improving robustness against failures or fall recovery has been studied in several previous works (Hwangbo et al., 2019; Castano et al., 2019; Cordie et al., 2024), it is learned as a single skill and cannot be combined with other skills.

1.1 Reproducing gait patterns

Behavior cloning or imitation learning approaches have been applied to reproduce various gaits from reference motions. Reference motions can be captured from animal locomotion, which may be limited in variety or generated using model predictive controllers (Reske et al., 2021), which require domain knowledge and computational efficiency. In general, such approaches can be difficult to scale beyond the dataset, thereby lacking robustness and diversity of the learned behaviors. There have been attempts to model or learn parameterized control policies to achieve different styles of walking. A phase-guided controller is used to learn gait transitions between walking, trotting, pacing, and bounding on a quadrupedal robot (Shao et al., 2021). A single policy is learned to control various gaits with variable footswing, posture, and speed (Margolis and Agrawal, 2023). Simple balance control has been inspired by a bicycle to achieve high-speed running on a quadruped robot (Hattori et al., 2025). However, such approaches require handcrafted, control-specific behavior parameters, which demand domain knowledge and are not intuitive for gait switching.

1.2 Generative models in multi-skill locomotion

Recent advances in generative models have achieved low-level control of locomotion gaits on quadrupedal robots. Variational

autoencoders have been applied to learn a disentangled and two-dimensional latent representation across locomotion gaits with respect to footswing heights and lengths (Mitchell et al., 2024). Given the desired gait type and swing characteristics, it achieved low-level control of trotting, crawling, and pacing on quadrupedal robots. In addition, diffusion models have demonstrated the capability of achieving multi-skill locomotion control with a single policy, including trotting, hopping, pacing, walking, and running (Huang et al., 2024), along with walking, crawling, and their transitions (O'Mahoney et al., 2025). However, these generative models require expert demonstrations, and the performance is dependent on the quality of the dataset. Moreover, the gait type was conditioned on a certain input to the control policy. In contrast, gait types were autonomously discovered by our proposed hierarchical framework, which covers both high-level and low-level multi-skill locomotion control.

1.3 Foundation models in legged locomotion

Applying foundation models in robot learning applications is a favorable approach for achieving generalized robot tasks and behaviors. Pre-trained vision-language models (VLMs) usually focus on high-level reasoning and planning to select from a set of existing low-level skills (Chen et al., 2025). However, certain low-level locomotion skills can be difficult to obtain in practice. Several attempts have also been made to apply pre-trained large language models (LLMs) to achieve multiple gaits via low-level interfaces, such as foot contact patterns (Tang et al., 2023). In general, these foundation models in robotic applications require either careful prompt engineering or a huge amount of robotic data for fine-tuning. In practice, robotic data can be difficult to obtain in certain cases, and fine-tuning of large-scale models may require substantial computational resources.

1.4 Bio-inspired multi-gait locomotion

Unlike the above approaches requiring reference motions, some robotics research has applied deep reinforcement learning to acquire animal-like gait transitions based on various criteria inspired by biological principles, where reference motions are not necessary. By minimizing energy consumption (Miranda et al., 2025), the robot can achieve gait transitions between walking, trotting, and fly-trotting at different speed ranges using a single policy (Liang et al., 2025) or a hierarchical structure (Yang et al., 2022), along with gait transitions from walking to trotting to bouncing (Fu et al., 2021). Another bio-inspired research modulates gait transitions according to Froude numbers (Humphreys et al., 2023). A more recent work learned gait transitions from walking to trotting on flat ground and trotting to pronking when crossing gaps according to viability (Shafiee et al., 2024). However, galloping cannot emerge or be incorporated at high speed on mechanical robots in the above frameworks. Moreover, a series of works utilizes central pattern generators (CPGs) to produce different gaits by deep reinforcement learning on quadrupedal robots. The most recent work adopted a coupling-driven approach to learn a policy via deep reinforcement learning to modulate the parameters of CPGs, producing nine gaits and transitions, including galloping, based on the cost of transport (CoT) (Bellegarda et al., 2025). In contrast with bio-inspired multi-gait locomotion, our proposed framework can produce a galloping gait. Moreover, our approach is not constrained by biological principles as we can define customized cost terms for optimizing gait switch timing. In addition to biology-inspired criteria, we can also include other cost terms, such as task-related costs.

1.5 State-of-the-art quadrupedal locomotion

A parallel line of research demonstrated impressive dynamic parkour skills in legged robots (Caluwaerts et al., 2023; Cheng et al., 2024; Zhuang et al., 2023; He et al., 2024). However, these works usually focused on navigating the robot along a series of challenging terrains and obstacles. In most cases, a simple goal-reaching task is considered in these navigation tasks, while gait patterns are not taken into account. In contrast, this study focuses on multi-skill navigation and control tasks, i.e., reaching arbitrary goals with various gait patterns and their transitions.

1.6 Contributions

To summarize, the advantages of our proposed approach over the existing literature include (1) we do not require reference trajectories or expert demonstrations, and our model can learn multi-skill locomotion purely from scratch; (2) animal-like galloping gait can be activated at high-speed locomotion; (3) autonomous fall recovery is incorporated in the multi-skill policy, enabling high robustness and requiring less human intervention; and (4) flexible gait-switch criteria are automatically discovered for mechanical robots. To the best of our knowledge, our work is the first multi-skill learning and optimization framework that is compatible

with incorporating and synthesizing multiple highly dynamic locomotion gaits (especially galloping) and producing natural, dynamically feasible transitions by automatically discovered gait-switch criteria. Our work demonstrates four skills on a quadruped robot in the real world, including prompt fall recovery at any stage during multi-skill locomotion. To summarize, the contributions of this study include the following:

- Incorporating highly dynamic locomotion skills of bounding and galloping in addition to trotting into learning one coherent multi-skill policy, without the need for reference trajectories.
- Demonstrating successful trotting, bounding, and galloping and their dynamically feasible and continuous transitions with one synthesized multi-skill policy on a real quadruped robot.
- Successful failure recovery at any stage of different gaits.
- Automatic discovery of gait-switch criteria as motor learning progresses, which converges to a higher training reward faster than baselines.

Our hierarchical multi-skill learning and optimization framework is shown in Figure 2, which includes (1) a set of pre-trained reusable single-skill neural network policies, each representing a single locomotion skill; (2) a task-level neural network that generates weights for each skill to produce our multiplicative composite policy; (3) the composite multi-skill policy; and (4) the outer optimization loop for the discovery of gait-switch criteria represented by the relative goal distances in the horizontal plane that activate the switch from trotting to bounding and the switch from bounding to galloping, respectively.

In the following sections, we first review the details of our hierarchical learning and optimization framework in Sections 2.1 and 2.2. Then, we demonstrate and analyze the learned multi-skill locomotion policy on a real quadruped robot in Section 3. Finally, we conclude our work in Section 4.

2 Materials and methods

2.1 Hierarchical multi-skill learning framework

2.1.1 Learning individual skills

The robot learns five individual skills separately using a systematic deep reinforcement learning framework, namely, fall recovery, trotting, pacing, bounding, and galloping. Each locomotion skill is a feedback control policy represented by a neural network, which is learned using the Soft Actor–Critic (SAC) algorithm (Haarnoja et al., 2018). Details of the key components for our deep reinforcement learning framework are given below. Each individual skill is a fully connected neural network with two hidden layers. Each hidden layer has 256 neurons and uses a ReLU activation function, and the output layer uses a tanh activation function. The output of each neural network is 24-dimensional, including the mean and variance of all 12 joints. For skill i , μ_i^j and σ_i^j represent the mean and variance of the j th joint, respectively, and the final desired position of the j th joint, \vec{q}_p^j , is sampled from the corresponding Gaussian distribution $\mathcal{N}(\mu_p^j, \sigma_p^j)$.

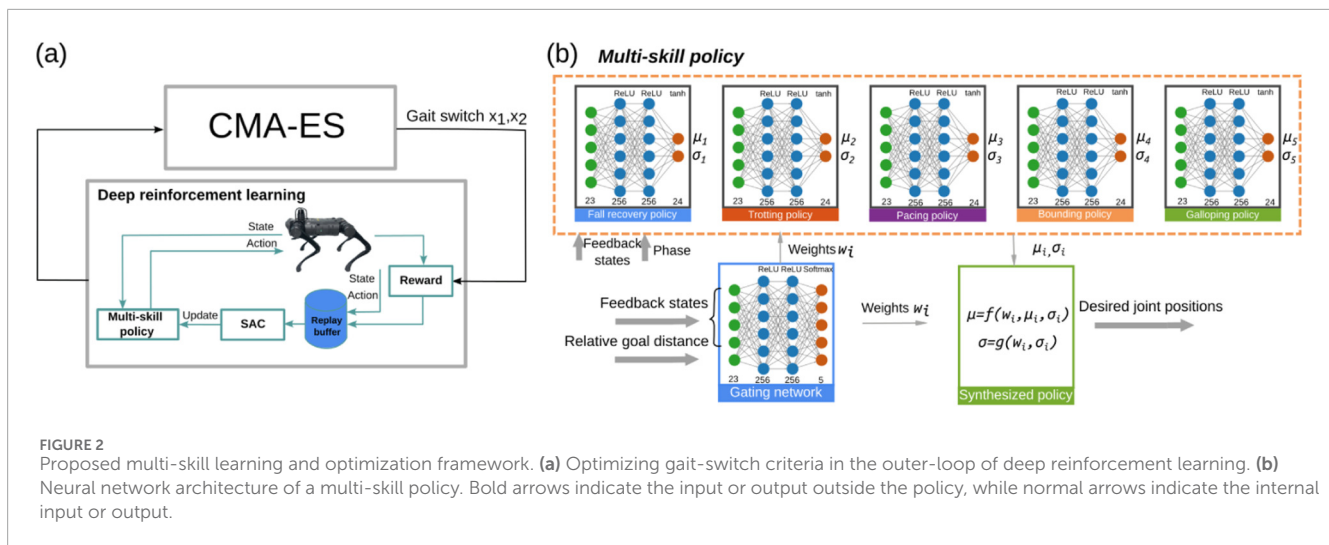


FIGURE 2 Proposed multi-skill learning and optimization framework. (a) Optimizing gait-switch criteria in the outer-loop of deep reinforcement learning. (b) Neural network architecture of a multi-skill policy. Bold arrows indicate the input or output outside the policy, while normal arrows indicate the internal input or output.

2.1.1.1 State observation and action space

Following the key feedback states in learning locomotion skills (Yu et al., 2023), the state input to the actor neural network includes (1) normalized gravity vector in the robot local frame, which reflects the body orientation of the robot, (2) base angular velocity, (3) base linear velocity in the robot heading frame, and (4) joint positions. For learning periodic locomotion skills, we also included a two-dimensional phase vector ($\sin 2\pi\phi, \cos 2\pi\phi$) to represent continuous temporal information that encodes phase ϕ from 0% to 100% of a periodic motion. The actions are the desired joint positions for 12 joints, including hip roll, hip pitch, and the knee joints of four legs.

2.1.1.2 Reward design

Trotting and bounding were learned with a fixed desired velocity, while galloping was learned by maximizing velocity. The reward function for learning individual policies is composed of continuous and discrete reward terms. For continuous reward terms, we use a radial basis function (RBF) to formulate as in Equation 1:

$$\varphi(x, \hat{x}, \alpha) = \exp(-\alpha(\hat{x} - x)^2), \tag{1}$$

where x is the continuous physical quantity, \hat{x} is the corresponding reference, and α is the shape parameter that controls the width of RBF. The formulation and weight of each reward term are provided in Tables 1 and 2, respectively. There are 11 reward terms in total for training individual skills. The essential reward terms that distinguish different skills are the base linear velocity rewards (different desired velocity ranges for different gaits) and reference foot contact rewards (different contact patterns for different gaits; see Section 2.1.1.3). The remaining reward terms are commonly used in the legged locomotion field to maintain locomotion stability, such as preserving a certain orientation and height, minimizing energy consumption by reducing joint torques and velocities, and preventing falls by penalizing unintended body contacts with the ground while encouraging proper foot contacts.

2.1.1.3 Reference foot-contact reward

The last reward term in Table 1, i.e., the reference foot-contact reward, is the key to learning different gait types without

reference. The desired foot-contact pattern for each gait type is inspired by quadrupedal animals (Owaki and Ishiguro, 2017), as shown in Figure 3. In this study, we assume trotting, bounding, and galloping gaits as speed increases, as a proof-of-concept. It should be noted that the order of gait types is not fixed. Here, we determine the gait type at different stages according to its characteristics.

2.1.2 Learning multi-skill locomotion

The task objective of the robot is to track an arbitrary goal in the horizontal plane with natural gait transitions from trotting to bounding to galloping. The goal is represented by (d_g, θ_g) , and the goal position in the horizontal plane is $(d_g \cos \theta_g, d_g \sin \theta_g)$. At each episode, we set a fixed goal, where $d_g \sim (0m, 15m)$ and $\theta_g \sim (-180^\circ, 180^\circ)$. The multi-skill locomotion policy is synthesized on-the-fly by the multiplicative composition (Peng et al., 2019) of the low-level pre-trained individual policies according to the output from the high-level gating network. We introduce more details about each component separately.

2.1.2.1 Gating network

Our high-level gating network is a fully connected neural network with two hidden layers. Each hidden layer has 256 neurons and uses a ReLU activation function, while the output layer uses a Softmax function. The gating network receives the following as input: gravity vector, base angular velocity, base linear velocity, joint positions, and the normalized distance between the robot and the goal in the horizontal plane. The gating network outputs the weights for each locomotion skill, which add up to one.

2.1.2.2 Composite multi-skill policy

During the training of multi-skill locomotion, the parameters of the expert networks are transferred from the pre-trained single-skill policies and remain fixed throughout. That is, only the gating network parameters are updated by backpropagation of the gradient obtained from the designed reward. The weights for each locomotion skill generated by the gating network are then applied to synthesize the multi-skill Gaussian policy $\pi(a|s, g)$ by the multiplicative composition of n pre-trained single skills as in Equation 2:

TABLE 1 Reward terms for learning quadruped locomotion skills.

| Physical quantity | Reward term |
|--------------------------|--|
| Base orientation | $w_\phi \times \varphi(\phi, [0, 0, -1], -2.35)$ |
| Base height | $w_h \times \varphi(h, \hat{h}, -51.16)$ |
| Base linear velocity | $w_v \times \begin{cases} v_{x_{base}}^{world2}, & \text{gallop} \\ \varphi(v_{base}^{world}, \hat{v}_{base}^{world}, -18.42), & \text{else} \end{cases}$ |
| Joint torque | $w_\tau \times \varphi(\tau, 0, -0.004)$ |
| Joint velocity | $w_{\dot{q}} \times \varphi(\dot{q}, 0, -0.032)$ |
| Body-ground contact | $w_{bg} \times \begin{cases} 0, & \text{base in contact with ground} \\ 1, & \text{base not in contact with ground} \end{cases}$ |
| Foot-ground contact | $w_{fg} \times \begin{cases} 0, & \text{no foot in contact with ground} \\ 1, & \text{foot in contact with ground} \end{cases}$ |
| Symmetric foot placement | $w_{pf} \times \begin{cases} \varphi(p_{foot}^{base}, \hat{p}_{foot}^{base}, -51.16), & \text{recovery} \\ \varphi(1/4 \sum_{n=1}^4 (p_{foot,n}^{world}, p_{base}^{world}), -51.16), & \text{gaits} \end{cases}$ |
| Swing and stance | $w_{hf} \times \varphi(h_{foot}^{world}, v_{foot}^{world}, \hat{h}_{foot}^{world}, v_{foot}^{world}, -460.50)$ |
| Yaw velocity | $w_\psi \times \varphi(\dot{\psi}, 0, -7.47)$ |
| Reference foot contact | $w_f \times \begin{cases} 0, & \text{not match desired foot contact} \\ 1, & \text{match desired foot contact} \end{cases}$ |

TABLE 2 Reward term weights for learning single locomotion skills.

| Task | W_ϕ | W_h | W_v | W_τ | $W_{\dot{q}}$ | W_{bg} | W_{fg} | W_{pf} | W_{hf} | W_ψ | W_f |
|---------------|----------|-------|-------|----------|---------------|----------|----------|----------|----------|----------|-------|
| Fall recovery | 0.189 | 0.189 | 0.114 | 0.076 | 0.076 | 0.083 | 0.083 | 0.189 | 0.000 | 0.000 | 0.000 |
| Gait | 0.068 | 0.068 | 0.170 | 0.017 | 0.017 | 0.048 | 0.000 | 0.034 | 0.034 | 0.068 | 0.476 |

$$\pi(a|s, g) = \frac{1}{Z(s, g)} \prod_{i=1}^n \pi_i(a|s) w_i(s, g), \quad w_i(s, g) \geq 0, \quad (2)$$

where $\pi_i(a|s)$ is the i th single-skill neural network policy, $w_i(s, g)$ is the weight for the corresponding skill to influence the composite policy, and $Z(s, g)$ is the normalization factor. The synthesized policy is a multiplicative composition of n Gaussian policies, i.e., n single skills. As discussed by Peng et al. (2019), the multiplicative composition of Gaussian primitives results in another Gaussian policy, i.e., the composite policy. Due to the use of Gaussian primitives, the composite mean μ^j and composite variance σ^j of the j th joint of the synthesized Gaussian policy are obtained as follows:

$$\mu^j(s, g) = \frac{1}{\sum_{i=1}^n \frac{w_i(s, g)}{\sigma_i^j(s, g)}} \sum_{i=1}^n \frac{w_i(s, g)}{\sigma_i^j(s, g)} \mu_i^j(s, g), \quad (3)$$

$$\sigma^j(s, g) = \left(\sum_{i=1}^n \frac{w_i(s, g)}{\sigma_i^j(s, g)} \right)^{-1}. \quad (4)$$

Here, μ_i^j and σ_i^j represent the mean and variance of the i th skill network of the j th joint, respectively. The final desired position of the j th joint \hat{q}^j is sampled from the composite

Gaussian distribution $\mathcal{N}(\mu^j, \sigma^j)$. The derivation of Equation 3 and Equation 4 can be found in Supplementary Material. It should be noted that we adopted a multiplicative model instead of an additive model in a hierarchical learning framework, such as a mixture of experts (MoE) (Jacobs et al., 1991), to avoid conflicting behaviors or blending artifacts caused by the sum of primitives, as reported by Peng et al. (2019).

2.1.2.3 Control framework

The pre-trained single-skill policies, the gating network, and the composite multi-skill policy run at 25 Hz together generate desired joint positions that are tracked by joint-level PD controllers at 1,000 Hz. The PD controllers receive the desired joint positions \hat{q} , measured joint positions q , and joint velocities \dot{q} as input and output the joint torque commands $\tau = K_p(\hat{q} - q) + K_d(0 - \dot{q})$.

2.1.2.4 Reward design

For multi-skill learning, we design three groups of reward terms. The most important group of reward terms is related to goal tracking r_g , another group is the reference foot-contact reward r_f , and the last group includes the remaining reward terms in learning single

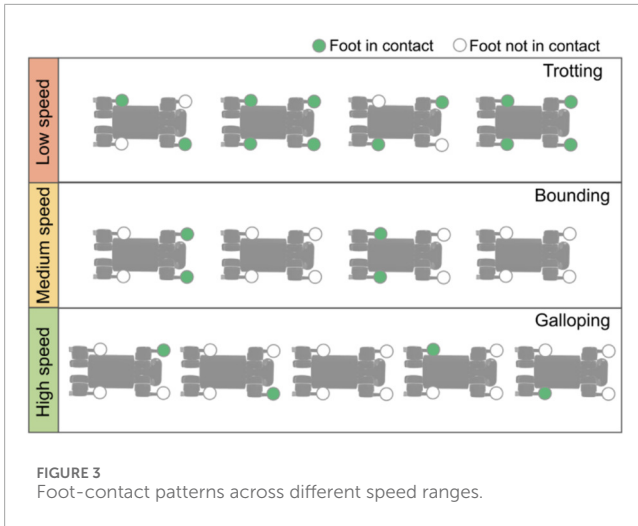


FIGURE 3 Foot-contact patterns across different speed ranges.

locomotion skills r_e . We set the overall reward r as the weighted sum of these terms as in Equation 5:

$$r = 0.6r_g + 0.2r_f + 0.2r_e. \tag{5}$$

2.1.2.4.1 Goal-tracking reward. Our goal-tracking reward consists of three terms. First, the relative position reward r_{p_g} , which encourages minimization of the relative distance between the robot and the goal in the horizontal plane $d \geq 0$ as in Equation 6:

$$r_{p_g} = \varphi \left(P_{goal}^{world}, P_{base}^{world}, -0.74 \right). \tag{6}$$

Second, the robot velocity reward r_{v_g} is used together with the two other reward terms, encouraging the robot to track the goal as quickly as possible, although the relative position to the goal is dominant, as reflected in the reward weights given in Equation 7. It should be noted that the other reward terms, apart from the goal-tracking reward, also constrain the learned behaviors to be reasonable and feasible.

$$r_{v_g} = v^2. \tag{7}$$

The third term is the robot heading reward r_{ϕ_g} , which encourages alignment of the robot heading toward the goal as in Equation 8:

$$r_{\phi_g} = \varphi \left(u_{goal,base}^{base}, [1, 0, 0], -2.35 \right), \tag{8}$$

where $u_{goal,base}^{base}$ is the unit vector pointing from the robot base to the goal in the base frame. Our full goal-tracking reward is $r_g = r_{h_z} r_{\phi} (8r_{p_g} + 4r_{v_g} + 4r_{\phi_g})$, which increases the goal-tracking reward weight when the robot is closer to the nominal standing pose. r_{h_z} and r_{ϕ} are the base height reward and base orientation reward, respectively, as shown in Table 1. The agent will prioritize these two rewards to ensure that the robot maintains the desired height and orientation in the initial stages and then progresses to maximize the goal-tracking reward.

2.1.2.4.2 Reference foot-contact reward. Regarding the reference foot-contact reward in Table 1, learning different gaits requires

different reference contact patterns. For multi-skill training, we activate different gaits according to the relative distance d between the goal and the robot, as described in the goal-tracking reward. Specifically, we use the trotting contact pattern as the reference to calculate this reward if $|d| < x_1$, the bounding contact pattern if $x_1 \leq |d| < x_2$, and the gallop contact pattern if $|d| \geq x_2$, where x_1 and x_2 are the gait switch criteria, which are discussed in Section 2.2 via an optimization loop outside the motor learning progress. In addition, it should be noted that for single skill policies, trotting has a similar velocity range as pacing and is a more common and stable gait. Therefore, when training our multi-skill policy, trotting was activated by the related reward terms when the goal was close rather than pacing. Nevertheless, technically, pacing can also be included for training a new multi-skill policy if needed.

2.2 Discovery of the skill-switching criteria

In this section, we propose setting up an optimization problem in the outer loop of the motor learning process to automatically discover the gait-switch criteria from trotting to bounding and from bounding to galloping. We use covariance matrix adaptation evolution strategy (CMA-ES) (Hansen, 2016), which is a derivative-free evolution strategy inspired by biological evolution for optimization. Here, we use the relative distance between the robot and the goal as the gait-switch criterion, as a proof of concept. We aim to find the gait-switch criteria x_1, x_2 to maximize the sum of goal-tracking reward r_g over each episode. The optimization problem is formulated as follows:

$$\begin{aligned} & \arg \min_{x_1, x_2} \sum_{i=1}^N -r_g(x_1, x_2) \\ & \text{s.t.} \quad 0 \leq x_1 \leq 15, \\ & \quad \quad 0 \leq x_2 \leq 15, \\ & \quad \quad x_1 < x_2. \end{aligned}$$

Here, x_1, x_2 are the decision variables representing the relative distance between the robot and the goal in the horizontal plane to activate switching between trotting and bounding and between bounding and galloping gaits, respectively. N is the number of time steps in an episode.

It should be noted that our framework is not restricted to using relative distance as the gait-switch criterion. To demonstrate the effectiveness and generalizability of our proposed framework, we also use velocity as the gait-switch criterion in a velocity-tracking locomotion task, which can enrich our framework with more locomotion tasks and scenarios. Section 3.5.2 contains additional details.

3 Results

This section first introduces the experimental setup and then presents the optimization results for the gait-switch criteria. We then demonstrate that the proposed multi-skill policy achieves versatile locomotion gaits and their continuous transitions. Moreover, we showcase robust multi-skill locomotion in various test scenarios. Furthermore, our proposed framework illustrates its generalizability

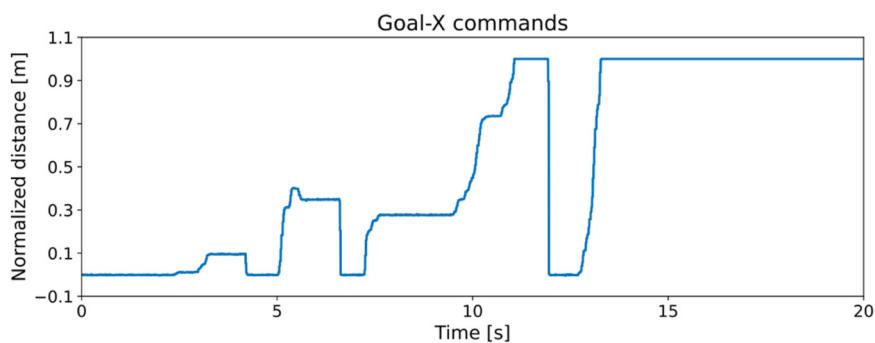


FIGURE 4 Normalized relative goal command in the robot heading frame is provided to encourage fall recovery, trotting, bounding, galloping, and their transitions.

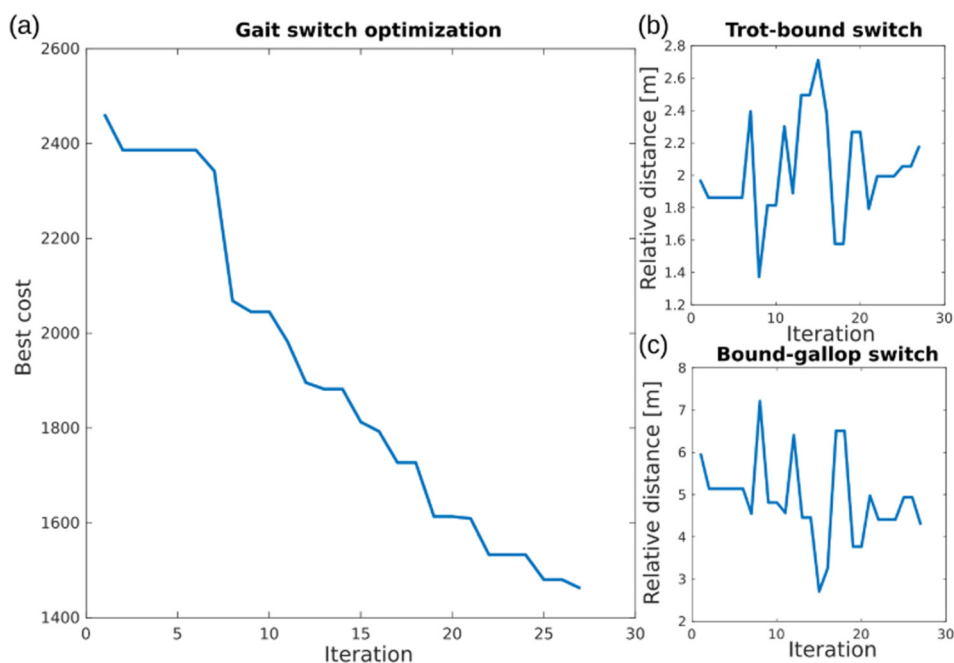


FIGURE 5 Results of CMA-ES optimization for gait-switch criteria in learning multi-skill locomotion. **(a)** Best cost during CMA-ES optimization. **(b)** Optimized relative distance for switching from trotting to bounding. **(c)** Optimized relative distance for switching from bounding to galloping.

by acquiring multi-skill locomotion with two different formulations of gait-switch criteria: distance and velocity. Comprehensive ablation studies validate that our approach outperforms the baseline, with improved learning performance and more continuous gait transitions.

3.1 Experimental setup

3.1.1 Multi-skill training setup

We sample 5,000 steps from the composite multi-skill policy for each training epoch, i.e., 20 episodes without early termination. Each episode lasts 10 s; the batch size is 128; the replay buffer size is 1e6; the learning rate is 3e-4; weight decay is 1e-6; the soft target update

is 0.001; and the discount factor is 0.995 for fall recovery and 0.955 for locomotion gaits.

3.1.2 Skill-switch criterion optimization setup

We set the initial gait-switch criteria as $x_1 = 2.0\text{ m}$ and $x_2 = 5.0\text{ m}$ to warmstart the optimization; the population size is 50, and $\sigma = 1.0\text{ m}$. The CMA-ES optimization runs for an iteration for every 20 iterations of inner-loop deep reinforcement learning.

3.1.3 Goal trajectory setup

We provide normalized relative goal distance in the x- and y-axes in the robot heading frame via a joystick in real-world tests to encourage the emergence of multiple dynamic skills and their transitions. An example goal trajectory is shown in [Figure 4](#).

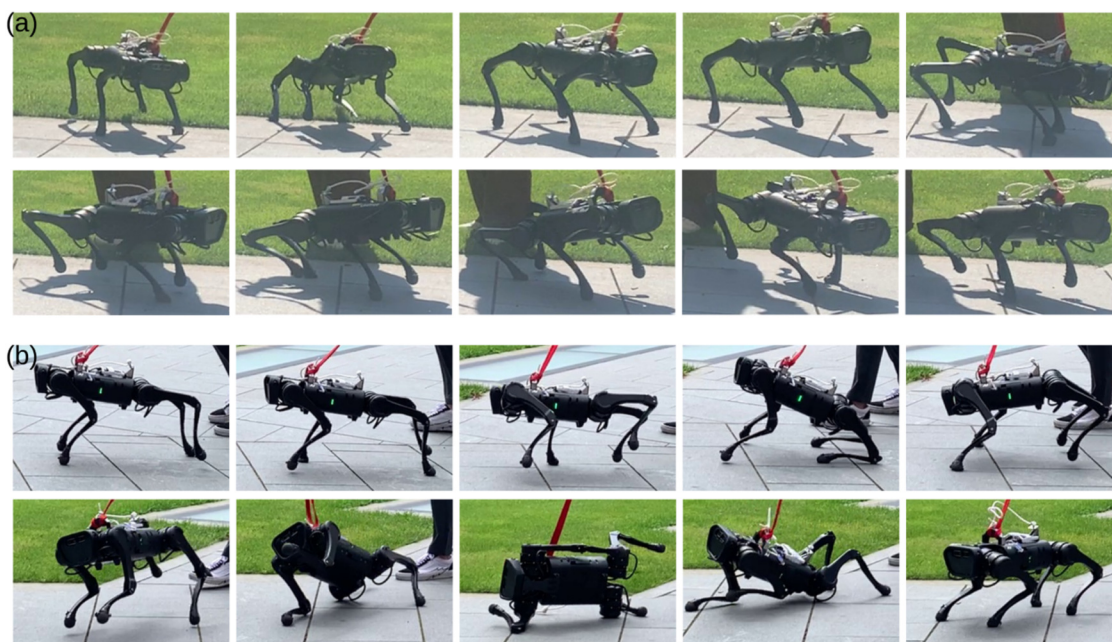
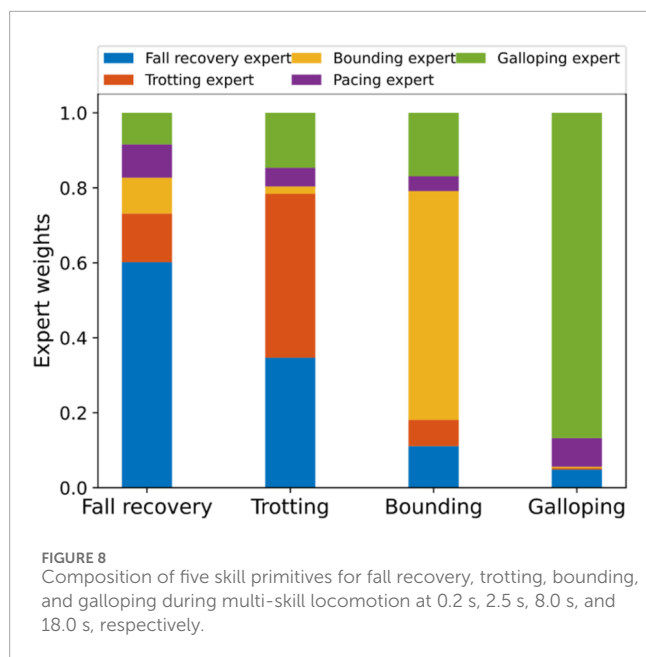


FIGURE 6 Comparison of multi-skill locomotion using our proposed approach and baseline. **(a)** Learned multi-skill locomotion by following goal trajectories in Figure 4. **(b)** Baseline approach by manually switching between learned single skills. The robot failed after a discrete switch from bounding to galloping.



FIGURE 7 Versatile locomotion with continuous gait transitions on the real robot. **(a)** Estimated horizontal speed of the robot during multi-skill locomotion. **(b)** Expert weights showing that each motion utilizes all five experts, with one related expert being dominant.



3.1.4 Velocity estimation

During the deployment of multi-skill policies learned in simulation on real robots, sensing errors and uncertainties usually cause discrepancies between simulation and the real world. Unlike other state observations we selected for learning, base linear velocity cannot be obtained directly and needs to be estimated via leg kinematics or visual odometry, where the estimation results do not perform well during foot slipping or highly dynamic motions (Ji et al., 2022). Therefore, similar to Ji et al. (2022), we train a separate velocity estimator to obtain the estimation of unavailable or unreliable states given the sensory information of more reliable states. The input to the state estimator is 66-dimensional, including the gravity vector from roll and pitch measurements from the IMU, a two-step history of the gravity vector, the base angular velocity from the IMU, a two-step history of the base angular velocity, joint positions, a two-step history of joint positions, and joint velocity from motor encoders. The output is three-dimensional estimated base linear velocity. The estimator network is composed of two hidden layers, each with 256 neurons and a ReLU activation function. After we obtain the locomotion policies in simulation, we collect 215,000 pairs of input–output for training the estimator network via supervised learning. We use mean squared error loss (comparing ground-truth velocity with the velocity estimated by the neural network) for training, with a learning rate of 0.001, a weight decay of 0.0005, and a batch size of 1,024.

3.2 Optimized skill-switch criteria

Figure 5 shows that the best cost value keeps decreasing and reaches a local minimum value after 27 iterations of optimization, where the corresponding trot–bound and bound–gallop switch criteria are $x_1 = 2.2\text{ m}$ and $x_2 = 4.3\text{ m}$, respectively. It should be noted that the optimized gait-switch criteria are not exactly where the

gait transitions occur in practice since they are only incorporated into reward functions. Instead, the gait transitions during multi-skill locomotion are naturally learned via the optimized gait-switch criteria.

3.3 Multi-skill locomotion with continuous skill transitions

With the learned multi-skill locomotion policy, the robot is able to demonstrate trotting, bounding, galloping, and prompt fall recovery whenever necessary, as shown in Figure 6a and the Supplementary Video S1. The corresponding goal commands provided via joystick are shown in Figure 4. Here, we report only the experimental results obtained in the real world. The video shows more simulation and robustness tests.

3.3.1 Estimated speed

We show the estimated speed in the horizontal plane of the robot heading frame for 20 seconds of the multi-skill locomotion (Figure 7a). The robot shows an increasing velocity for trotting, bounding, and galloping skills.

3.3.2 Expert weights

Figure 7b shows the weights for each single-skill policy generated by the gating network correspondingly. The fall recovery expert dominates during the recovery motion. For the three locomotion gaits exhibited, each corresponding expert has the largest weight among all. However, compared to bounding and galloping gaits, where the corresponding expert dominates, the trotting expert co-acts more together with the other experts during trotting, contributing to the synthesized policy. Furthermore, to obtain a detailed view of the influence of each expert in different quadruped locomotion skills, we visualized the weight of each expert for the four demonstrated skills at different time-steps. Figure 8 clearly shows the composition of single-skill policies for each motion, with each motion utilizing all five single skills.

3.3.3 Euler angles

The corresponding roll and pitch angles are shown in Figure 9. When the robot encountered falls, the roll and pitch angles increased at first and returned to the normal range during fall recovery. In other cases, these two Euler angles have clear cyclic patterns. Moreover, the magnitude of the Euler angles increases as the robot progresses from trotting to bounding to galloping, indicating that the motion becomes more dynamic.

3.4 Robustness tests

Supplementary Video S1 showcases the robustness tests of the learned multi-skill policy in physics simulation, including (1) successfully traversing terrains with random obstacles (Figure 10), (2) locomoting with varying body mass, and (3) locomoting with input noise. Please refer to the video for the robot in action.

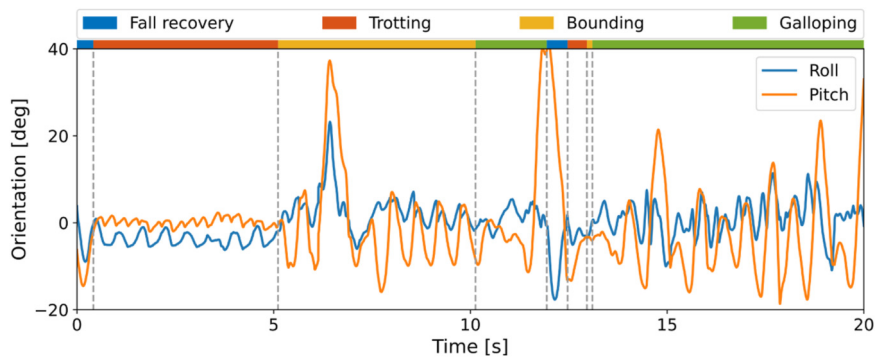


FIGURE 9 Roll and pitch angles during multi-skill locomotion in the real world.

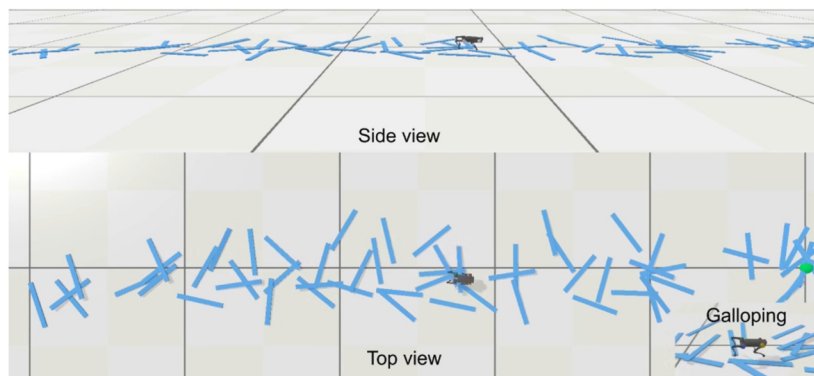


FIGURE 10 Learned multi-skill policy enabling the robot to traverse the terrain with random obstacles with natural gait transitions in physics simulation. The robot can perform highly dynamic galloping gait on a rough terrain.

3.5 Ablation studies

3.5.1 Distance as the gait-switch criterion: discrete switch vs. our approach

We compare our proposed multi-skill learning and optimization approach with the baseline approach, i.e., manual switching between different skill primitives. For the single skills, trotting and bounding were learned with a fixed desired velocity, while galloping was learned by maximizing velocity. After multi-skill learning with fixed parameters of each expert network, trotting motion is synthesized by the gating network at a lower speed range, and galloping motion is synthesized in a more dynamically feasible pattern. The snapshots in Figure 6 and Supplementary Video S1 contain more details.

For the baseline approach, the robot failed when manually switching from bounding to galloping, sometimes causing automatic shutdown due to the power protection for Unitree robots. In cases where failure occurs without triggering power protection, we can manually activate the fall-recovery skill, after which the robot can recover from the failure and return to a standing state. However, to resume locomotion, it requires another discrete switch from standing to trotting, causing further instability issues. In contrast, our multi-skill policy can directly transition from failure to

trotting without the intermediate phase in a dynamic fashion. When discretely switching from trotting to bounding at an improper gait phase, the knee joints of the rear legs may reach very close to the ground, or the front legs may lift very high in the following several time-steps. As shown in Figure 11, manual switch caused dynamic instability, such as abrupt changes in estimated velocity. In contrast, our approach enables a smoother and continuous gait transition in real-world deployment.

3.5.2 Velocity as the gait-switch criterion

In addition to robot's distance to the goal, our framework can also use other physical quantities as the gait-switch criteria, such as velocity. We formulate this in a velocity-tracking task setting. The velocity command to follow in the x-direction is sampled from the range of 0 ~ 3m/s during training. The goal-tracking reward terms are replaced with velocity-tracking reward terms. The reference foot-contact reward is segmented by the desired velocities for gait switching. For the outer optimization loop, the cost function is changed to minimize CoT as in Equation 9:

$$CoT = \frac{\sum_i^{12} \max(\tau_i \omega_i + \alpha \tau_i^2, 0)}{mg\|\mathbf{v}\|}, \tag{9}$$

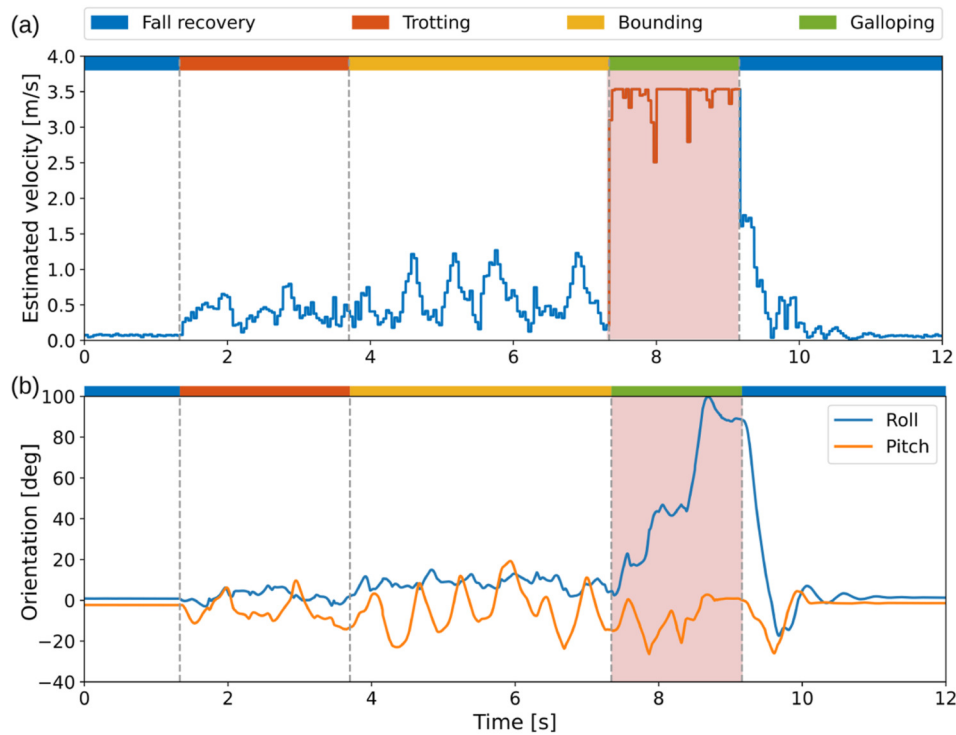


FIGURE 11 Performance of the baseline approach by manually switching from fall recovery to trotting to bounding in the real world. Robot failed to switch to galloping from bounding (red shaded areas) but was then able to perform a successful recovery from failure to standing still. **(a)** Estimated horizontal speed. **(b)** Roll and pitch angles.

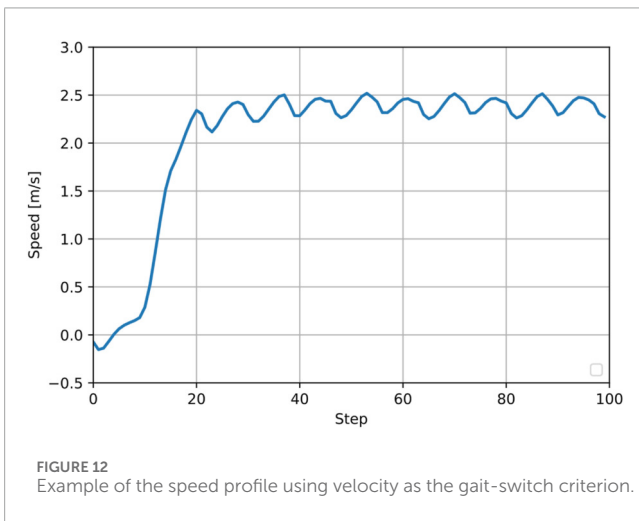


FIGURE 12 Example of the speed profile using velocity as the gait-switch criterion.

where τ_i and ω_i are the joint torque and joint velocity of the i th joint, respectively; m is the robot mass; g is gravity; $\|\mathbf{v}\|$ is the velocity norm; and $\alpha = 0.3$, which is robot-specific. We set the initial gait-switch criteria as $v_1 = 0.6m/s$ and $v_2 = 1.2m/s$ to warmstart optimization, with a population size of 50 and $\sigma = 0.2m/s$. The CMA-ES optimization runs for an iteration for every 20 iterations of inner-loop deep reinforcement learning. Figure 12 shows the smooth velocity curve when tracking a velocity command of $2.5m/s$.

In the following sections, we perform ablation studies on various design choices based on this implementation. In all subsequent ablation studies, the metric used to evaluate and benchmark learning performance is the training reward, and each training curve shows the mean and standard deviation of the reward across three training trials, averaged using a sliding window of 20 steps for visualization.

3.5.2.1 Outer optimization loop

We removed the outer optimization loop and retained only the hierarchical RL component of our framework and compared its learning performance with that of the full framework including the optimization loop. Figure 13a shows that our framework yields a higher reward than the baseline without the outer optimization loop.

3.5.2.2 Ablation on optimization parameters

Here, we ablate two important parameters in CMA-ES optimization. (1) Optimization update intervals with respect to RL updates. In our framework, we run one iteration of CMA-ES optimization for every 20 iterations of RL updates, which is noted as 1/20. In this ablation study, we compare this parameter 1/20 with 1/1 and 1/100. From Figure 13b, we find that optimizing one iteration per RL iteration results in a final reward that is far from optimal. Compared with one optimization per 100 RL updates, our implementation converges to a higher reward earlier. (2) Population size. Population size is the number of candidate solutions sampled for each generation in CMA-ES. In our

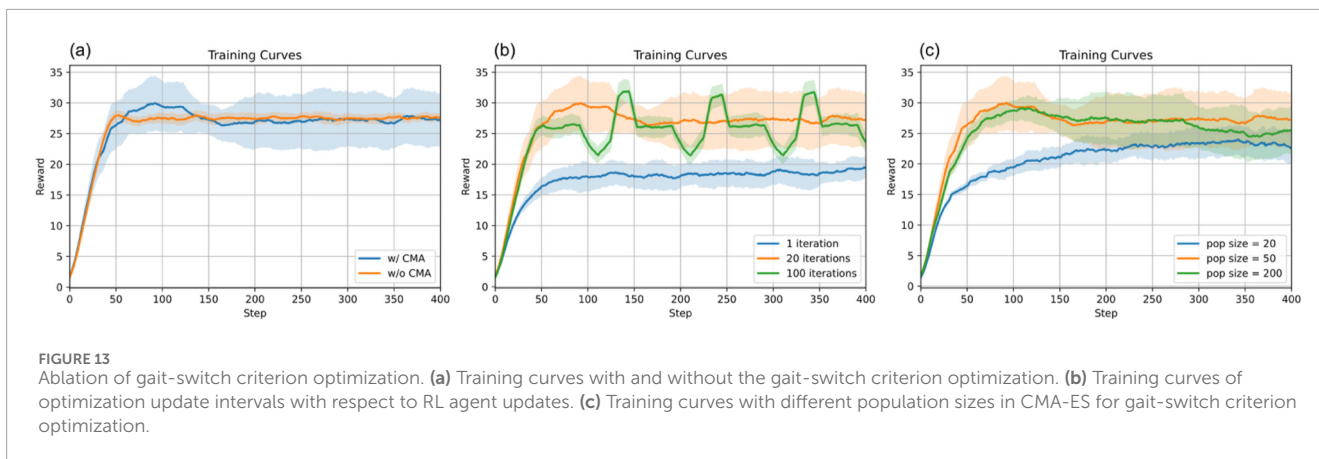


FIGURE 13 Ablation of gait-switch criterion optimization. (a) Training curves with and without the gait-switch criterion optimization. (b) Training curves of optimization update intervals with respect to RL agent updates. (c) Training curves with different population sizes in CMA-ES for gait-switch criterion optimization.

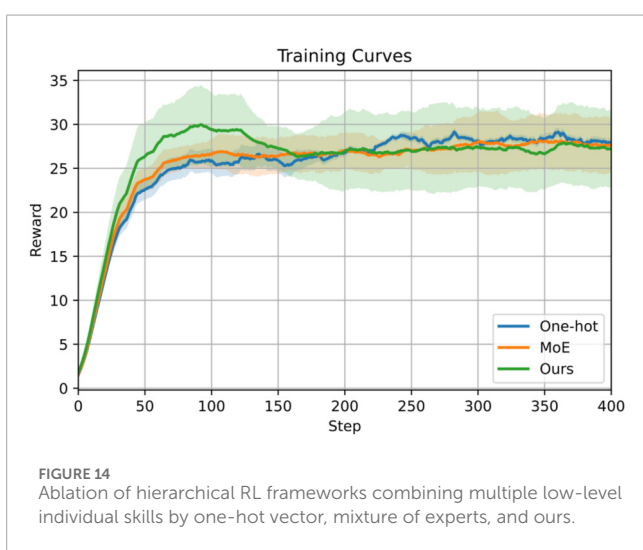


FIGURE 14 Ablation of hierarchical RL frameworks combining multiple low-level individual skills by one-hot vector, mixture of experts, and ours.

implementation, we set this parameter to 50. Here, we compare it with 20 and 200. From Figure 13c, we find that setting the population size as 20 is not sufficient to converge to a high reward. Compared to 100, our chosen population size converges to a slightly higher reward more quickly.

3.5.2.3 Ablation on hierarchical RL frameworks

Different hierarchical RL frameworks modulate and fuse low-level individual skills in different ways. Our framework uses a high-level gating network to modulate individual skills by fusing their Gaussian distributions. Here, we compare this implementation with two commonly used approaches. Note that the outer optimization loop is included in this comparison. (1) One-hot vector: The gating network learns to generate a one-hot vector to select one low-level skill per step. (2) MoE: The gating network learns to generate weights for the output of each expert that are summed as the final output. Figure 14 shows that MoE converges faster than the one-hot vector. Furthermore, our implementation of the hierarchical RL framework achieves a faster convergence rate and a higher reward than the one-hot vector and MoE baselines.

4 Discussion

This research developed a hierarchical learning and optimization framework to achieve multi-skill locomotion with optimized gait-switch criteria without the need for any reference trajectories or expert demonstrations. The robot demonstrates continuous gait transitions among trotting, bounding, and galloping skills as locomotion speed increases. Our learned multi-skill policy can also incorporate the fall recovery skill, which enables the robot to recover promptly and resume locomotion whenever it becomes unstable or falls during different gaits. Thus, the robot requires less human intervention to operate autonomously in a remote working space, enabling versatile applications. Compared with the existing end-to-end learning framework using a single policy, our hierarchical framework is bio-inspired and more efficient in fine-tuning for various tasks since it does not require learning from scratch to adapt to new tasks. Moreover, by optimizing gait-switch criteria as motor learning progresses, we avoid manually specifying the criteria with biased human knowledge distillation. The formulation can easily be adapted and generalized to different tasks or scenarios by customizing the cost function. It should be noted that the three-segment reward terms to encourage different foot-contact patterns based on the robot's distance to the goal (gait switch criteria) are discrete; however, the actual gait transition does not occur abruptly. This is because the distance we optimize is the desired distance to switch and not the actual distance, and the actual distance is also determined during learning by the other reward terms regarding smooth and stable locomotion.

One limitation of our approach is that we found that sim-to-real discrepancies still exist in the galloping motion. In the simulation, the galloping motion is more stable without any failures. In the real world, we cannot ensure a 100% success rate of galloping for very long periods. This can be due to various reasons. Compared to other locomotion skills, galloping is inherently a very unstable locomotion skill, since only one foot is in contact with the ground at one time. Slight sim-to-real discrepancies can cause huge differences and even failures, such as deformable foot pads on the Unitree A1 robot, ground friction, and velocity estimation for out-of-the-distribution motion. Another reason is that the goal commands used in real-world tests differ from those in the simulation. For training in simulation, we provide the goal position directly in the world frame,

while in the real world, due to the lack of body-position feedback, we provide normalized relative goal distance in the robot heading frame via joystick; thus, it is not possible to reproduce the same goal commands as in simulation via joystick.

For future work, we plan to further resolve the sim-to-real gap in galloping motion. We would also like to analyze the scalability of the proposed framework to more skills and more complex tasks as it would need more reward engineering. Furthermore, since our proposed approach can generate multi-skill locomotion data without any reference, one interesting application of our proposed framework would be preparing datasets for the training and fine-tuning of generalist policies for legged robots, such as diffusion models or OpenVLA (Kim et al., 2025).

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

WY: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Visualization, Writing – original draft. FA: Data curation, Writing – review and editing, Software. VA: Writing – review and editing, Data curation, Investigation. CY: Software, Writing – review and editing, Investigation, Methodology, Conceptualization. IH: Investigation, Writing – review and editing, Supervision, Validation. DK: Investigation, Writing – review and editing, Funding acquisition, Validation, Supervision. ZL: Investigation, Supervision, Conceptualization, Writing – review and editing, Methodology.

Funding

The author(s) declared that financial support was received for this work and/or its publication. This work was supported by the UKRI Future Leaders Fellowship [MR/V025333/1] (RoboHike).

References

- Bellegarda, G., Shafiee, M., and Ijspeert, A. (2025). “Allgaits: learning all quadruped gaits and transitions,” in 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, 15929–15935.
- Caluwaerts, K., Iscen, A., Kew, J. C., Yu, W., Zhang, T., Freeman, D., et al. (2023). Barkour: benchmarking animal-level agility with quadruped robots. *arXiv preprint arXiv:2305.14654*. doi:10.48550/arXiv.2305.14654
- Castano, J. A., Zhou, C., and Tsagarakis, N. (2019). “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), 41–46.
- Chen, A. S., Lessing, A. M., Tang, A., Chada, G., Smith, L., Levine, S., et al. (2025). “Commonsense reasoning for legged robot adaptation with vision-language models,” in 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, 12826–12833. doi:10.1109/ICRA55743.2025.11127234
- Cheng, X., Shi, K., Agarwal, A., and Pathak, D. (2024). “Extreme parkour with legged robots,” in IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 11443–11450. doi:10.1109/ICRA57147.2024.10610200
- Cordie, T., Roberts, J., Dunbabin, M., Dungavell, R., and Bandyopadhyay, T. (2024). Enabling robustness to failure with modular field robots. *Front. Robotics AI* 11, 1225297. doi:10.3389/frobt.2024.1225297
- Fu, Z., Kumar, A., Malik, J., and Pathak, D. (2021). “Minimizing energy consumption leads to the emergence of gaits in legged robots,” in *Conference on robot learning (PMLR)*.
- Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). “Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International conference on machine learning ICML 2018* (Stockholm, Sweden: Proceedings of Machine Learning Research (PMLR)), 1861–1870.
- Hansen, N. (2016). The cma evolution strategy: a tutorial. *arXiv preprint arXiv:1604.00772*. doi:10.48550/arXiv.1604.00772
- Hattori, S., Suzuki, S., Fukuhara, A., Kano, T., and Ishiguro, A. (2025). Bicycle-inspired simple balance control method for quadruped robots in high-speed running. *Front. Robotics AI* 11, 1473628. doi:10.3389/frobt.2024.1473628

Acknowledgements

The authors would like to thank Jianwei Liu from University College London and Daniel Marques and Jacques Cloete from Oxford Robotics Institute for helping with the robot experiments.

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declared that generative AI was used in the creation of this manuscript. The last image in Figure 1a was generated by ChatGPT.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2026.1697159/full#supplementary-material>

- He, T., Zhang, C., Xiao, W., He, G., Liu, C., and Shi, G. (2024). "Agile but safe: learning collision-free high-speed legged locomotion," in *Robotics: science and systems*.
- Hoyt, D. F., and Taylor, C. R. (1981). Gait and the energetics of locomotion in horses. *Nature* 292, 239–240. doi:10.1038/292239a0
- Huang, X., Chi, Y., Wang, R., Li, Z., Peng, X. B., Shao, S., et al. (2024). "Diffuseloco: real-time legged locomotion control with diffusion from offline datasets," in *Conference on robot learning*.
- Humphreys, J., Li, J., Wan, Y., Gao, H., and Zhou, C. (2023). Bio-inspired gait transitions for quadruped locomotion. *IEEE Robotics Automation Lett.* 8, 6131–6138. doi:10.1109/lra.2023.3300249
- Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., et al. (2019). Learning agile and dynamic motor skills for legged robots. *Sci. Robotics* 4, eaau5872. doi:10.1126/scirobotics.aau5872
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation* 3, 79–87. doi:10.1162/neco.1991.3.1.79
- Ji, G., Mun, J., Kim, H., and Hwangbo, J. (2022). Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robotics Automation Lett.* 7, 4630–4637. doi:10.1109/lra.2022.3151396
- Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., et al. (2025). "Openvla: an open-source vision-language-action model," in Proceedings of The 8th Conference on Robot Learning, 270, 2679–2713.
- Liang, B., Sun, L., Zhu, X., Zhang, B., Xiong, Z., Li, C., et al. (2025). "Adaptive energy regularization for autonomous gait transition and energy-efficient quadruped locomotion," in 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, 5350–5356. doi:10.1109/ICRA55743.2025.11128812
- Margolis, G. B., and Agrawal, P. (2023). "Walk these ways: tuning robot control for generalization with multiplicity of behavior," in *Conference on robot learning (PMLR)*, 22–31.
- Miranda, S., Vázquez, C. R., and Navarro-Gutiérrez, M. (2025). Energy consumption analysis and optimization in collaborative robots. *Front. Robotics AI* 12, 1671336. doi:10.3389/frobt.2025.1671336
- Mitchell, A. L., Merkt, W., Papatheodorou, A., Havoutis, I., and Posner, I. (2024). "Gaitor: learning a unified representation across gaits for real-world quadruped locomotion," in *Conference on robot learning (PMLR)*.
- Owaki, D., and Ishiguro, A. (2017). A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping. *Sci. Reports* 7, 1–10. doi:10.1038/s41598-017-00348-9
- O'Mahoney, R., Mitchell, A. L., Yu, W., Posner, I., and Havoutis, I. (2025). "Offline adaptation of quadruped locomotion using diffusion models," in 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, 9974–9980. doi:10.1109/ICRA55743.2025.11128726
- Peng, X. B., Chang, M., Zhang, G., Abbeel, P., and Levine, S. (2019). "Mcp: learning composable hierarchical control with multiplicative compositional policies," in 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Advances in Neural Information Processing Systems, Vancouver, Canada, 32.
- Reske, A., Carius, J., Ma, Y., Farshidian, F., and Hutter, M. (2021). "Imitation learning from mpc for quadrupedal multi-gait control," in 2021 IEEE international conference on robotics and automation (ICRA) (IEEE), 5014–5020.
- Shafiee, M., Bellegarda, G., and Ijspeert, A. (2024). Viability leads to the emergence of gait transitions in learning agile quadrupedal locomotion on challenging terrains. *Nat. Commun.* 15, 3073. doi:10.1038/s41467-024-47443-w
- Shao, Y., Jin, Y., Liu, X., He, W., Wang, H., and Yang, W. (2021). Learning free gait transition for quadruped robots via phase-guided controller. *IEEE Robotics Automation Lett.* 7, 1230–1237. doi:10.1109/lra.2021.3136645
- Tang, Y., Yu, W., Tan, J., Zen, H., Faust, A., and Harada, T. (2023). "Saytap: language to quadrupedal locomotion," in Proceedings of The 7th Conference on Robot Learning, 229, 3556–3570.
- Yang, C., Yuan, K., Zhu, Q., Yu, W., and Li, Z. (2020). Multi-expert learning of adaptive legged locomotion. *Sci. Robotics* 5, eabb2174. doi:10.1126/scirobotics.abb2174
- Yang, Y., Zhang, T., Coumans, E., Tan, J., and Boots, B. (2022). "Fast and efficient locomotion via learned gait transitions," in *Conference on robot learning (PMLR)*, 773–783.
- Yu, W., Yang, C., McCreavy, C., Triantafyllidis, E., Bellegarda, G., Shafiee, M., et al. (2023). Identifying important sensory feedback for learning locomotion skills. *Nat. Mach. Intell.* 5, 919–932. doi:10.1038/s42256-023-00701-w
- Yuan, K., and Li, Z. (2022). Multi-expert synthesis for versatile locomotion and manipulation skills. *Front. Robotics AI* 9, 970890. doi:10.3389/frobt.2022.970890
- Zhuang, Z., Fu, Z., Wang, J., Atkeson, C., Schwertfeger, S., Finn, C., et al. (2023). "Robot parkour learning," in *Conference on robot learning (PMLR)*, 73–92.