machines

MDPI

Article

Quadruped Robot Motion Control Based on an Improved PPO Algorithm

Erbiao Yu [1,2,†], Shunlu Wang [3,†], Zuhou Teng [4,†], Lei Wang [4,†] and Xiaoteng Tang [4,*]

[1] Department of Automotive and Modern Transportation, Fuzhou Liming Vocational & Technical College, Fuzhou 350109, China
[2] School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou 350108, China
[3] School of Statistics and Mathematics, Zhejiang Gongshang University, Hangzhou 310018
[4] College of Physics and Electronic Information Engineering, Minjiang University, Fuzhou 350108, China; 13398202103@163.com (L.W.)
* Correspondence: tang_xt@163.com
† These authors contributed equally to this work and share first authorship.

Abstract

This paper proposes LA-PPO, an improved Proximal Policy Optimization algorithm for quadruped robot locomotion control on mixed terrain. To address partial observability, temporal dependence in contact states, and non-uniform importance of historical information in complex-terrain quadruped locomotion, LA-PPO integrates Long Short-Term Memory (LSTM) and Multi-Head Attention (MHA) within an Actor–Critic framework. The LSTM module models temporal dependencies in historical observations, while the MHA module adaptively emphasizes historical information most relevant to the current action decision. Based on IsaacGym, we construct a mixed-terrain environment consisting of flat regions, sloped regions, and random rough-terrain regions and conduct algorithmic comparisons, statistics over multiple random seeds, reward component ablation studies, and attention mechanism analyses for both walking and trotting gaits. Simulation results show that LA-PPO achieves the highest final reward and the longest mean episode length in both gaits. Compared with the PPO baseline, the final reward and mean episode length are improved by approximately 42.3% and 42.7%, respectively, in the walking task, and by approximately 39.8% and 25.7%, respectively, in the trotting task. Real-robot tests further show that the learned policy can perform walking and trotting on flat ground, sloped terrain, and random rough terrain, demonstrating preliminary sim-to-real transfer capability.

Keywords: quadruped robot; Proximal Policy Optimization; Long Short-Term Memory; Multi-Head Attention

Check for updates

Academic Editor: Zheng Chen

Received: 12 April 2026
Revised: 15 May 2026
Accepted: 19 May 2026
Published: 30 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Quadruped robots, by virtue of their legged morphology, can cross obstacles, traverse slopes, and maintain mobility on terrains where wheeled platforms are difficult to operate stably. They are therefore well suited to mobile tasks in unstructured environments [1–3]. However, stable and adaptive quadruped locomotion control remains challenging because it involves high-dimensional continuous control, intermittent foot–ground contact, actuator constraints, sensor noise, and terrain disturbances. Classical control methods, such as Model Predictive Control (MPC) [4] and Zero-Moment Point (ZMP)-based stability methods [5, 6], can perform well when robot models, contact timing, and terrain assumptions are sufficiently accurate. Under unknown slopes, rough ground, uncertain contact, or gait

transition scenarios, however, these methods usually require careful modeling and manual parameter tuning, and their robustness can be affected [2,7]. Consequently, increasing attention has been paid to learning-based locomotion control methods that enable robots to learn adaptive policies through interaction with simulated environments.

Deep Reinforcement Learning (DRL) can directly optimize closed-loop control policies without relying on complete analytical models, and has been applied to agile quadruped locomotion, natural rough-terrain walking, rapid environmental adaptation, perceptive locomotion, and complex obstacle navigation [3,7–10]. Among common DRL algorithms, off-policy methods such as Soft Actor–Critic (SAC) [11], Twin Delayed Deep Deterministic Policy Gradient (TD3) [12], and Deep Deterministic Policy Gradient (DDPG) [13] can reuse samples stored in replay buffers and therefore achieve high data efficiency in continuous control tasks. Nevertheless, in legged locomotion tasks with rich contact dynamics, large-scale parallel sampling, and complex reward compositions, training stability and hyperparameter sensitivity remain important concerns. Proximal Policy Optimization (PPO) [14] updates policies using on-policy data and constrains excessive policy changes through a clipped surrogate objective; it is therefore widely used in GPU-based massively parallel simulation and sim-to-real legged locomotion research [9,15]. Based on these considerations, this study adopts PPO as the base optimization algorithm and compares it with SAC, TD3, and DDPG as representative off-policy baselines.

For partial observability and historical dependence in legged locomotion control, existing studies can be broadly divided into two structural directions. The first introduces recurrent memory modules, such as LSTM or RNN, into the policy or value network so that the policy can infer contact quality, dynamic parameters, and external disturbances that are difficult to identify from a single observation frame [16,17]. The second introduces attention mechanisms or Transformer structures to fuse and reweight proprioceptive, exteroceptive, or historical features, thereby emphasizing information that is more relevant to the current decision [7,18,19]. These studies show that both recurrent memory and attention mechanisms can improve the use of historical information on complex terrains. However, existing methods usually emphasize recurrent memory or attention/Transformer structures for specific functions. For low-level quadruped locomotion control on mixed terrain, targeted structural design and systematic comparison are still needed to exploit both the temporal memory capability of LSTM and the critical history selection capability of MHA within the same PPO Actor–Critic policy.

Based on the above analysis, we propose LA-PPO, a cascaded LSTM–MHA PPO structure for low-level quadruped locomotion control on mixed terrain. The core idea of LA-PPO is to connect LSTM temporal encoding and MHA-based key historical segment selection within the same PPO Actor–Critic policy. Specifically, the model first uses LSTM to temporally encode the most recent eight steps of proprioceptive, foot contact, and local terrain observations. Then, it uses the current LSTM hidden state as the query and the complete historical hidden sequence as the keys and values, allowing MHA to adaptively reweight critical moments within the historical window.

Compared with PPO using only LSTM, LA-PPO does not rely solely on the recurrent network to implicitly compress historical information. Instead, it explicitly computes attention weights over the complete sequence of LSTM hidden states, thereby emphasizing time segments that are more relevant to the current action decision. Compared with PPO using only MHA, LA-PPO first models temporal continuity in contact switching, velocity error evolution, and posture recovery through recurrent units before applying attention-based reweighting. In this way, LA-PPO makes temporal dependency modeling and critical historical information selection complementary within the same on-policy

optimization framework, improving the stability and robustness of quadruped gait control on mixed terrain.

The main contributions of this paper are as follows:

1. We propose the LA-PPO framework, which integrates LSTM and MHA for quadruped robot locomotion control on mixed terrain. The framework adopts a cascaded design of "LSTM temporal encoding + single-query MHA-based historical reweighting", simultaneously modeling temporal dependence and key historical segment selection in the low-level policy network. This distinguishes it from structural variants that rely only on recurrent encoding or only on attention-based reweighting.
2. We design a reward function that includes velocity tracking, posture stability, a ZMP-inspired support margin stability term, and gait priors. Reward ablation experiments quantify the influence of each component on walking and trotting performance.
3. We conduct algorithmic comparisons, statistical evaluation over multiple random seeds, and attention-weight analysis in a mixed terrain composed of flat, sloped regions, and random rough-terrain regions. Real-robot tests are further conducted to preliminarily validate the deployability of the learned policy.

2. PPO Algorithm Integrating LSTM and MHA

2.1. PPO Algorithm

The Proximal Policy Optimization (PPO) algorithm is a representative policy-gradient method. Policy-gradient methods directly optimize a parameterized policy by updating policy parameters along the gradient direction to maximize expected cumulative reward. The general form of the policy gradient is:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a)] \tag{1}$$

where s and a denote the state and action, respectively, π_θ is the parameterized policy, $Q^{\pi_\theta}(s,a)$ is the action-value function, and θ represents the policy parameters.

Compared with algorithms such as TRPO, PPO is simpler to implement, computationally more efficient, and more stable during training [14,20]. PPO has two main implementation forms: PPO-Clip and PPO-Penalty. PPO-Clip directly limits the magnitude of policy updates through a clipping mechanism, making it intuitive and efficient. PPO-Penalty controls updates by introducing a KL-divergence penalty term, which is theoretically more rigorous but requires more complex parameter tuning. Therefore, this study adopts PPO-Clip, whose objective function is defined as:

$$L^{CLIP}(\theta) = \mathbb{E}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right] \tag{2}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the policy ratio, $\hat{A}_t$ is the estimated advantage function, ϵ is the clipping parameter, usually set to 0.2, and $\text{clip}(x, 1-\epsilon, 1+\epsilon)$ denotes the truncation function that constrains x to the interval $[1-\epsilon, 1+\epsilon]$. The clipping mechanism prevents the policy update from deviating excessively, thereby avoiding performance collapse caused by overly large updates.

In advantage estimation, there is a trade-off between variance and bias. Single-step temporal difference estimates usually have low variance but high bias, whereas estimates based on complete trajectories often have low bias but high variance. Generalized Advantage Estimation (GAE) balances bias and variance by exponentially averaging multi-step returns, improving the accuracy of advantage estimation. GAE is calculated as follows:

$$\hat{A}_t^{\text{GAE}}(\gamma,\lambda) = \sum_{l=0}^{\infty}(\gamma\lambda)^l \delta_{t+l} \tag{3}$$

where $\delta_{t+l} = r_{t+l} + \gamma V(s_{t+l+1}) - V(s_{t+l})$ is the temporal difference error, γ is the discount factor, and λ is the GAE parameter used to control the bias–variance trade-off.

2.2. Long Short-Term Memory Networks

LSTM is a specialized type of Recurrent Neural Network (RNN) commonly used to alleviate the gradient vanishing and gradient exploding problems that occur when traditional RNNs model long sequences. With memory cells and gating mechanisms, LSTM can retain historical information relevant to the current decision. In quadruped locomotion control, foot contact switching, velocity error variation, and body posture adjustment all exhibit strong temporal correlations. Therefore, we use LSTM to encode historical observations and provide continuous temporal features for the subsequent policy output. As shown in Figure 1, an LSTM unit mainly includes the following structures:

1. Forget gate: This gate determines what information should be discarded from the cell state. It applies a sigmoid activation function to the current input and previous hidden state, producing a value between 0 and 1 to indicate the proportion of information to retain or forget:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \tag{4}$$

2. Input gate: This gate determines what information is written into the cell state. The sigmoid function outputs a value between 0 and 1 to control the update ratio, while the hyperbolic tangent function tanh generates the candidate cell state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{5}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C). \tag{6}$$

3. Cell-state update: The cell state is updated according to the outputs of the forget and input gates:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t. \tag{7}$$

4. Output gate: This gate determines what information is output from the cell state at the current time step. The sigmoid function generates the output ratio, and the tanh function activates the cell state:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{8}$$

$$h_t = o_t \odot \tanh(C_t). \tag{9}$$

where h_t is the current hidden state, x_t is the current input, and all W and b parameters are updated during training. Here, σ denotes the sigmoid function, tanh denotes the hyperbolic tangent function, and $\odot$ denotes element-wise multiplication. The sigmoid and tanh functions are:

$$\sigma(x) = \frac{1}{1+e^{-x}}, \quad \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{10}$$

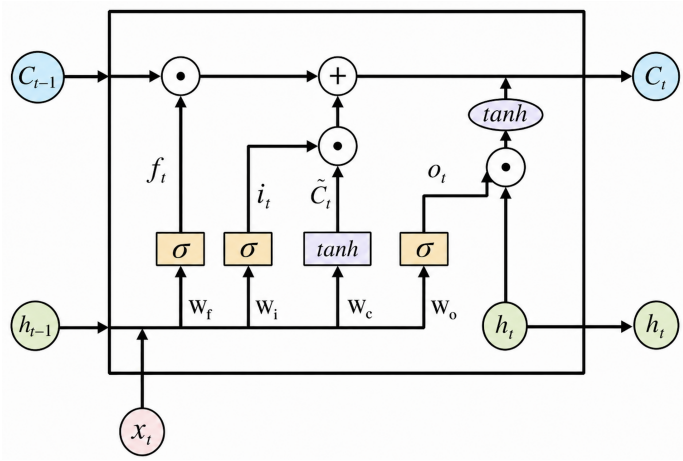


Figure 1. Long Short-Term Memory cell structure.

2.3. Multi-Head Attention Mechanism

Attention mechanisms assign weights to different positions in an input sequence, enabling a model to emphasize information that is more relevant to the current task. The general form is:

$$\text{Attention}(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{11}$$

where Q, K, and V denote the query, key, and value matrices, respectively. These matrices have compatible dimensions, and d_k denotes the dimension of each key vector. Dividing the dot product by $\sqrt{d_k}$ prevents excessively large dot-product values when the feature dimension increases, thereby reducing softmax saturation.

Depending on the task, attention mechanisms may take different forms. To illustrate the basic computation for same-source sequence attention, the standard self-attention form is first given. The query, key, and value are all obtained from the same input sequence through linear transformations:

$$\text{Self-Attention}(X) = softmax\left(\frac{XW_Q(XW_K)^T}{\sqrt{d_k}}\right)XW_V. \tag{12}$$

where X is the input sequence, and W_Q, W_K, and W_V are learnable parameter matrices used to map the input sequence to the corresponding query, key, and value matrices.

To enable the model to attend to information from different perspectives and representation subspaces simultaneously, we introduce the MHA mechanism. MHA is essentially a parallel extension of single-head attention; it linearly projects queries, keys, and values into different representation subspaces for multi-view feature extraction. The structure of MHA is shown in Figure 2, and its calculation is:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_n)W_o. \tag{13}$$

where W_o is a learnable parameter matrix, $\text{head}_i = \text{Attention}(Q_i, K_i, V_i)$, and Concat denotes concatenation.

In LA-PPO, we do not directly perform full self-attention over the raw observation sequence. Instead, single-query temporal attention is constructed over the hidden sequence encoded by LSTM. The current hidden state is used as the query, and the historical hidden sequence is used as the keys and values to aggregate historical information relevant to the current action decision.

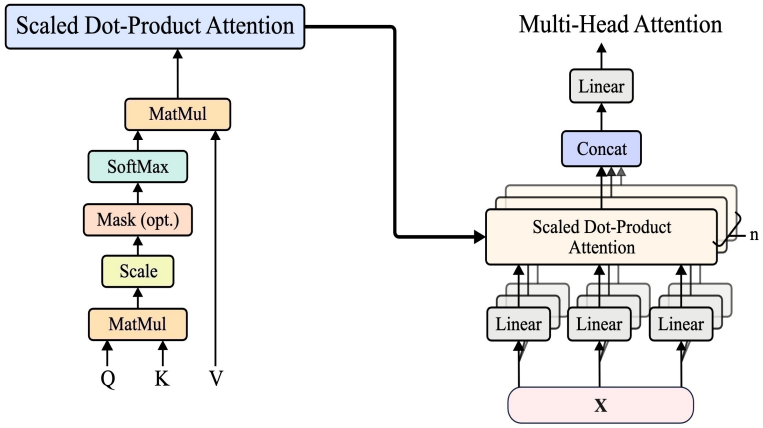


Figure 2. Structure of the Multi-Head Attention mechanism.

2.4. Integration of LSTM and Multi-Head Attention

PPO usually adopts the Actor–Critic paradigm; therefore, the structures of the policy and value networks directly affect the use of historical information. LSTM-PPO captures temporal dependence by introducing LSTM, but in complex locomotion sequences, different historical time steps contribute differently to the current decision. To enhance selective use of critical temporal cues, this study further introduces MHA on top of the LSTM-PPO backbone to form LA-PPO.

The Actor and Critic of LA-PPO use two completely independent networks. The two networks have the same structure but do not share any parameters. Specifically, the Input FC, LSTM, MHA, and MLP heads are modeled separately, and no shared encoder exists between the Actor and Critic. Each branch performs forward computation in the order: historical observation input $\rightarrow$ input mapping $\rightarrow$ LSTM $\rightarrow$ MHA $\rightarrow$ MLP output layer.

More specifically, each branch first receives an eight-step history of observations, which is reshaped from a flat representation into a temporal tensor with 196 dimensions at each time step. The sequence is first passed through an input mapping layer that projects the features at each time step from 196 dimensions to 128 dimensions. It is then fed into a two-layer LSTM, producing a temporal hidden representation of 256 dimensions at each time step.

At the connection between LSTM and MHA, we adopt single-query temporal attention. The LSTM hidden state at the last time step is used as the query vector, while the complete LSTM output sequence in the historical time window is used simultaneously as the keys and values. The MHA module then performs weighted aggregation within this history window and generates a context representation most relevant to the current decision.

The context representation is subsequently passed through a linear mapping to obtain branch-specific features. Finally, the Actor branch outputs a 12-dimensional action mean, and the Critic branch outputs a one-dimensional state value. The policy distribution is diagonal Gaussian: the action standard deviation is not directly output by the network forward pass but is represented by an independent learnable parameter vector with the same dimensionality as the action. This vector, together with the action mean, forms the policy distribution. Thus, LSTM mainly undertakes temporal-dependence modeling, while MHA aggregates information that is more critical for the current decision from the historical window.

The network implementation uses the ELU activation function. The fully connected hidden layers of both Actor and Critic are $[512, 256, 128]$, and the two branches have the same structure but no shared parameters. The historical sequence is maintained by an environment buffer in a FIFO manner as the most recent eight observations. It is stored in flat form and reshaped to $[8, 196]$ during the policy forward pass. MHA is configured

with four heads, each with 64 dimensions (total dimension 256), and the attention dropout is 0.0. This parameter combination is consistent with the structure shown in Figure 3 and is intended to empirically balance temporal modeling capability and training stability.

During training, let $O_t = [o_{t-7}, \ldots, o_t]$ denote the historical observation window input to the policy. The Actor parameters θ include its Input FC, LSTM, MHA, MLP head, and action standard-deviation parameters, while the Critic parameters ϕ include its independent Input FC, LSTM, MHA, and value head. The end-to-end objective of LA-PPO can be written as

$$\mathcal{L}_{\text{LA-PPO}}(\theta, \phi) = -L^{CLIP}(\theta) + c_v \, \mathbb{E}_t \left[(V_\phi(O_t) - \hat{R}_t)^2 \right] - c_e \, \mathbb{E}_t [\mathcal{H}(\pi_\theta(\cdot|O_t))], \quad (14)$$

where $\hat{R}_t$ is the target return estimated by GAE, and c_v and c_e are the value-loss coefficient and entropy coefficient, respectively. During backpropagation, gradients from the policy loss and entropy term are propagated from the Actor output to its MHA, LSTM, and input-mapping layers. Gradients from the value loss are propagated from the Critic output to its corresponding modules. Because the Actor and Critic do not share parameters, gradients from one branch do not propagate into the other branch.

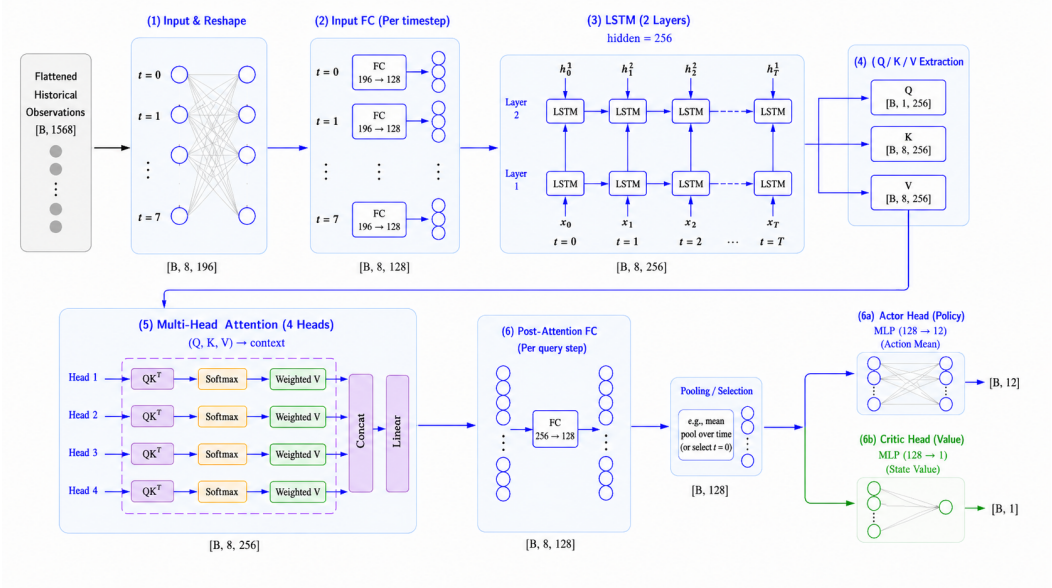


Figure 3. LA-PPO network architecture.

3. Experimental Design

3.1. Observation Space and Action Space

In the reinforcement-learning implementation, the policy input is not equivalent to the complete dynamic state of the robot, but is composed of measurable or estimable observations. We construct a 196-dimensional single-step observation vector for the quadruped robot, including four types of features: body motion, joint motion, foot contact, and local terrain height.

1. Body-state vector: a six-dimensional vector composed of the three-dimensional linear velocity and three-dimensional angular velocity of the robot base in the world coordinate system. It describes the overall translational and rotational motion state of the robot.
2. Joint-state vector: a 36-dimensional vector composed of the current angular positions (12 dimensions), angular velocities (12 dimensions), and joint torques (12 dimensions) of the robot's 12 actuated joints. It characterizes the instantaneous motion state of the leg actuators.

3. Foot-contact-state vector: a four-dimensional binary vector representing the sole contact state of the four legs. Each bit corresponds to one leg, where 1 indicates contact and 0 indicates no contact.
4. Terrain-height vector: a 150-dimensional vector constructed from ground-height measurements at 150 sampling points around the robot. It provides local terrain-undulation information.

Therefore, the observation vector at time t is expressed as:

$$o_t = [o_t^{body}, o_t^{joint}, o_t^{foot}, o_t^{terrain}] \in \mathbb{R}^{196}. \tag{15}$$

The total dimension of this observation vector is 196, which is consistent with the input dimension of the policy network.

The action space is designed as a 12-dimensional continuous vector representing the target joint angles of the 12 actuated joints of the quadruped robot. Each action component corresponds to the desired angular position of one joint, enabling precise control of robot-limb motion. The action space can be expressed as:

$$a_t = [a_{1,t}, a_{2,t}, \ldots, a_{12,t}] \in \mathbb{R}^{12}, \tag{16}$$

where $a_{i,t}$ represents the target angle of the ith joint at time t.

3.2. Reward Function

The reward function simultaneously constrains motion objectives and guides gait patterns. Considering that this study separately trains walking and trotting gaits, the reward function consists of shared motion terms and gait-specific terms. To avoid ambiguity in notation, the shape parameters of sub-rewards are separated from the total reward weights: only shape parameters are retained inside each sub-reward, while gait contributions are uniformly controlled by w_{walk} and w_{trot} in the total reward. The detailed design is as follows:

1. Velocity-tracking reward: This term encourages the robot to move at the target velocity and is a weighted combination of linear- and angular-velocity tracking terms:

$$r_{vel} = \beta_v \exp(-\alpha_v \|v_{xy} - v_{xy}^{target}\|^2) + \beta_\omega \exp(-\alpha_\omega \|\omega_z - \omega_z^{target}\|^2), \quad \beta_v + \beta_\omega = 1. \tag{17}$$

Here, v_{xy} is the current linear velocity on the horizontal plane, v_{xy}^{target} is the target linear velocity, ω_z is the current angular velocity about the vertical axis, ω_z^{target} is the target angular velocity, α_v and α_ω are scaling parameters, and β_v and β_ω are the internal combination coefficients for linear- and angular-velocity tracking.

2. Posture-control reward: This term maintains body stability by penalizing roll and pitch angles:

$$r_{pose} = \exp(-\alpha_p(\phi^2 + \theta^2)). \tag{18}$$

Here, ϕ and θ denote the current roll and pitch angles, respectively, and α_p is the scaling parameter.

3. ZMP-inspired support-margin reward: Inspired by the Zero-Moment Point (ZMP) stability criterion and the support polygon [6], this reward is designed as follows. When the contact points are insufficient to form a stable support polygon, a small stability reward is given. When a support polygon can be formed, the reward is computed from the minimum distance between an approximate support point and the boundary of the support polygon. Let the planar foot positions be $p_j \in \mathbb{R}^2$,

$j \in \{\mathrm{FL}, \mathrm{FR}, \mathrm{RL}, \mathrm{RR}\}$, the contact indicator be $c_j \in \{0, 1\}$, the contact set be $\mathcal{C} = \{j \mid c_j = 1\}$, and the number of contact legs be $n_c = |\mathcal{C}|$. In the implementation,

$$\hat{z} = \frac{1}{4} \sum_j p_j \tag{19}$$

is used as a planar approximation of the support-stability evaluation point. This implementation is not a strict dynamic ZMP calculation; instead, it approximates the projection of the center of mass by the mean position of the four feet. When $n_c \geq 3$, a support-polygon edge set $\mathcal{E}$ is constructed from the contact feet: for three contacts, it forms a triangle; for four contacts, the contact points are first sorted by their polar angles relative to their geometric center and then connected into a quadrilateral edge set. The minimum distance is then computed as

$$d_{min} = \min_{e \in \mathcal{E}} d(\hat{z}, e), \tag{20}$$

where the point-to-segment distance is

$$d(\hat{z}, [a, b]) = \|\hat{z} - (a + \tau^*(b - a))\|_2, \quad \tau^* = \mathrm{clip}\left(\frac{(\hat{z} - a)^\top (b - a)}{\|b - a\|_2^2}, 0, 1 \right). \tag{21}$$

The support-stability reward is then written in piecewise form:

$$r_{zmp} = \begin{cases} 0.1\, n_c, & n_c < 3, \\ \mathrm{clip}\left(\dfrac{d_{min}}{0.10}, 0, 1 \right), & n_c = 3, \\ \mathrm{clip}\left(\dfrac{d_{min}}{0.15}, 0, 1 \right), & n_c = 4. \end{cases} \tag{22}$$

Equivalently, when $n_c \in \{3, 4\}$, it can be written as $r_{zmp} = \mathrm{clip}(\alpha_z d_{min}, 0, 1)$, where $\alpha_z = 1/d_{ref}$; $d_{ref} = 0.10\,\mathrm{m}$ for three-leg support and $d_{ref} = 0.15\,\mathrm{m}$ for four-leg support. This design makes the reward monotonically related to the safety margin between the approximate support point and the support boundary: the larger the distance, the higher the stability reward.

In the above definitions, p_j is the planar coordinate of the jth foot; c_j is the contact state (1 for contact and 0 for swing); $\mathcal{C}$ is the index set of contact legs; n_c is the number of contact legs; $\hat{z}$ is the planar approximation of the support-stability evaluation point; $\mathcal{E}$ is the edge set of the support polygon; $e = [a, b]$ denotes a line segment corresponding to one edge, with endpoints $a, b \in \mathbb{R}^2$; $d(\hat{z}, e)$ is the Euclidean distance from the point to the segment; d_{min} is the minimum distance from $\hat{z}$ to all edges; τ^* is the projection coefficient clipped to $[0, 1]$ by $\mathrm{clip}(\cdot, 0, 1)$; $\|\cdot\|_2$ denotes the two-norm; d_{ref} is the distance-normalization scale; and α_z is the corresponding normalization coefficient.

The above three reward terms are shared by all experiments. The following reward terms are designed for specific gait tasks.

4. Walking-gait reward: This term is used for walking and encourages a stable pattern with one swing leg and three supporting legs. It is activated only during walking-gait training. Let the foot contact state be $c_i \in \{0, 1\}$, $i \in \{\mathrm{FL}, \mathrm{FR}, \mathrm{RL}, \mathrm{RR}\}$:

$$r_{\mathrm{walk}} = \mathbb{I}\left(\sum_i c_i = 3 \right). \tag{23}$$

Here, $\mathbb{I}(\cdot)$ denotes the indicator function, which returns 1 when the condition is satisfied and 0 otherwise. The condition $\sum_i c_i = 3$ indicates that exactly three feet are in contact with the ground. In the project implementation, the two equivalent criteria of "three feet in contact" and "one foot in swing" are combined with weights of 0.5/0.5, which is numerically consistent with the above expression.

5. Diagonal-trot gait reward: To encourage a natural diagonal trot gait, we design a gait reward based on foot-contact states and diagonal-leg coordination. It is activated only during trotting-gait training. Let $c_{FL}, c_{FR}, c_{RL}, c_{RR} \in \{0, 1\}$ denote the foot-contact states:

$$r_{\text{trot}} = 0.5 \times \frac{\mathbb{I}(c_{FL} = c_{RR}) + \mathbb{I}(c_{FR} = c_{RL})}{2} + 0.5 \times \mathbb{I}(c_{FL} \neq c_{FR}). \tag{24}$$

The condition $c_{FL} = c_{RR}$ indicates that the front-left and rear-right feet (diagonal pair 1) have the same contact state; $c_{FR} = c_{RL}$ indicates that the front-right and rear-left feet (diagonal pair 2) have synchronized contact states; and $c_{FL} \neq c_{FR}$ constrains the two diagonal leg groups to form an alternating contact pattern.

The final reward is a weighted sum of the sub-rewards. According to the training task of each gait, the total reward is defined as follows.

For the walking gait:

$$R_{\text{total}}^{\text{walk}} = w_{vel} r_{vel} + w_{pose} r_{pose} + w_{zmp} r_{zmp} + w_{walk} r_{\text{walk}}. \tag{25}$$

For the trotting gait:

$$R_{\text{total}}^{\text{trot}} = w_{vel} r_{vel} + w_{pose} r_{pose} + w_{zmp} r_{zmp} + w_{trot} r_{\text{trot}}. \tag{26}$$

Here, w_{vel}, w_{pose}, w_{zmp}, w_{walk}, and w_{trot} are the corresponding reward weights. These weights and the other sub-reward shape parameters are listed in Table 1.

Table 1. Reward function parameter settings.

Parameter	Meaning	Value
w_{vel}	Weight of velocity-tracking reward	2.5
w_{pose}	Weight of posture-control reward	1.5
w_{zmp}	Weight of ZMP-inspired support-margin reward	0.8
w_{walk}	Weight of walking-gait reward	1/0
w_{trot}	Weight of trotting-gait reward	0/1
α_v	Scaling coefficient for linear-velocity tracking error	4.0
α_ω	Scaling coefficient for angular-velocity tracking error	4.0
β_v	Internal weight of linear-velocity tracking	0.7
β_ω	Internal weight of angular-velocity tracking	0.3
α_p	Scaling coefficient for posture error	4.0
$\alpha_{z,3}$	ZMP-distance normalization coefficient under three-leg support	10.0
$\alpha_{z,4}$	ZMP-distance normalization coefficient under four-leg support	6.67

Table 1 summarizes the weight and shape parameters in the reward function. Velocity tracking, posture control, and support stability constitute the basic motion objectives. Walking and trotting rewards are activated mutually exclusively according to the task to guide the corresponding foot contact pattern. This design allows the reward function to constrain motion velocity, body posture, support stability, and gait structure simultaneously.

4. Experiments and Results Analysis

To validate the effectiveness of LA-PPO for quadruped locomotion control on mixed terrain, this section presents the experimental environment and simulation settings, mixed-terrain simulation results, ablation experiments and mechanism analysis, and real-world environment testing. The experimental section first describes the training and evaluation conditions, then reports the main performance results, analyzes the role of key modules, and finally tests the transfer performance of the policy on a physical platform.

4.1. Experimental Environment and Simulation Settings

All simulation experiments were conducted on the IsaacGym platform. IsaacGym is a GPU-accelerated physics simulator for robotics research and supports large-scale parallel environment sampling, which substantially improves reinforcement learning training efficiency. The hardware and software configuration used in this study is shown in Table 2. These hardware settings mainly improve parallel simulation and policy optimization efficiency, whereas the consistency between simulation and the real system is mainly determined by the robot dynamic model, contact parameters, terrain settings, and domain randomization configuration.

Table 2. Hardware and software configuration for simulation experiments.

Parameter	Value
GPU model	RTX 4090 × 2
GPU memory	24 GB × 2
CPU model	i9-13900K
System memory	128 GB
Storage	2 TB NVMe SSD
Operating system	Ubuntu 24.04
Framework	PyTorch 2.0.1
CUDA version	CUDA 12.1
Python version	Python 3.9

Simulation evaluation was conducted on mixed terrain. The terrain consists of flat regions, fixed-slope regions, and random rough-terrain regions. Both walking and trotting gaits were trained and evaluated on the same mixed terrain to examine basic locomotion ability, slope adaptation, and rough-terrain robustness simultaneously. Figure 4 shows the execution states of the two gaits in different terrain regions and is used to illustrate the simulation scenario and policy execution process.

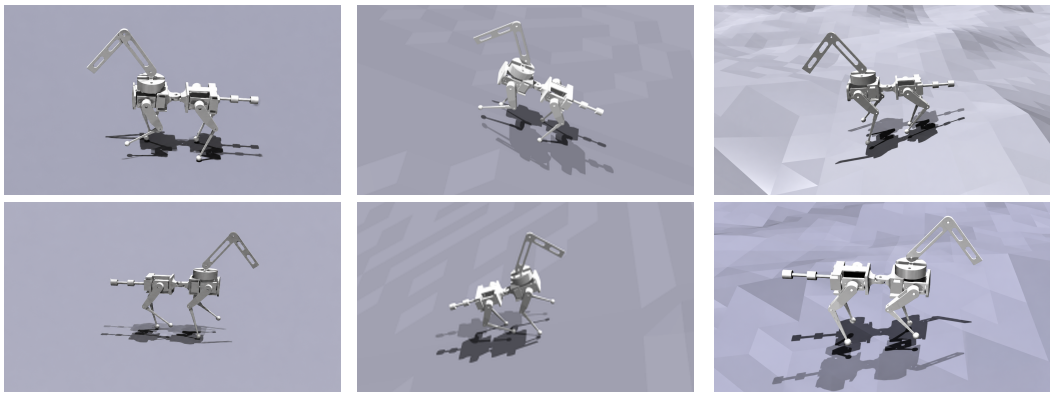


Figure 4. Simulation examples on mixed terrain. The **top row** shows, from left to right, walking on flat ground, sloped terrain, and random rough terrain. The **bottom row** shows, from left to right, trotting on flat ground, sloped terrain, and random rough terrain.

This study uses mean episodic reward and mean episode length (steps) as the main evaluation metrics. Mean episodic reward is the average cumulative reward over multiple evaluation episodes in each iteration and measures overall task completion quality. Mean episode length is the average number of sustained steps in the evaluation episodes and reflects policy stability and robustness. Under the same evaluation protocol, higher values of both metrics usually indicate better policy performance.

4.2. Mixed-Terrain Simulation Results

This section presents the structural ablation and algorithm comparison experiments for walking and trotting. In the structural ablation, LSTM-PPO (MHA removed) and MHA-PPO (LSTM removed) are used as variants of LA-PPO and are compared with PPO and LA-PPO to analyze the contribution of each module. These on-policy models share the same PPO optimization parameters, as shown in Table 3. In addition, SAC, TD3, and DDPG are used as off-policy baselines. To ensure consistent input information, PPO, TD3, DDPG, and SAC use an eight-step historical observation stack as a 1568-dimensional input vector, whereas LSTM-PPO, MHA-PPO, and LA-PPO preserve the eight-step history in temporal sequence form. All models are trained and tested under the same mixed terrain, observation and action definitions, and evaluation protocol, and key hyperparameters are tuned before the final comparison. The experimental results are shown in Figures 5 and 6; the horizontal axis is the number of iterations, the left subplot shows mean episodic reward, and the right subplot shows mean episode length. Table 4 further summarizes final performance and training stability under multiple random seeds.

Table 3. On-policy training-parameter settings.

Parameter	Value
Number of environments	1024
Number of iterations	1000
Episode length	25 s
Rollout length	48
Learning epochs per iteration	5
Number of mini-batches per iteration	4
Total environment steps	4.9152×10^7
Learning rate	1×10^{-3}
Learning rate schedule	auto
Discount factor γ	0.99
GAE parameter λ	0.95
Clipping parameter	0.2
Target KL	0.01
Entropy coefficient	0.01
Value loss coefficient	1.0
Maximum gradient norm	1.0

In the walking gait, LA-PPO obtains the highest final reward and maintains the longest and relatively stable episode length in the later training stage. MHA-PPO is higher than PPO in both final reward and episode length, indicating that attention-based reweighting helps select historical information relevant to the current control decision. LSTM-PPO has a higher final reward than PPO, but its final episode length is lower than that of PPO, suggesting that simply introducing recurrent encoding may improve reward optimization but does not necessarily improve sustained locomotion ability. Off-policy baselines such as SAC, TD3, and DDPG also improve during training, but their final rewards and mean episode lengths remain lower than those of LA-PPO.

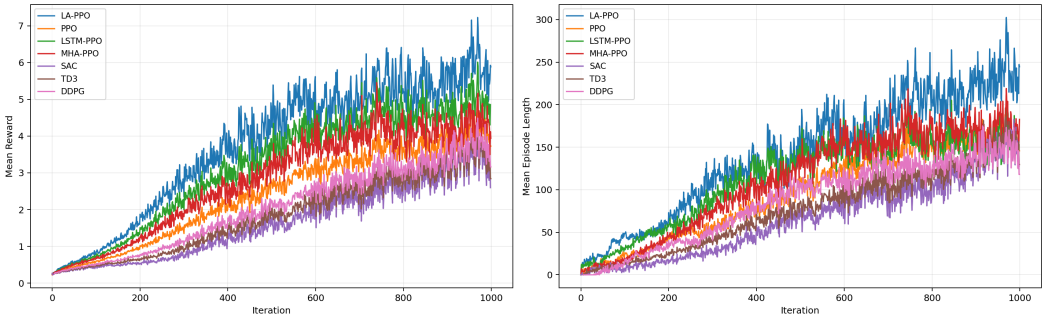


Figure 5. Model comparison on mixed terrain under the walking gait.

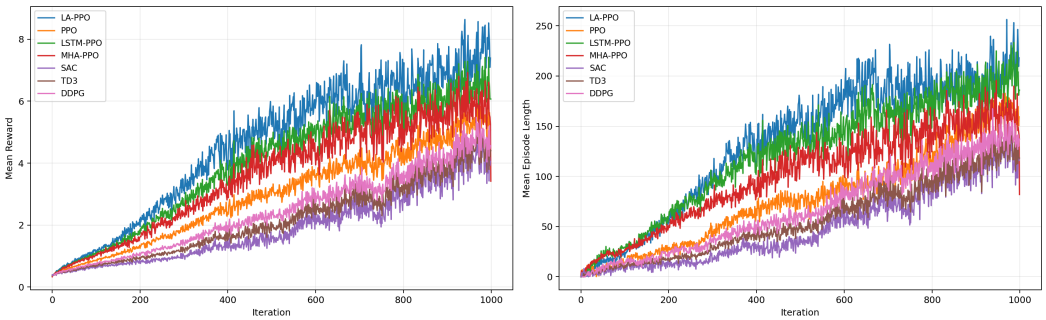


Figure 6. Model comparison on mixed terrain under the trotting gait.

In the trotting gait, LA-PPO still achieves the best overall performance. LSTM-PPO is higher than PPO in both final reward and episode length. MHA-PPO has a higher final reward than PPO but a slightly lower episode length, indicating that the benefits of a single module can vary with gait and evaluation metric. SAC, TD3, and DDPG improve more slowly and converge to relatively lower stable levels. Taken together, these results show that LA-PPO achieves higher final policy quality and better gait stability on mixed terrain containing flat, sloped regions, and random rough-terrain regions.

To further evaluate the stability of different algorithms under random initialization and sampling perturbation, we summarize results using the random seed as the statistical unit. Each comparison model is independently trained with five different random seeds. The final reward and final episode length are calculated as averages over the last 100 training iterations. The convergence iteration is defined as the iteration at which the mean reward first reaches 90% of the corresponding model's final stable reward and remains stable thereafter. Table 4 reports the seed-level statistics as mean $\pm$ standard deviation.

As shown in Table 4, in the walking gait, LA-PPO achieves a final reward of 5.798 and a final episode length of 232.040, both of which are the highest among all methods. Compared with PPO (4.075 and 162.575), the reward and episode length are improved by approximately 42.3% and 42.7%, respectively. In the trotting gait, the final reward and episode length of LA-PPO are 7.293 and 202.915, respectively, which are approximately 39.8% and 25.7% higher than those of PPO (5.218 and 161.404). The convergence iteration in the table is defined as the point at which each model reaches 90% of its own final stable reward. This definition is more suitable for describing the stabilization process of each model and should not be used alone as an absolute cross-model sample-efficiency metric. Under this definition, LA-PPO reaches the stable interval earlier than PPO and SAC in both gaits. Together with final reward and episode length, the results show that LA-PPO exhibits better overall training performance than the main baselines. Meanwhile, LSTM-PPO and MHA-PPO improve over PPO in some metrics, but the benefit of a single module is not completely consistent. The cascaded fusion of the two modules in LA-PPO achieves the best comprehensive performance on the main metrics of both gaits.

Table 4. Statistical results over multiple random seeds.

Gait	Model	Final Reward	Final Episode Length/Steps	Convergence Iteration
Walk	LA-PPO	5.798 ± 0.019	232.040 ± 0.481	503.0 ± 8.0
	LSTM-PPO	4.877 ± 0.015	157.540 ± 0.386	551.0 ± 12.0
	MHA-PPO	4.273 ± 0.013	172.471 ± 0.346	548.0 ± 11.0
	PPO	4.075 ± 0.011	162.575 ± 0.280	693.0 ± 8.0
	SAC	3.285 ± 0.009	145.539 ± 0.410	734.0 ± 10.0
	TD3	3.468 ± 0.010	149.799 ± 0.295	734.0 ± 9.0
	DDPG	3.675 ± 0.011	145.304 ± 0.302	734.0 ± 9.0
Trot	LA-PPO	7.293 ± 0.024	202.915 ± 0.436	602.0 ± 6.0
	LSTM-PPO	6.408 ± 0.020	192.981 ± 0.390	632.0 ± 10.0
	MHA-PPO	5.817 ± 0.018	159.384 ± 0.343	632.0 ± 9.0
	PPO	5.218 ± 0.013	161.404 ± 0.251	783.0 ± 7.0
	SAC	4.023 ± 0.011	116.868 ± 0.376	837.0 ± 8.0
	TD3	4.339 ± 0.012	123.388 ± 0.281	828.0 ± 9.0
	DDPG	4.623 ± 0.013	136.120 ± 0.288	827.0 ± 9.0

In addition to the main model comparison, we further analyze the performance sources of LA-PPO from two perspectives: reward design and attention weights. Reward component ablation evaluates the influence of key reward terms on training, while MHA heatmaps show how the policy selects information within the historical window.

4.2.1. Reward Component Ablation

Under the same mixed-terrain setting, we construct reward removal variants to evaluate the influence of key reward terms on training. Specifically, the ablation settings include removing the support stability reward, removing the walking gait reward for walking analysis, and removing the diagonal trot gait reward for trotting analysis.

The reward ablation experiments keep the network structure, training steps, and other hyperparameters unchanged and modify only the reward configuration. Training metrics are recorded at each iteration. Figure 7 shows the training curves of the complete reward and each removal variant. Because different reward configurations change the training objective and reward scale, the figure is mainly used to observe changes in training trends and policy behavior after removing a specific reward term, rather than to compare absolute reward values alone.

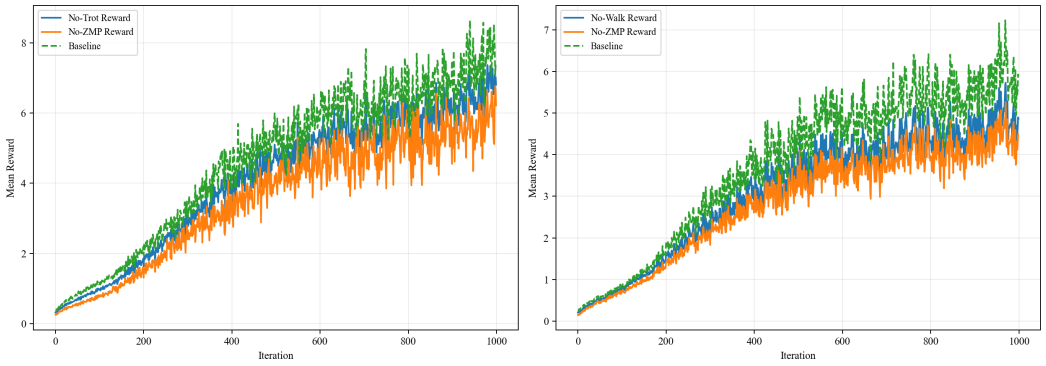


Figure 7. Reward component ablation for walking and trotting gaits on mixed terrain.

Figure 7 shows that the complete reward design exhibits a higher reward growth trend than the reward removal variants. After removing the support stability reward, the mean reward decreases in both gaits, suggesting that stability shaping based on the support polygon safety margin may help improve locomotion stability on slopes and rough regions.

After removing gait-specific rewards, the corresponding gait curves also decrease (no-walk reward in the walking analysis and no-trot reward in the trotting analysis), indicating that explicit gait priors positively contribute to forming the target foot contact pattern.

4.2.2. MHA Mechanism Analysis

To analyze how the attention module uses the historical window, we further compute the average attention distribution of each head over the eight-step historical observation window for the walking and trotting gaits.

To obtain the heatmaps shown in Figure 8, the trained LA-PPO policy is first set to evaluation mode, and the attention weights at each decision time are recorded while the robot runs on mixed terrain. For each sample, the four attention heads correspond to weight vectors over the eight-step historical window ($t-7$ to t). These weights are obtained by normalizing the attention scores using $\mathrm{softmax}(QK^\top/\sqrt{d_k})$. The walking and trotting segments are then averaged along the time dimension, producing an average attention matrix of "attention head $\times$ historical time step", which is visualized as a heatmap.

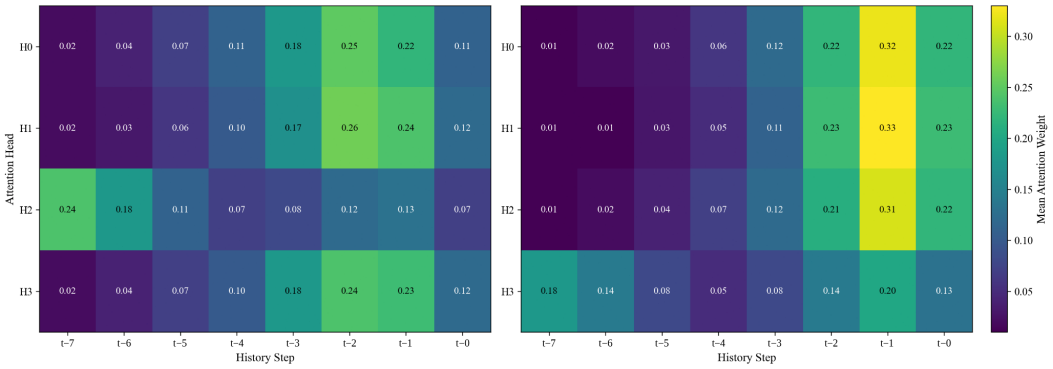


Figure 8. Multi-head attention heatmaps for walking and trotting gaits.

The heatmaps show that attention weights are not uniformly distributed over all historical steps. Some attention heads focus more on recent observations ($t-2$, $t-1$, and the current step t), which is consistent with the need for immediate state feedback in quadruped locomotion control. At the same time, other heads retain weights for earlier historical steps, indicating that the policy does not depend only on the current observation but combines contextual information over a longer time range. The attention distributions differ between walking and trotting, further indicating that MHA can form different historical information weighting patterns for different locomotion modes.

4.3. Real-World Environment Testing

To evaluate the practical transfer capability of LA-PPO, the trained policy is directly deployed on a physical quadruped robot platform and tested in real-world environments. During deployment, the observation order, action scaling, and joint limit constraints are kept consistent with training, and the policy output is clipped to the safety range used during training.

To improve robustness to model errors, contact uncertainty, and environmental variation, domain randomization is introduced during training. The detailed parameters are shown in Table 5. Each randomized variable is uniformly sampled within its preset range at the reset of each episode, thereby expanding the training distribution and improving the stability of sim-to-real transfer.

Table 5. Domain randomization parameters used during training.

Parameter	Randomization Range
Additional mass on `base_link`	$[-0.5, 1.0]$ kg
Initial position x	$[-0.5, 0.5]$ m
Initial position y	$[-0.5, 0.5]$ m
Initial yaw angle	$[-3.14, 3.14]$ rad
Base linear velocity noise	$[-0.1, 0.1]$
Base angular velocity noise	$[-0.2, 0.2]$
Gravity projection noise	$[-0.05, 0.05]$
Joint position noise	$[-0.01, 0.01]$
Joint velocity noise	$[-1.5, 1.5]$
Height scan noise	$[-0.1, 0.1]$

The real-world tests cover three types of terrain (flat ground, sloped terrain, and random rough terrain) and two gaits (walking and trotting). Each terrain–gait combination is tested five times, resulting in a total of 30 trials. Figure 9 shows representative robot postures during real-world testing. The six subfigures correspond to walking and trotting on the three terrain types. For ease of comparison, key frames are selected from trial videos during stable forward motion, and the camera angle and image scale are kept as consistent as possible.

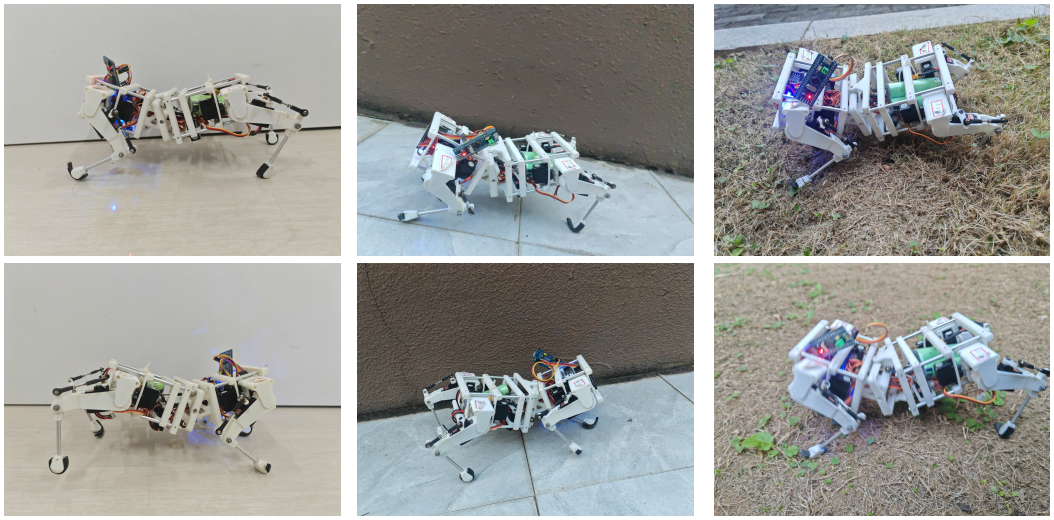


Figure 9. Real-world testing scenarios. The **top row** shows, from left to right, walking on flat ground, sloped terrain, and random rough terrain. The **bottom row** shows, from left to right, trotting on flat ground, sloped terrain, and random rough terrain.

In a single trial, the test is judged as failed if the robot falls, requires manual reset, or triggers an emergency stop. If the robot completes the motion within the prescribed test duration without triggering any of these events, the trial is judged as successful. Table 6 summarizes the number of successful trials for each terrain–gait combination. Duration and travel distance are calculated only over the subset of successful trials and are reported as mean $\pm$ standard deviation; failed trials are counted only in the success count and are not included in the mean duration or distance.

Table 6 shows that LA-PPO achieves 5/5 success in both flat-ground walking and trotting, with successful trials lasting approximately 30 s, indicating stable execution on low-disturbance terrain. On slopes, both gaits achieve 4/5 successful trials. On random rough terrain, walking achieves 4/5 success and trotting achieves 3/5 success. The variance in duration and travel distance further increases for successful trials, reflecting more pro-

nounced performance fluctuation under complex contact conditions. It should be noted that different command speeds are used for different terrains, so differences in travel distance cannot be attributed solely to terrain difficulty. Overall, increased terrain complexity leads to lower success counts and larger trajectory fluctuations. Nevertheless, the learned policy can still complete locomotion tasks on all three real terrains, demonstrating preliminary practical transfer capability.

Table 6. Quantitative results of real-world environment testing.

Terrain	Gait	Command Speed/(m/s)	Duration/s	Travel Distance/m	Success Count
Flat ground	Walk	0.45	30.0 ± 0.8	13.2 ± 0.4	5/5
Flat ground	Trot	0.85	30.0 ± 0.7	24.8 ± 0.6	5/5
Slope	Walk	0.40	30.0 ± 1.2	11.6 ± 0.7	4/5
Slope	Trot	0.75	30.0 ± 1.5	20.5 ± 1.0	4/5
Rough terrain	Walk	0.35	30.0 ± 2.0	9.8 ± 1.1	4/5
Rough terrain	Trot	0.65	30.0 ± 2.4	16.7 ± 1.5	3/5

Note: Duration and travel distance are computed only over successful trials; failed trials are included only in the success count.

5. Conclusions

This paper proposes LA-PPO, an improved PPO algorithm integrating LSTM and MHA for quadruped robot locomotion control on mixed terrain. The method uses LSTM to model temporal dependencies in historical observations and uses MHA to adaptively weight critical information within the historical window, thereby enhancing the policy's ability to use historical information in complex terrains.

Simulation experiments show that LA-PPO achieves the highest final reward and the longest mean episode length on mixed terrain containing flat, sloped regions, and random rough-terrain regions. Statistics over multiple random seeds further show that the method maintains a stable advantage in final performance. Reward component ablation and attention heatmap analysis provide auxiliary explanations from the perspectives of training objective design and historical information utilization.

Real-robot tests show that the trained policy can perform walking and trotting on flat ground, sloped terrain, and random rough terrain, providing preliminary quantitative support for sim-to-real transfer. Future work will further expand the scale of real-world testing, introduce more terrain types and real-robot baseline comparisons, and combine visual, tactile, and other multimodal perception to improve adaptation to complex environments.

Author Contributions: E.Y. designed the LA-PPO framework, implemented the core code, conducted the main experiments, and drafted the manuscript. S.W. designed and optimized the state/action spaces and reward functions and contributed to experimental planning and manuscript revision. Z.T. conducted simulation and real-world experiments and was responsible for data organization, validation, and figure preparation. L.W. contributed to code debugging, hyperparameter tuning, comparative experiments, result statistics, and manuscript revision. X.T. conceived and supervised the study, provided critical academic revision, and handled submission and correspondence. E.Y., S.W., Z.T. and L.W. contributed equally to this work and share first authorship. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical approval and informed consent were not required for this study.

Data Availability Statement: The data supporting the findings of this study are available in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Raibert, M.; Blankespoor, K.; Nelson, G.; Playter, R. BigDog, the rough-terrain quadruped robot. *IFAC Proc. Vol.* **2008**, *41*, 10822–10825. [CrossRef]
2. Hutter, M.; Gehring, C.; Jud, D.; Lauber, A.; Bellicoso, C.D.; Tsounis, V.; Hwangbo, J.; Bodie, K.; Fankhauser, P.; Bloesch, M.; et al. ANYmal—A highly mobile and dynamic quadrupedal robot. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 38–44.
3. Lee, J.; Hwangbo, J.; Wellhausen, L.; Koltun, V.; Hutter, M. Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.* **2020**, *5*, eabc5986. [CrossRef] [PubMed]
4. Di Carlo, J.; Wensing, P.M.; Katz, B.; Bledt, G.; Kim, S. Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–9.
5. Vukobratović, M.; Borovac, B. Zero-moment point—Thirty five years of its life. *Int. J. Humanoid Robot.* **2004**, *1*, 157–173. [CrossRef]
6. Kajita, S.; Kanehiro, F.; Kaneko, K.; Fujiwara, K.; Harada, K.; Yokoi, K.; Hirukawa, H. Biped walking pattern generation by using preview control of zero-moment point. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003; Volume 2, pp. 1620–1626.
7. Miki, T.; Lee, J.; Hwangbo, J.; Wellhausen, L.; Koltun, V.; Hutter, M. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.* **2022**, *7*, eabk2822. [CrossRef] [PubMed]
8. Kumar, A.; Fu, Z.; Pathak, D.; Malik, J. RMA: Rapid motor adaptation for legged robots. In Proceedings of the Robotics: Science and Systems XVII, Virtually, 12–16 July 2021. [CrossRef]
9. Hwangbo, J.; Lee, J.; Dosovitskiy, A.; Bellicoso, D.; Tsounis, V.; Koltun, V.; Hutter, M. Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* **2019**, *4*, eaau5872. [CrossRef] [PubMed]
10. Hoeller, D.; Rudin, N.; Sako, D.; Hutter, M. ANYmal parkour: Learning agile navigation for quadrupedal robots. *Sci. Robot.* **2024**, *9*, eadi7566. [CrossRef] [PubMed]
11. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
12. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1587–1596.
13. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the International Conference on Learning Representations, San Juan, PR, USA, 2–4 May 2016.
14. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347. [CrossRef]
15. Rudin, N.; Hoeller, D.; Reist, P.; Hutter, M. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning (CoRL), London, UK, 8–11 November 2021*; PMLR: Cambridge, MA, USA, 2022; Volume 164, pp. 91–100.
16. Choi, S.; Ji, G.; Park, J.; Kim, H.; Mun, J.; Lee, J.H.; Hwangbo, J. Learning quadrupedal locomotion on deformable terrain. *Sci. Robot.* **2023**, *8*, eade2256. [CrossRef] [PubMed]
17. Siekmann, J.; Valluri, S.; Dao, J.; Bermillo, L.; Duan, H.; Fern, A.; Hurst, J. Learning memory-based control for human-scale bipedal locomotion. In Proceedings of the Robotics: Science and Systems XVI, Virtually, 12–16 July 2020. [CrossRef]
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
19. Lai, H.; Zhang, W.; He, X.; Yu, C.; Tian, Z.; Yu, Y.; Wang, J. Sim-to-real transfer for quadrupedal locomotion via terrain transformer. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 5141–5147.
20. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1889–1897.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.