# Reinforcement Learning for Elegant Walking in Quadruped Robots

Trung Dang
*University of Massachusetts Amherst*
Amherst, USA
tmdang@umass.edu

Anh Tran
*University of Massachusetts Amherst*
Amherst, USA
anhtran@umass.edu

***Abstract*—Recent advancements in locomotion control have significantly improved the agility and stability of robots. Among these developments, Reinforcement Learning (RL) has emerged as a promising approach, enabling robots—particularly quadruped robots—to tackle remarkably challenging tasks, such as parkour and navigating complex, unstructured terrains. However, RL-based controllers often exhibit challenges, such as generating asymmetric or unpredictable gaits. Additionally, current RL policies frequently adopt a one-size-fits-all approach for all terrains, resulting in excessive noise and unnatural movements on structured and flat surfaces. In this project, we propose a training pipeline to achieve stable and refined walking patterns in quadruped robots. We define *elegance* as replicating the walking patterns of animals, characterized by long, low strides and consistent body height. Our approach involves refining training environment parameters and terrain design to promote natural, symmetric gaits across terrain. Furthermore, we explore using a depth camera to enhance the robot's ability to climb stairs and navigate rough terrains. The supplementary videos are available at `dmtrung14.github.io/cs690k-robot-dog-catwalk`**

***Index Terms*—Reinforcement Learning, Quadruped Robots, Locomotion Control, Unitree Go1**

## I. Introduction

### A. Motivation

Locomotion control has been a long-standing challenge in robotics. Planning the motion of end-effectors along a predefined trajectory often involves solving complex, high-degree inverse kinematics equations, which typically admit infinitely many solutions. As a result, motion planning is generally approached using one of two methods: optimization-based controllers or learning-based controllers. Optimization techniques, such as the Newton-Raphson method or gradient descent, reformulate the problem into computationally tractable linear or quadratic programming problems, enabling high-frequency solutions. However, these methods lack exteroceptive data, limiting their ability to navigate challenging obstacle courses or perform teleoperation tasks in extreme conditions. Incorporating sensor data into optimization-based controllers could expand the observation space, but this often exceeds the computational capabilities of onboard devices, which are frequently as compact as a Raspberry Pi. In contrast, RL-based controllers offer a lower computational burden, as data only need to pass through the neural network once. This characteristic allows seamless integration of sensory inputs, facilitating locomotion in diverse environments. Despite these advantages, RL-based controllers often require extensive reward shaping and struggle to generalize across terrains. For instance, a robot trained to climb stairs may lift its feet excessively on flat terrain, resulting in an unnatural gait, whereas one optimized for smooth walking on flat ground may fail to clear obstacles like stairs. Without sufficient reward shaping, robots may also converge to unexpected behaviors, such as pronking on flat surfaces. This raises a natural question: *Can we develop an RL policy that achieves both agility and elegance with minimal reward shaping?*

### B. Related Works

*1) Optimization-based controllers:* Recent advancements in MPC and optimization techniques have enhanced model-based controllers, enabling them to support a broader range of gaits [1], [2] and perform challenging maneuvers such as jumping and backflipping [3], [4]. However, these models remain incapable of navigating uneven terrains without exteroceptive data. Frankhauser et al. [5] propose an efficient framework for generating point cloud and elevation maps using depth cameras, but it relies on accurate knowledge of the robot's current body pose, typically obtained through contact sensors. This requirement creates challenges for lower-cost quadrupeds.

*2) RL-based controllers:* Extensive prior research has established robust frameworks for quadruped robot locomotion [6], [7]. Nevertheless, early reinforcement learning policies struggled in more complex environments, often failing due to obstacles or losing traction. Recent studies [8], [9] have achieved significant progress in motion agility, enabling robots to navigate challenging terrains such as stepping stones or perform advanced locomotion tasks like landing on inclined platforms. However, these policies are optimized for extreme objectives, often resulting in noisy, forceful steps even on flat surfaces. Siekmann et al. [10] utilize the Von Mises distribution to define a trajectory for the robot, reinforcing it through a smoothness reward during training. Although this method enables quick convergence to effective walking gaits on flat terrain, it restricts exploration in more challenging terrains, ultimately making it unsuitable for navigating more demanding tasks.

In this project, we leverage terrain shaping as an alternative to reward shaping. Instead of directly rewarding or penalizing actions, terrain shaping introduces sparse, intermittent obsta-

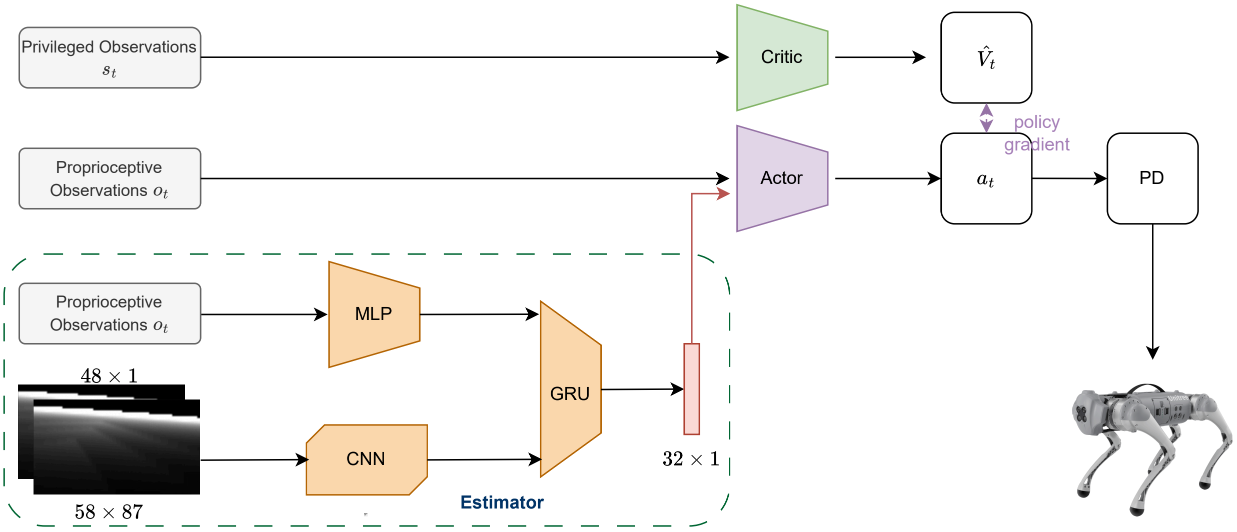
Privileged Observations $s_t$
Proprioceptive Observations $o_t$
Proprioceptive Observations $o_t$
48 × 1
58 × 87
MLP
CNN
GRU
32 × 1
Estimator
Critic
Actor
$\hat{V}_t$
policy gradient
$a_t$
PD

Fig. 1. Architecture of the proposed reinforcement learning framework for quadruped locomotion. Privileged and proprioceptive observations are processed through an actor-critic network. Proprioceptive data is further refined using MLP, CNN, and GRU modules, and actions are executed via a PD controller guided by policy gradient updates.

cles to enforce desirable behaviors, such as appropriate gait and foot height, without explicit reward functions. This method reduces the risk of overfitting to specific terrains, fostering robust and generalizable policies. Additionally, we design the training environment in IsaacLab [11] to promote symmetry in robot observations and employ empirical normalization techniques to incentivize symmetrical policies. Experimental evaluations on both simulated and real robots demonstrate marked improvements in locomotion quietness, robustness, and symmetry compared to existing methods.

## II. Method

We use IsaacLab as our training environment. This allows us to circumvent the two-stage training paradigm prevalent in prior learning-based policies with camera integration. The depth image is preprocessed through a combination of convolutional and fully connected layers down to a $(32,)$ latent vector and concatenated with proprioception data to form an end-to-end training pipeline. In this section, we provide a detailed overview of the neural network and other design choices.

### A. Overview

We leverage the framework from [12], which consists of an actor, critic, and estimator that takes both proprioceptive and exteroceptive data from the depth image to predict the pose of the robot $\hat{o}_t$ and the elevation map $\hat{m}_t$. The proprioceptive observation $o_t$ is a 48-dimensional vector directly measured from joint encoders and the IMU, such that:

$$o_t = \begin{bmatrix} v_t & \omega_t & g_t & c_t & \theta_t & \dot{\theta}_t & a_{t-1} \end{bmatrix}^T \tag{1}$$

where $v_t, \omega_t$ are the linear and angular velocities, $g_t$ is the gravitational vector of the body frame, $c_t$ is the command, $\theta_t, \dot{\theta}_t,$ and $a_{t-1}$ are joint position, joint velocity, and previous action. The action space is a 12-dimensional vector corresponding to the 12 joints of the quadruped robot. If the depth image module is used, a latent vector of dimension $(32,)$ will be appended, raising the total dimension of the observation to $(80,)$. For the value network, or the critics' observation, an elevation map $m_t$ of dimension 187 is included, raising the total size of $s_t$ to $(235,)$.

TABLE I
Reward Function Elements

| **Reward** | **Equation** $(r_i)$ | **Weight** $(w_i)$ |
|---|---|---|
| Lin. velocity tracking | $\exp\{-4(v_{xy}^{cmd} - v_{xy})^2\}$ | 4.5 |
| Ang. velocity tracking | $\exp\{-4(\omega_{yaw}^{cmd} - \omega_{yaw})^2\}$ | 1.5 |
| Linear velocity (z) | $\frac{v_z^2}{2}$ | -0.5 |
| Angular velocity (xy) | $\frac{\omega_{xy}^2}{2}$ | -0.05 |
| Orientation | $\frac{\lvert g\rvert^2}{2}$ | -1.0 |
| Joint accelerations | $\ddot{\theta}^2$ | $-2.5 \times 10^{-7}$ |
| Joint power | $\lvert\tau\rvert\lvert\dot{\theta}\rvert$ | $-2 \times 10^{-5}$ |
| Collision | $-n_{collision}$ | -10.0 |
| Action rate | $(a_t - a_{t-1})^2$ | -0.01 |
| Smoothness | $(a_t - 2a_{t-1} + a_{t-2})^2$ | -0.01 |
| Base Height | $(h_t - 31)^2$ | -1.0 |

$$o_t^{\text{depth}} = \begin{bmatrix} o_t & d \end{bmatrix}^T \quad s_t = \begin{bmatrix} o_t & m_t \end{bmatrix}^T \tag{2}$$

### B. Training details

*1) Simulation Platform:* We set up 3000 parallel environments on L40S GPU and leveraged IsaacLab to train 10,000 iterations in approximately 20 hours. Nonetheless, empirical evidence shows that our model consistently converges after between 5,000 and 7,000 iterations.

*2) Reward Function:* The reward functions and parameters are specified in Table I. For the blind policy, we adopted the weights from [6], [7], [12]. However, this will lead to extremely negative rewards in vision-based policy due to the increased complexity of the observation space. Therefore, to promote actions rather than early termination, we incremented linear velocity and angular velocity rewards threefold. To maintain consistent body height, we constrained base height to 31cm. We avoid using other reward functions as in [8] to emphasize the robustness of our design.

*3) Design of Environment:* The scene is arranged in a $m \times n$ grid, depending on the number of terrains and the granularity of levels specified. On the $x-$basis, the terrain varies by type, on the $y-$basis, the terrain increases in difficulty. For flat terrain training, the scene is a large $n \times n$ flat surface. Since we did not use gait shaping reward functions such as Raibert heuristic [13] or periodic reward functions [10], a purely flat terrain will converge to any gait that can launch the robot in the desired direction, such as pronking and cantering.

We introduce sparse obstacles to trip the robot should they ever optimize to such gaits. For the blind policy, we make the terrain extremely rough, with randomly sampled spikes in the range of $[4, 20]$ centimeters. For the vision-based policy, randomly rough terrain would counteract since the policy will partially optimize to the visual data, which contains terrain information $0.3$ to $3$ meters ahead of the robot. Therefore, we implemented more predictable terrains such as horizontal rails and waves. The benefits of implementing such terrains are three-fold: (1) it trips the robots if they ever converge to an unnatural gait, (2) it teaches the robot the desired feet raise height, beyond which no improvement in rewards can be achieved, and (3) it teaches the robot the desired orientation in sloped terrains.

On rough terrains, the visual observation is generally symmetric in the $y-axis$, meaning if the robot sees stairs in the positive $y$ direction, it will likely see stairs in the negative $y$ direction, and similarly for other terrains. However, the visual observation is not symmetric in the $x-$axis, with stairs in the negative $x$ direction and flatter terrains such as discrete obstacles and rails in the positive $x$ direction. This again leads to asymmetric gaits, such as galloping on the left foot or not using the left rear leg. To counter this phenomenon, we made the environment symmetric in $x$ direction by appending flatter terrains to the negative side of stairs, thereby enforcing a symmetric policy. The full description of the subterrains is listed in Table II.

TABLE II
SUB-TERRAIN CONFIGURATION FOR UNITREE GO1 ROUGH TERRAINS

| Name | Proportion | Value Range |
|---|---|---|
| HF Pyramid Slope (Left) | 0.1 | $(0.0, 0.4)$ |
| HF Pyramid Slope (Left) | 0.1 | $(0.0, 0.4)$ |
| Horizontal Rails (Left) | 0.2 | $(0.04, 0.07)$ |
| Random Rough (Left) | 0.2 | $(0.02, 0.06)$ |
| Pyramid Stairs (Left) | 0.2 | $(0.05, 0.18)$ |
| Pyramid Stairs | 0.2 | $(0.05, 0.18)$ |
| Pyramid Stairs Right) | 0.2 | $(0.05, 0.18)$ |
| Random Rough (Right) | 0.2 | $(0.02, 0.06)$ |
| Horizontal Rails (Right) | 0.2 | $(0.04, 0.07)$ |
| HF Pyramid Slope (Right) | 0.1 | $(0.0, 0.4)$ |
| HF Pyramid Slope (Inverted Right) | 0.1 | $(0.0, 0.4)$ |

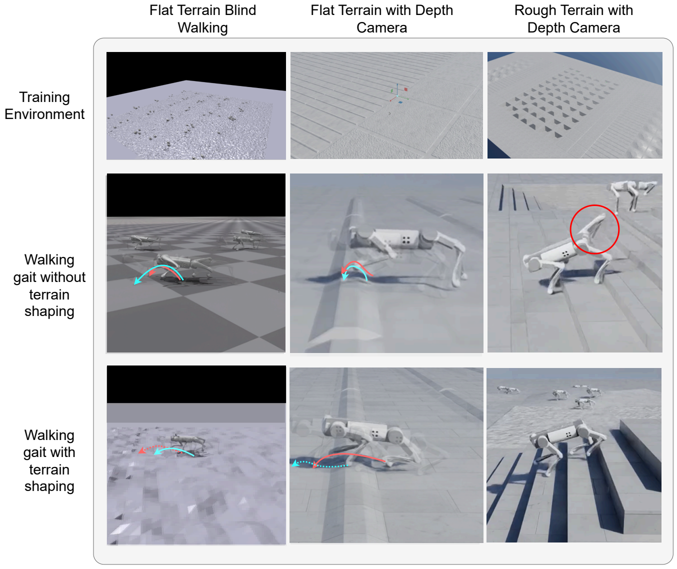
Flat Terrain Blind Walking
Flat Terrain with Depth Camera
Rough Terrain with Depth Camera
Training Environment
Walking gait without terrain shaping
Walking gait with terrain shaping

Fig. 2. The terrain design and snapshots of the robot walking in each of the three training environments. The blue arrows represent the trajectory of the front-left foot, and the red arrows represent the trajectory of the front-right foot. Dotted arrows are the projected trajectory, which is not yet covered at the time of the snapshot.

*4) Empirical Normalization:* Finally, we remove any remaining possible asymmetry by normalizing the observation. The observation $x$ is normalized by:

$$x_{\text{norm}} = \frac{x - \mu}{\sigma + \epsilon} \tag{3}$$

Here $\epsilon = 10^{-2}$ is the noise injected to the standard deviation $\sigma$ of the observations and $\mu$ is the mean of the observations currently in the rollout storage.

## III. RESULTS

### A. Summary

We evaluated our RL-based locomotion pipeline on flat and rough terrains, both with and without the use of depth cameras, to assess its performance in achieving stable, symmetric, and robust walking patterns.

*1) Flat terrain:* The RL policy demonstrated stability and symmetry during flat terrain tests. The robot maintained a consistent body height and exhibited smooth walking patterns characterized by low, natural strides. In both the blind and vision-based training framework, the robot significantly improved in coordinating footsteps, raising only one foot at a time, and achieving a fully symmetric walking gait. Additionally, the policy also converges to longer and flatter trajectories.

*2) Rough Terrain:* On rough terrains, the robot effectively navigated obstacles such as horizontal rails, random rough surfaces, and sloped terrains. By designing a symmetric training environment and using empirical normalization, we alleviated the asymmetry in the resulting gait and avoided convergence to suboptimal policies such as walking with 3 legs or leaping on one side of the robot.

### *B. Comparison with Baseline*

Compared to the default configurations which are provided in IsaacLab or previously implemented [6], [7], [12], we achieved:

- **Fewer Failure Modes**: The robot avoided unnatural gaits (e.g., pronking or excessive leg lifting) often observed in policies without terrain shaping.
- **Improved Symmetry and Stability**: Training with symmetric terrains reduced asymmetric gaits, enhancing locomotion robustness across diverse terrains.
- **Enhanced Adaptability**: The vision-based policy showed improved obstacle navigation compared to baseline proprioception-only approaches.

## IV. Conclusion

In this project, we addressed the dual challenges of achieving agility and elegance in quadruped robot locomotion through an RL-based training pipeline that minimizes reliance on reward shaping. By leveraging *terrain shaping*, we introduced sparse and intermittent environmental obstacles, enabling the robot to learn natural walking patterns without explicit reward engineering. This approach reduced the risk of overfitting and fostered robust, generalizable policies. Our experimental results demonstrated significant improvements in gait symmetry, stability, and adaptability compared to traditional RL methods. The combination of terrain shaping, curricular training environments, and empirical normalization allowed the robot to achieve smooth and elegant locomotion on flat terrains while maintaining the versatility to navigate challenging environments with obstacles. This work underscores the potential of terrain-based learning strategies for advancing robotic locomotion. Future research could extend these methods by integrating multi-sensory inputs or addressing real-world dynamics, further bridging the gap between simulation and practical deployment.

## References

[1] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, “Feedback mpc for torque-controlled legged robots,” *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4730–4737, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:155092463

[2] D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, “Dynamic locomotion and whole-body control for quadrupedal robots,” *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3359–3365, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:6381062

[3] Y. Yang, G. Shi, X. Meng, W. Yu, T. Zhang, J. Tan, and B. Boots, “Cajun: Continuous adaptive jumping using a learned centroidal controller,” in *Conference on Robot Learning*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259187920

[4] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, “Dynamic locomotion in the mit cheetah 3 through convex model-predictive control,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.

[5] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, “Robot-centric elevation mapping with uncertainty estimates,” in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.

[6] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, “Learning to walk in minutes using massively parallel deep reinforcement learning,” *ArXiv*, vol. abs/2109.11978, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237635100

[7] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning quadrupedal locomotion over challenging terrain,” *Science Robotics*, vol. 5, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:224828219

[8] G. B. Margolis and P. Agrawal, “Walk these ways: Tuning robot control for generalization with multiplicity of behavior,” *Conference on Robot Learning*, 2022.

[9] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, “Extreme parkour with legged robots,” *arXiv preprint arXiv:2309.14341*, 2023.

[10] J. Siekmann, Y. Godse, A. Fern, and J. W. Hurst, “Sim-to-real learning of all common bipedal gaits via periodic reward composition,” *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7309–7315, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:226237257

[11] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, “Orbit: A unified simulation framework for interactive robot learning environments,” *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[12] S. Luo, S. Li, R. Yu, W. Zhicheng, J. Wu, and Q. Zhu, “Pie: Parkour with implicit-explicit learning framework for legged robots,” 08 2024.

[13] M. Raibert, B. B, M. Chepponis, J. Koechling, and J. Hodgins, “Dynamically stable legged locomotion,” p. 207, 09 1989.