# ABSTRACT

This report presents a novel approach to quadruped locomotion control across diverse terrains, integrating reinforcement learning (RL) techniques with proprioceptive observations. While existing literature focuses on enabling quadrupeds to follow various gait patterns or employing trot gaits for challenging landscapes, little attention has been given to controllers capable of demonstrating different gaits across varied terrain types. Our study introduces an RL-based methodology for controlling quadruped locomotion over a range of terrains, leveraging multiple gaits including trotting, hopping, bounding, and pacing. We utilize RL policies to facilitate the emergence of gaits adept at traversing uneven landscapes, while enforcing diverse behaviors such as gait selection, body height adjustments, and step height modulation.

Importantly, we leverage the asymmetric actor-critic framework wherein the actor receives partial state information (POMDP), while the critic has access to the full state, including privileged information. This setup enhances the adaptability and robustness of the learning process by simulating real-world partial observability scenarios. We propose an asymmetric reward architecture wherein robots navigating uneven terrain receive lesser coefficients of negative auxiliary rewards compared to those on flat surfaces. This adaptation, based on the Isaac Gym environment, optimizes locomotion strategies by balancing risk and performance across different terrains.

Additionally, we integrate Control Barrier Function-based rewards to imbue our controller with less aggressive and more energy-efficient locomotion. This incorporation enhances the adaptability and safety of our system, enabling the quadruped to navigate complex environments while conserving energy resources. By demonstrating the efficacy of our approach through simulations and real-world experiments, we illustrate how our RL-based controller seamlessly adapts to varying terrain conditions, offering a promising avenue for the development of agile and efficient robotic locomotion systems.

# LIST OF FIGURES

# LIST OF TABLES

# TABLE OF CONTENTS

# 1. INTRODUCTION

Reinforcement learning (RL) is a framework for developing controller by interacting with a environment and continuously improving the controller based upon the feedback received in the form of rewards. Owing to the development of RL algorithms capable of handling continuous action space [1], [2], RL has been demonstrated by several studies [3], [4], [5] as a powerful alternative or replacement for model-based methods of designing a controller for quadruped robots. Often, even with state-of-the-art RL algorithms, training controller for a particular robot requires several million interactions in the simulation. To that end, an impetus has been provided to the quadruped locomotion research by Isaac Gym [3] that allows training several agents in parallel using proprioceptive inputs within a short span of time.

Several works [3], [6], [4], [7], [8] have shown amazing results in the direction of quadruped walking by utilizing position control based on proprioceptive observations, including but not limited to joint angles, joint angular velocity, robot orientation, etc. Although, proprioception-based controller is suitable flat terrain walking but will struggle when faced with terrains including uneven surfaces, slopes or slippery surfaces. A common approach taken by works such [4], [5], [7] involves making use of adaptation module to estimate height map around the quadruped or ground friction by taking as input of history of proprioception states. [5] has even shown impressive results on stair without any vision-based input. This speaks in favour of encouraging development of robust proprioception based controllers that can in future assist vision-based quadruped controller even in the event of failure of vision modality.

A robust quadruped controller might require different types of gait such pacing, bounding or pronking if it has to traverse different terrain. One of the seminal works that focus on developing a RL-based controller capable to demonstrating different gait behaviors is [8]. [8] uses Raibert heuristic [9] to learn variety of gaits. Their controller is trained for several gait parameters including step height, body height, step frequency, etc.. The multiplicity of behaviour allows traversal on different terrain but requires a human intervention to determine the gait parameters. [5] is able to traverse different terrains such as stairs without any help from human by automatically changing step height. Nevertheless, [5] is capable of moving using only trot gait. There are very few works that focus on traversing different terrains using different gait behavior[10].

In this work we focus on developing a controller using RL algorithm that is capable of walking on different terrains using different type of gait behaviour by using entirely proprioception. We leverage asymmetric actor-critic [11] architecture to eliminate the training of adaptation module for locomotion on diverse terrain. At the same time, we propose the use of asymmetric reward functions for acquiring good behaviour both of flat and non-flat terrains. Further, we try to enforce different gait behaviors using Raibert heuristic. Moreover, we also incorporate control barrier function based rewards to make the controller less aggressive and more energy efficient[12].

## *1.1. Preliminaries*

### A. Reinforcement Learning

Usually, a task in reinforcement learning (RL) is described using the framework of Markov Decision Process (MDP). An MDP M is defined using the tuple $M = \{S,A,R,P,\gamma\}$. Here, S denotes the set of all states an RL agent can witness and A denotes the set of all actions available to the agent. Moreover, $R : S \times A \times S \rightarrow R$ is the reward function and $P : S \times A \rightarrow B(S)$ is the transition probability kernel, where B(S) is the Borel σ-algebra over the set S. Given an MDP, our objective (η(π)) is to find a policy (π) to maximize long term discounted summation of rewards $(\eta(\pi) = {}^{P}{}_{t}(\gamma^{t}r(st, at, st+1))$ . In the case of continuous state and action space, actor-critic algorithms [1], [2] based on approximate policy iteration scheme are used to obtain near-optimal policies.

### B. Asymmetric Actor-Critic

Actor-Critic algorithms majorly contain two components: Actor and Critic. Actor is function of states and provides actions depending on states while critic component contains information about long term discounted reward achievable from a particular state while following a policy. Usually, the states passed to an actor and critic are same. However, it has been shown previously in [11] that in case the environment is partially observable, providing full states to the critic and partial states to the actor provides performance benefits as compared to providing partial observation to both the components. Therefore, in this work we will use asymmetric version of proximal policy optimization (PPO) [1] algorithm with rewards shaped using Raibert heuristic and barrier function .

### C. Reward Shaping

Reward shaping is a critical component of developing controller for quadruped. Raibert heuristic ([9], [8]) is used to design reward function to allow the robot to change various parameters of gait such as step height, step frequency, stance width, etc. In the absence of Raibert heuristic, enforcing a particular gait parameter such as stance width becomes difficult. Moreover, barrier function based reward shaping is used to make the quadruped controller energy efficient and less aggressive behavior[12]. Barrier functions can be constructed to make sure the states of the system always satisfies a particular constraint. If a function h satisfies 2 and 1, where κ is an increasing continuous function, δC denotes boundary of a set C and Int(C) denotes interior of the set C, then it can be guaranteed that the set C of states satisfying desirable constraints is a forward invariant set. Consequently, to satisfy 2 the reward function can be constructed as follows: $r'(s, \dot{s}, a) = r(s, a) + r^{BF}(s, (\dot{s}))$, where $r^{BF}(s, (\dot{s})) = \dot{h}(s, \dot{s}) + \lambda(s)$ and $\lambda \in R_{>0}$.

$$
\begin{aligned}
\mathcal{C} &= s \in \mathcal{S} : h(s) \geq 0 \\
\delta\mathcal{C} &= s \in \mathcal{S} : h(s) = 0 \\
\mathrm{Int}(\mathcal{C}) &= s \in \mathcal{S} : h(s) > 0
\end{aligned} \tag{1}
$$

$$
\dot{h}(s, \dot{s}) + \kappa(h(s)) \geq 0 \tag{2}
$$

The structure of the paper is as follows: In section II. we throw light of the previous works. Further, in section III, we establish the objectives of this work. In section IV, we talk about our specific details of our method and the structure of the controller. Next, we discuss the results obtained from out method in section V followed by conclusion of our work in section VI.

# 2. LITERATURE REVIEW

Quadrupedal locomotion, particularly in challenging terrains, has garnered significant interest in the robotics community due to its applications in search and rescue missions, inspections, exploration, and disaster response. Various approaches have been explored to develop robust locomotion policies for quadruped robots, with deep reinforcement learning (DRL) emerging as a promising technique. In this section, we review recent advancements in this field, focusing on methodologies, limitations, and opportunities for improvement.

## *2.1. Deep Reinforcement Learning for Quadruped Locomotion:*

Recent research has witnessed the application of DRL algorithms to train locomotion policies for quadruped robots. "***DreamWaQ"***, proposed by [authors] et. al., introduces a framework for learning robust quadrupedal locomotion on uneven terrains such as slopes, stairs, uneven ground, solely from proprioceptive inputs using a DRL algorithm. Notably, the framework utilizes an asymmetric actor-critic architecture to implicitly imagine terrain properties, resulting in the emergence of locomotion behaviors, primarily trot gait, driven by the minimization of energy consumption.

Other works like "***Walk These Ways"***, proposed by [8] et. al., contribute to the development of low-level quadruped controllers capable of executing diverse structured behaviors, including various gaits and movements. This approach emphasizes the utility of interpretable high-level control interfaces, facilitating the collection of quadruped demonstrations for diverse tasks. Moreover, the incorporation of ***Multiplicity of Behavior (MoB)*** techniques enables the learning of a single policy encoding a structured family of locomotion strategies, allowing rapid adaptation to diverse environments without the need for extensive retraining. The *"Walk These Ways"* framework showcases MoB as a practical tool for out-of-distribution generalization, offering diverse locomotion strategies such as crouching, hopping, high-speed running, stair traversal, and rhythmic dance. By learning multiple methods of achieving goals, MoB facilitates generalization across different tasks and environments, thereby bypassing the iterative cycle of reward and environment redesign typically required for out-of-distribution scenarios.

In this section, we delve into a detailed examination of the DreamWaQ, Walk These Ways methodologies and “Minimizing energy consumption leads to the emergence of dif.

### *2.2. DreamWaQ: Learning Robust Quadrupedal Locomotion With Implicit Terrain Imagination via Deep Reinforcement Learning:*

In their paper, "***DreamWaQ***", the authors introduce a framework designed to train a robust locomotion policy for quadruped robots using only proprioception inputs and deep reinforcement learning (RL) algorithms. Their contributions, articulated in three main points, underscore the innovative approach of their framework:

A. **Novel Locomotion Learning Framework:** The authors propose a pioneering locomotion learning framework characterized by an asymmetric actor-critic architecture. This architecture allows for the implicit imagination of terrain properties solely from proprioceptive inputs, representing a departure from traditional methods that rely on explicit terrain information.

B. **Context-Aided Estimator Network:** A context-aided estimator network is introduced to jointly estimate body state and environmental context. This integrated approach, in conjunction with the policy, outperforms existing learning-based methods, demonstrating the efficacy of leveraging contextual information in locomotion control.

C. **Robustness and Durability Evaluation:** The learned policy's robustness and durability are rigorously evaluated in real-world scenarios through walking experiments conducted in diverse outdoor environments. This empirical validation underscores the practical applicability and effectiveness of the proposed method in real-world settings.

At the core of ***DreamWaQ*** lies the concept of learning a robust representation of the state, serving as the foundation for predicting joint angles directly from proprioceptive inputs. The framework's learning process is illustrated in **Figure 1**, providing an overview of the ***DreamWaQ*** learning framework.

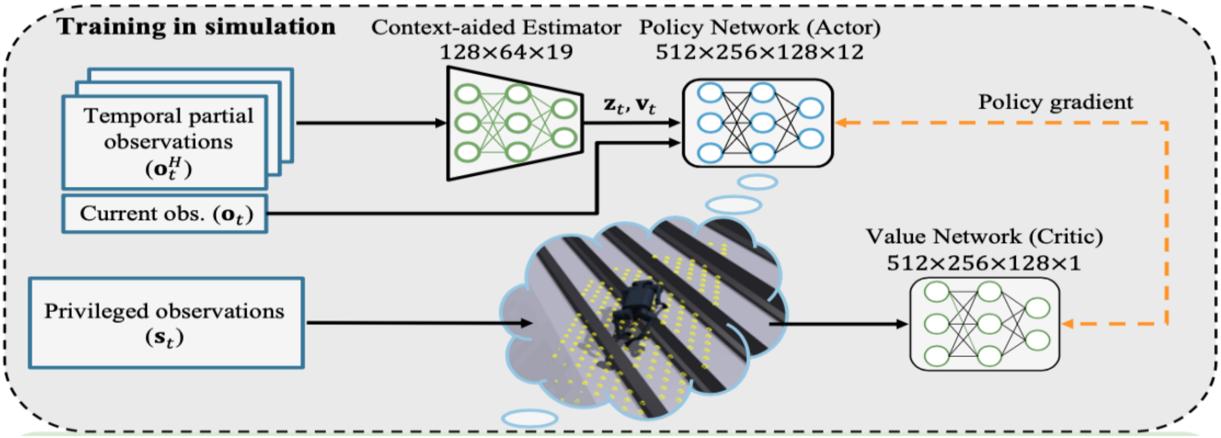

Training in simulation
Context-aided Estimator
128×64×19
Policy Network (Actor)
512×256×128×12
Temporal partial observations ($\mathbf{o}_t^H$)
Current obs. ($\mathbf{o}_t$)
$\mathbf{z}_t, \mathbf{v}_t$
Policy gradient
Value Network (Critic)
512×256×128×1
Privileged observations ($\mathbf{s}_t$)

**Fig. 1**

Recent research efforts have explored the teacher-student training paradigm, wherein a teacher network, trained in simulation with full state inputs including privileged extrinsic information such as terrain height maps and friction coefficients, serves as an expert policy to train a student network deployed in real-world hardware. However, the authors note certain limitations inherent in behavior cloning, particularly concerning the student policy's inability to learn failure states encountered by the teacher policy during early learning stages. Motivated by these limitations, the authors propose a unified learning framework featuring an asymmetric actor-critic architecture for robust locomotion behavior learning on uneven terrains.

## ***Terminologies***

$\mathbf{o}_t^H = \left[\mathbf{o}_t \quad \mathbf{o}_{t-1} \cdots \mathbf{o}_{t-H}\right]^T$ - temporal observation at time t over the past H measurements

$\mathbf{o}_t = \left[\boldsymbol{\omega}_t \quad \mathbf{g}_t \quad \mathbf{c}_t \quad \boldsymbol{\theta}_t \quad \dot{\boldsymbol{\theta}}_t \quad \mathbf{a}_{t-1}\right]^T$ - observation at time t (n x 1 vector)

$\mathbf{s}_t = \left[\mathbf{o}_t \quad \mathbf{v}_t \quad \mathbf{d}_t \quad \mathbf{h}_t\right]^T$ - privileged observation

**Z**t - latent representation of world state

**V**t - body linear velocity estimated by CENet

## ***Policy Network***

The observations to the policy network are:

1. **O**t
2. **Z**t
3. **V**t

**Z**t and **V**t are estimated by the Context-Aided Estimator Network while **O**t is obtained from joint encoders and IMU. Since the policy network is provided only with the partial observations, it ensures seamless transition to hardware implementation, thus bypassing the usual method of training a student network architecture.

### ***Value Network***

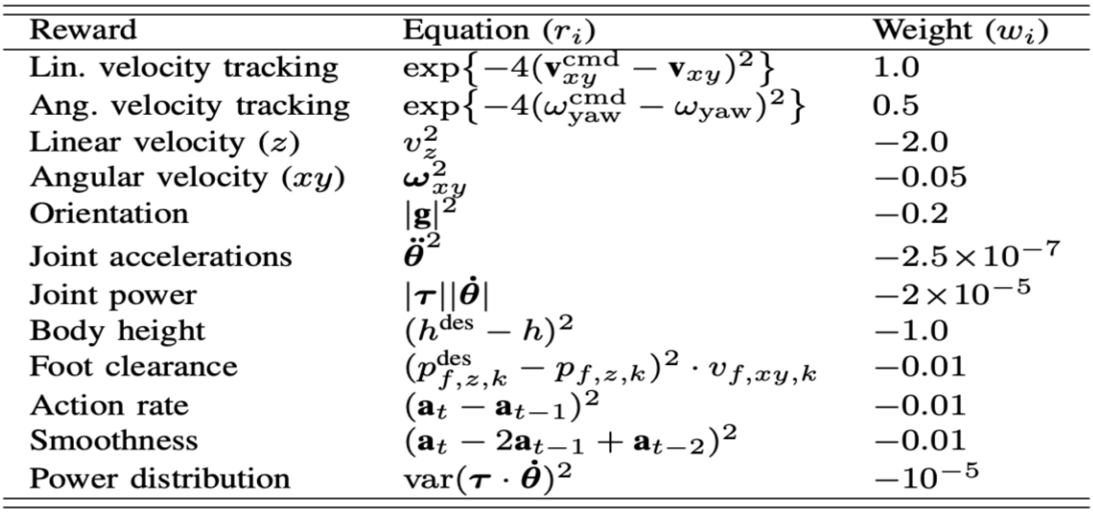
The value network receives the full state of the world, which includes partial observation **Ot,** body velocity **Vt,** disturbance force **dt** and height map scan **ht** and is trained to output a single value that represents the value of the state.

### ***Action Space***

The action space is a 12 x 1 vector representing the target joint angles of the robot with respect to the robot's initial stand still pose.

$$\boldsymbol{\theta}_{\text{des}} = \boldsymbol{\theta}_{\text{stand}} + \mathbf{a}_t.$$

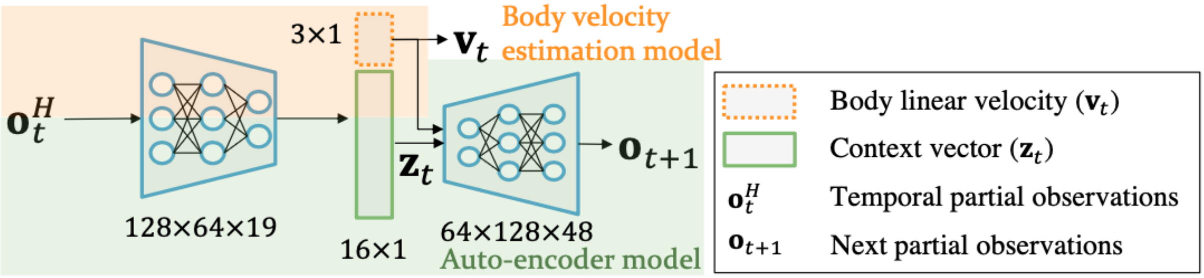
### ***Reward Function***



| Reward | Equation ($r_i$) | Weight ($w_i$) |
|---|---|---|
| Lin. velocity tracking | $\exp\{-4(\mathbf{v}_{xy}^{\text{cmd}} - \mathbf{v}_{xy})^2\}$ | 1.0 |
| Ang. velocity tracking | $\exp\{-4(\omega_{\text{yaw}}^{\text{cmd}} - \omega_{\text{yaw}})^2\}$ | 0.5 |
| Linear velocity ($z$) | $v_z^2$ | $-2.0$ |
| Angular velocity ($xy$) | $\boldsymbol{\omega}_{xy}^2$ | $-0.05$ |
| Orientation | $\|\mathbf{g}\|^2$ | $-0.2$ |
| Joint accelerations | $\ddot{\boldsymbol{\theta}}^2$ | $-2.5\times10^{-7}$ |
| Joint power | $\|\boldsymbol{\tau}\|\|\dot{\boldsymbol{\theta}}\|$ | $-2\times10^{-5}$ |
| Body height | $(h^{\text{des}} - h)^2$ | $-1.0$ |
| Foot clearance | $(p_{f,z,k}^{\text{des}} - p_{f,z,k})^2 \cdot v_{f,xy,k}$ | $-0.01$ |
| Action rate | $(\mathbf{a}_t - \mathbf{a}_{t-1})^2$ | $-0.01$ |
| Smoothness | $(\mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2})^2$ | $-0.01$ |
| Power distribution | $\text{var}(\boldsymbol{\tau} \cdot \dot{\boldsymbol{\theta}})^2$ | $-10^{-5}$ |

**Table. 1**

## ***Context - Aided Estimator Network***



Body velocity estimation model
3×1
$\mathbf{v}_t$
$\mathbf{o}_t^H$
128×64×19
16×1
$\mathbf{z}_t$
$\mathbf{o}_{t+1}$
64×128×48
Auto-encoder model
Body linear velocity ($\mathbf{v}_t$)
Context vector ($\mathbf{z}_t$)
$\mathbf{o}_t^H$ Temporal partial observations
$\mathbf{o}_{t+1}$ Next partial observations

**Fig. 2**

The ***Context-Aided Estimator Network*** is the centerpiece of this paper. In a higher level, CENet predicts a context vector **Zt,** that intends to represent the terrain properties, and the body linear velocity **Vt,** which in turn helps to get a better understanding of the proprioceptive inputs. The latent vector **Zt,** is used to estimate **Vt** as well as to predict **Ot+1**. CENet architecture can be inferred from **Fig. 3** which depicts an auto-encoder architecture. The authors use a β- variational auto-encoder (β-VAE) as the auto- encoder architecture. CENet is optimized using a hybrid loss function, defined as follows:

$$\mathcal{L}_{\mathrm{CE}} = \mathcal{L}_{\mathrm{est}} + \mathcal{L}_{\mathrm{VAE}}$$

where Lest and LVAE are the body velocity estimation and VAE loss, respectively. The VAE network is trained with the standard B-VAE loss, which consists of reconstruction and latent losses. The authors employed MSE for the reconstruction loss and Kullback-Leibler (KL) divergence as the latent loss. The VAE loss is formulated as:

$$\mathcal{L}_{\mathrm{VAE}} = MSE(\tilde{\mathbf{o}}_{t+1}, \mathbf{o}_{t+1}) + \beta D_{\mathrm{KL}}(q(\mathbf{z}_t|\mathbf{o}_t^H) \parallel p(\mathbf{z}_t))$$

where **Ot+1** is the reconstructed next observation, $q(\mathbf{z}_t|\mathbf{o}_t^H)$ is the posterior distribution of the at given $\mathbf{o}_t^H.\ p(\mathbf{z}_t)$ is the context's prior distribution parameterized by a Gaussian distribution.

### 2.3. *Walk These Ways*: Tuning Robot Control for Generalization with Multiplicity of Behavior:

"Walk These Ways" proposes a comprehensive framework comprising a low-level controller, an interpretable high-level control interface, and MoB techniques. These components work synergistically to enable quadruped robots to execute various behaviors across different terrains.

#### *Task and Behavior Sampling*

In order to learn graceful online transitions between behaviors, the authors resample the desired task and behavior within each training episode. To enable the robot to both run and spin fast, the authors sample task c = ($v^{cmd}$,$v^{cmd}$,$\omega^{cmd}$) using the grid adaptive curriculum strategy txyz from [3]. Then, they sampled a target behavior b .

First, they sampled ($\theta^{cmd}$,$\theta^{cmd}$,$\theta^{cmd}$) as t 123 one of the symmetric quadrupedal contact patterns (pronking, trotting, bounding, or pacing) which are known as more stable and which we found a sufficient basis for diverse useful gaits. Then, the remaining command parameters ($v^{cmd}$,$f^{cmd}$,$h^{cmd}$,$\varphi^{cmd}$,$h^{fcmd}$,$s^{cmd}$) are sampled independently yzzy and uniformly. Their ranges are given in Table 6.

#### *Policy Input*

The input to the policy is a 30-step history of observations ot−H...t, commands ct−H...t, behaviors bt−H...t, previous actions at−H−1...t−1, and timing reference variables tt−H...t.

The observation space ot consists of joint positions and velocities $q_t$,$\dot{q}_t$ (measured by joint encoders) and the gravity vector in the body frame $g_t$ (measured by accelerometer). The timing reference variables

tt =[sin(2πt$^{FR}$),sin(2πt$^{FL}$),sin(2πt$^{RR}$),sin(2πt$^{RL}$)]

are computed from the offset timings of each foot:

[$t^{FR}$, $t^{FL}$, $t^{RR}$, $t^{RL}$] = [t + $\theta^{cmd}$ + $\theta^{cmd}$, t + $\theta^{cmd}$ + $\theta^{cmd}$, t + $\theta^{cmd}$, t + $\theta^{cmd}$], 231312 where t is a counter variable that advances from 0 to 1 during each gait cycle and FR, FL, RR, RL are the four feet. This form is adapted from [8] to express quadrupedal gaits.

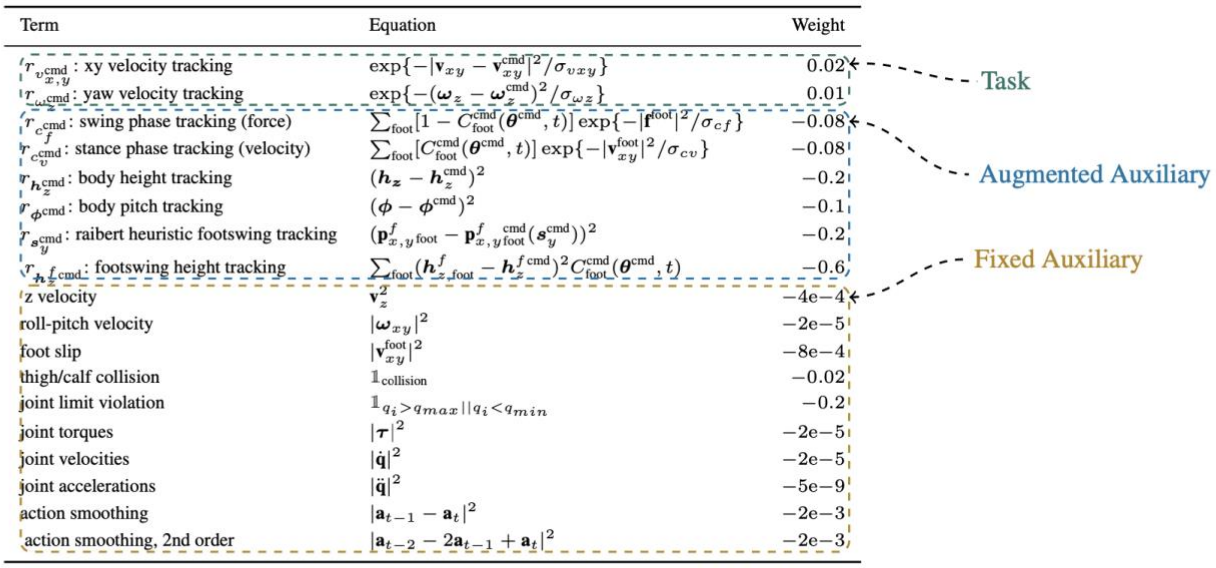
### *Policy Architecture*

The policy body is an MLP with hidden layer sizes [512, 256, 128] and ELU activations. Besides the above, the policy input also includes estimated domain parameters: the velocity of the robot body and the ground friction, which are predicted from the observation history using supervised learning in the manner of [7]. The estimator module is an MLP with hidden layer sizes [256, 128] and ELU activations. They did not analyze the impact of this estimation on performance but found it useful for visualizing deployments.

### *Action Space*

The action $a_t$ consists of position targets for each of the twelve joints. A zero action corresponds to the nominal joint position, qˆ. The position targets are tracked using a proportional- derivative controller with $k_p = 20$, $k_d = 0.5$.

### *Reward Structure*



| Term | Equation | Weight |
|---|---|---|
| $r_{v^{\text{cmd}}_{x,y}}$ : xy velocity tracking | $\exp\{-\lvert \mathbf{v}_{xy} - \mathbf{v}^{\text{cmd}}_{xy}\rvert^2/\sigma_{vxy}\}$ | 0.02 |
| $r_{\omega^{\text{cmd}}_z}$: yaw velocity tracking | $\exp\{-(\boldsymbol{\omega}_z - \boldsymbol{\omega}^{\text{cmd}}_z)^2/\sigma_{\omega z}\}$ | 0.01 |
| $r_{c^{\text{cmd}}_f}$: swing phase tracking (force) | $\sum_{\text{foot}}[1 - C^{\text{cmd}}_{\text{foot}}(\boldsymbol{\theta}^{\text{cmd}}, t)] \exp\{-\lvert \mathbf{f}^{\text{foot}}\rvert^2/\sigma_{cf}\}$ | $-0.08$ |
| $r_{c^{\text{cmd}}_v}$: stance phase tracking (velocity) | $\sum_{\text{foot}}[C^{\text{cmd}}_{\text{foot}}(\boldsymbol{\theta}^{\text{cmd}}, t)] \exp\{-\lvert \mathbf{v}^{\text{foot}}_{xy}\rvert^2/\sigma_{cv}\}$ | $-0.08$ |
| $r_{\boldsymbol{h}^{\text{cmd}}_z}$: body height tracking | $(\boldsymbol{h_z} - \boldsymbol{h}^{\text{cmd}}_z)^2$ | $-0.2$ |
| $r_{\boldsymbol{\phi}^{\text{cmd}}}$: body pitch tracking | $(\boldsymbol{\phi} - \boldsymbol{\phi}^{\text{cmd}})^2$ | $-0.1$ |
| $r_{\boldsymbol{s}^{\text{cmd}}_y}$: raibert heuristic footswing tracking | $(\mathbf{p}^f_{x,y\,\text{foot}} - \mathbf{p}^{f}_{x,y\,\text{foot}}{}^{\text{cmd}}(\boldsymbol{s}^{\text{cmd}}_y))^2$ | $-0.2$ |
| $r_{\boldsymbol{h}^{f\,\text{cmd}}_z}$: footswing height tracking | $\sum_{\text{foot}}(\boldsymbol{h}^f_{z,\text{foot}} - \boldsymbol{h}^{f\,\text{cmd}}_z)^2 C^{\text{cmd}}_{\text{foot}}(\boldsymbol{\theta}^{\text{cmd}}, t)$ | $-0.6$ |
| z velocity | $\mathbf{v}^2_z$ | $-4\text{e}-4$ |
| roll-pitch velocity | $\lvert \boldsymbol{\omega}_{xy}\rvert^2$ | $-2\text{e}-5$ |
| foot slip | $\lvert \mathbf{v}^{\text{foot}}_{xy}\rvert^2$ | $-8\text{e}-4$ |
| thigh/calf collision | $\mathbb{1}_{\text{collision}}$ | $-0.02$ |
| joint limit violation | $\mathbb{1}_{q_i > q_{max} \lvert\lvert q_i < q_{min}}$ | $-0.2$ |
| joint torques | $\lvert \boldsymbol{\tau}\rvert^2$ | $-2\text{e}-5$ |
| joint velocities | $\lvert \dot{\mathbf{q}}\rvert^2$ | $-2\text{e}-5$ |
| joint accelerations | $\lvert \ddot{\mathbf{q}}\rvert^2$ | $-5\text{e}-9$ |
| action smoothing | $\lvert \mathbf{a}_{t-1} - \mathbf{a}_t\rvert^2$ | $-2\text{e}-3$ |
| action smoothing, 2nd order | $\lvert \mathbf{a}_{t-2} - 2\mathbf{a}_{t-1} + \mathbf{a}_t\rvert^2$ | $-2\text{e}-3$ |

**Table. 2**

### *2.4. Minimizing Energy Consumption Leads to the Emergence of Gaits in Legged Robots*

In this paper, the authors show that learning to minimize energy consumption plays a key role in the emergence of natural locomotion gaits at different speeds in real quadruped robots. The same approach leads to unstructured gaits in rough terrains which is consistent with the findings in animal motor control.

Locomotion consumes a significant fraction of an animal's metabolic energy, suggesting that development of different gaits such as walk, trot, gallop, etc. are energy efficient at certain range of speeds.

It also points out the fact that animals transition between different gaits at different speeds in order to minimize their energy consumption. In this work, the authors design an end-to-end learning framework to show how energy minimization leads to the emergence of structured locomotion gait patterns in flat terrains as well as unstructured gaits in complex terrains at different commanded speeds.

This work leverages the use of the teacher-student training paradigm. The teacher network is provided with proprioceptive inputs along with privileged extrinsics information such as terrain height, terrain normal, gravity vector, etc at every time step and is trained in simulation. The student network has access to only the current proprioceptive inputs and history of proprioceptive inputs and predicted action outputs.

The goal of the student policy is to mimic the behavior of the teacher policy. More importantly, the student policy must be able to infer the privileged information with the history of proprioceptive inputs and actions.

The authors capitalize on their prior work, "RMA: Rapid Motor Adaptation for Legged Robots," to facilitate the adaptation of the policy learned in simulation onto physical hardware. This process is executed through the implementation of the student-teacher framework.

The main contributions of this paper, as stated by the authors include:

- Show that minimizing energy consumption plays a key role in the emergence of natural loco- motion patterns in both flat as well as complex terrains at different speeds without relying on demonstrations or predefined motion heuristics.

- Show that the emergent gaits at different target speeds correspond to conventional animals in the similar Froude number range (sheep/horse) without any sort of pre-programming.

- Present a distillation-based learning pipeline to obtain velocity-conditioned policy that displays smooth gait transition as the target speed is changed.

- Demonstrate the emergent behaviors, robustness analysis, and gait patterns in simulation as well as a real-world budget quadruped robot.

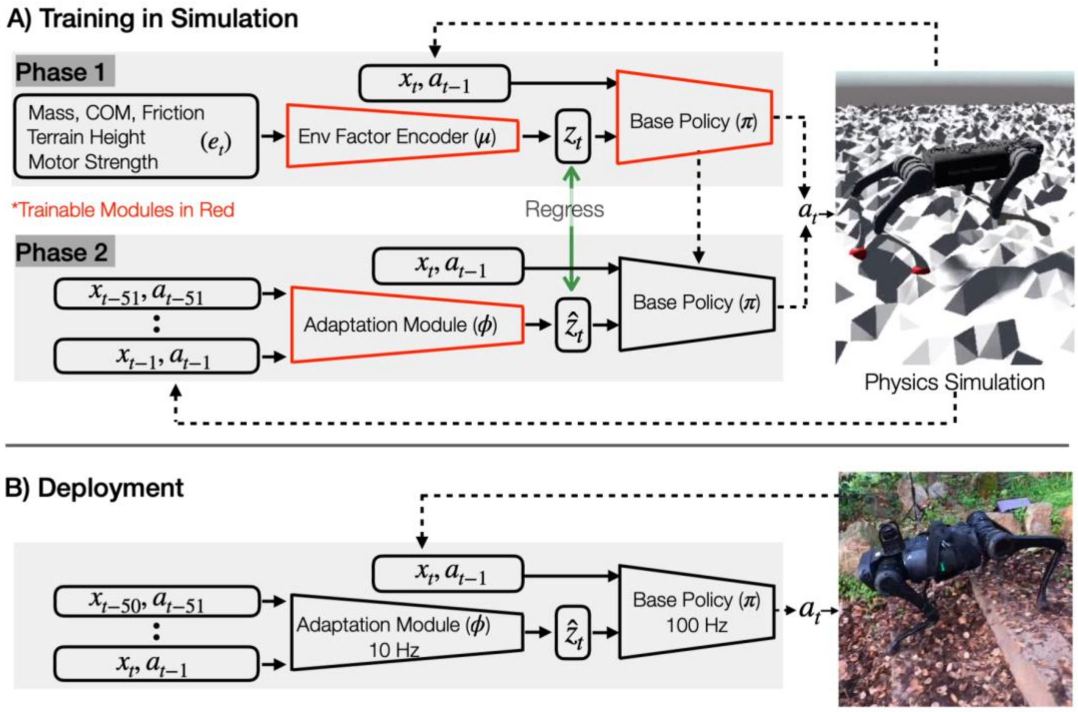


Fig. 2: RMA consists of two subsystems - the base policy $\pi$ and the adaptation module $\phi$. **Top:** RMA is trained in two phases. In the first phase, the base policy $\pi$ takes as input the current state $x_t$, previous action $a_{t-1}$ and the privileged environmental factors $e_t$ which is encoded into the latent extrinsics vector $z_t$ using the environmental factor encoder $\mu$. The base policy is trained in simulation using model-free RL. In the second phase, the adaptation module $\phi$ is trained to predict the extrinsics $\hat{z}_t$ from the history of state and actions via supervised learning with on-policy data. **Bottom:** At deployment, the adaptation module $\phi$ generates the extrinsics $\hat{z}_t$ at 10Hz, and the base policy generates the desired joint positions at 100Hz which are converted to torques using A1's PD controller. Since the adaptation module runs at a lower frequency, the base policy consumes the most recent extrinsics vector $\hat{z}_t$ predicted by the adaptation module to predict $a_t$. This asynchronous design was critical for seamless deployment on low-cost robots like A1 with limited on-board compute. Videos at: https://ashish-kmr.github.io/rma-legged-robots/

**Fig. 3**

## State Space

The network architectures of the policy network and the value network are symmetric. The state is 30 dimensional containing the joint positions (12 values), joint velocities (12 values), roll and pitch of the torso and binary foot contact indicators (4 values). The environment information as depicted in Fig. 4 includes center of mass position and the payload (3 dimensions), motor strength (12 dimensions), friction (1 dimension), linear speed in x direction vx (1 dimension), linear speed in y direction vy (1 dimension) and yaw speed ωyaw (1 dimension), making it a 19-dim vector.

**Action Space**

The action space is 12 dimensional corresponding to the target joint position for the 12 robot joints. The predicted joint angles are with respect to the robot’s initial stand-still position.

$$\boldsymbol{\theta}_{\text{des}} = \boldsymbol{\theta}_{\text{stand}} + \mathbf{a}_t .$$

**Reward Function**

$$r = r_{\text{forward}} + \alpha_1 * r_{\text{energy}} + r_{\text{alive}}$$

$$r_{\text{forward}} = -\alpha_2 * \left| v_x - v_x^{\text{target}} \right| - \left| v_y \right|^2 - \left| \omega_{\text{yaw}} \right|^2$$

$$r_{\text{energy}} = -\boldsymbol{\tau}^T \dot{\boldsymbol{q}},$$

The total reward is the summation of three reward terms, namely, forward reward, energy reward and survival reward. The forward reward term rewards the agent for walking straight at the specified speed, energy reward term penalizes energy consumption and the survival reward term is the survival bonus.

The authors use the A1 URDF to simulate the A1 robot in the RaiSim simulator. They generate complex terrains using the inbuilt fractal terrain generator for flat and uneven surfaces. They claim that tra the policy on a completely flat surface results in unnatural gaits and leads to lesser foot clearance from the ground.

Hence, they train the policies on simple fractal terrains with varying frequency of terrain heights instead of perfectly flat terrain. The policies are tested at 3 different target speeds, namely, 0.375 m/s, 0.9 m/s and 1.5 m/s. Walk gait is observed at 0.375 m/s, trot gait emerges at 0.9 m/s and gallop gait develops at 1.5 m/s.

# 4. OBJECTIVES

The objectives of this work are as follows:

- We focus on developing RL-based controllers for quadruped locomotion over different types of terrains using multiple gaits such as “trotting”, “hopping”, “bounding”, “pacing”.

- We combine the advantages of RL policies with emergent gaits that learn to traverse over uneven terrains and RL policies that enforce Multiplicity of Behaviors such as the type of gait, body height commands, step height commands, etc.

- Furthermore, we incorporate ‘Control Barrier Function’ based rewards to make our controller less aggressive and more energy efficient.

- Finally, we adopt our  RL policy onto our in-house build quadruped robots – “Mule” and “Stoch 3”.

# 4. METHODOLOGY AND WORK PLAN

We will first discuss about the state space used for training our controller using PPO algorithm. Subsequently, we will describe our asymmetric reward function and its role in acquiring locomotion behaviour on both flat and non-flat terrain.

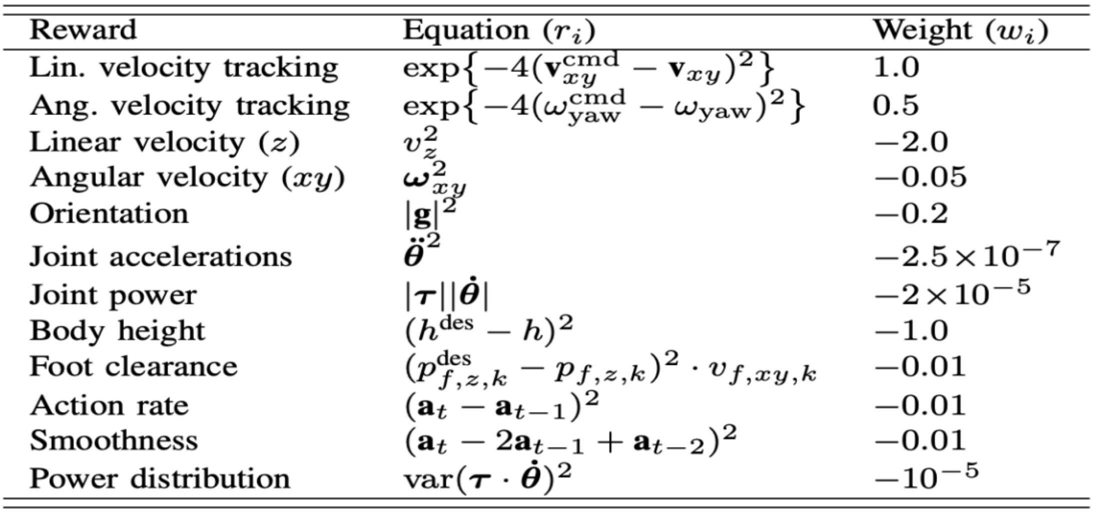
## A. State and action space

In this work, we utilize different states as inputs to the actor and critic modules. The need for asymmetric inputs arises due to the partial observability of the task of quadruped locomotion. Properties such as friction and restitution coefficients are not observed directly by the controller and hence are not part of the actor's input but are provided as input to the critic. The usage of asymmetric PPO is motivated by the performance benefits proven empirically in the study [11]. The input passed to the actor contains a history of length 30 $(o_{t-30:t})$ following observations $(o_t)$ : estimated linear velocity of the base $(v_t)$ angular velocity of the base $(\omega_t)$ , gravity vector with respect to the body frame $(g_t)$, joint positions $(q_t)$ , joint velocities $((\dot{q})_t)$, timing reference variable $(\tau_t)$ [8]. Further, the following commands are passed as part of the input: x-y linear velocity $(v^{cmd}, v^{cmd})$, yaw x-y rate $(\omega^{cmd})$, body height $(h^{cmd})$, step frequency $(f^{cmd})$, step height $(f\ h^{cmd})$, stance width $(s^{cmd})$, gait parameters $(\theta^{cmd}, \theta^{cmd}, \theta^{cmd})$ [8].

The quadruped is controlled by using joint position commands. The RL policy predicts perturbation (a_t) about the default joint position $(q_{def})$, and the resultant joint position $(a_t + q_{def})$ is finally sent to the PD controller. While training the quadruped in simulation, we used the actuator net [13], but for hardware deployment, we used a PD controller with a proportional gain of 20 and a derivative gain of 0.5 for unitree Go1 robot and a proportional gain of 220 and derivative gain of 3 for “Mule”.

### B. Reward Function

Our reward function consists of standard reward terms such as linear and angular velocity tracking reward, body height reward, orientation reward, etc. The entire list of reward term used in given in Fig [4]. However, our approach differs from the other works in term of the usage of asymmetric rewards for training on flat and non-flat terrains.



| Reward | Equation ($r_i$) | Weight ($w_i$) |
|---|---|---|
| Lin. velocity tracking | $\exp\{-4(\mathbf{v}_{xy}^{\text{cmd}} - \mathbf{v}_{xy})^2\}$ | $1.0$ |
| Ang. velocity tracking | $\exp\{-4(\omega_{\text{yaw}}^{\text{cmd}} - \omega_{\text{yaw}})^2\}$ | $0.5$ |
| Linear velocity ($z$) | $v_z^2$ | $-2.0$ |
| Angular velocity ($xy$) | $\boldsymbol{\omega}_{xy}^2$ | $-0.05$ |
| Orientation | $\lvert\mathbf{g}\rvert^2$ | $-0.2$ |
| Joint accelerations | $\ddot{\boldsymbol{\theta}}^2$ | $-2.5 \times 10^{-7}$ |
| Joint power | $\lvert\boldsymbol{\tau}\rvert\lvert\dot{\boldsymbol{\theta}}\rvert$ | $-2 \times 10^{-5}$ |
| Body height | $(h^{\text{des}} - h)^2$ | $-1.0$ |
| Foot clearance | $(p_{f,z,k}^{\text{des}} - p_{f,z,k})^2 \cdot v_{f,xy,k}$ | $-0.01$ |
| Action rate | $(\mathbf{a}_t - \mathbf{a}_{t-1})^2$ | $-0.01$ |
| Smoothness | $(\mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2})^2$ | $-0.01$ |
| Power distribution | $\text{var}(\boldsymbol{\tau} \cdot \dot{\boldsymbol{\theta}})^2$ | $-10^{-5}$ |

**Table. 3**

$$\Sigma_f = (h_z^f - h_g - h_z^f(cmd))^2 C^f(cmd)(\theta, t).$$

$$h_z^f - footheight\ in\ global\ frame$$

$$h_g - Ground\ height\ at\ position\ x + \delta_x\ ,\ y + \delta_y\ ,\ z + \delta_z\ where\ x\ y\ z\ are\ foot\ locations$$

The above modified reward encourages the agent to track step height targets relative to a delta x, y, z position added relative to its foot location in the direction of motion of the robot. Adding this delta x, y, z helps the robot to lift its leg up while in contact with stairs and traverse forward over slopes even with lower velocity commands.

### C. Asymmetric Actor Critic

The observation input provided to the policy and critic networks is delineated in the table above. Notably, the critic network is granted access to the full state St, enabling it to comprehensively assess the current environment.

In contrast, the policy network receives a history of observations and estimates privileged information through latent representations gleaned from this observation history. This nuanced approach ensures that both networks are equipped with pertinent information tailored to their respective roles in the decision-making process, thereby enhancing the effectiveness and adaptability of the overall system.

### D. Asymmetric Rewards

In order to encourage our agent to learn behaviours as well as learn to traverse uneven terrains, we introduce an asymmetric reward structure. We train our agent in one-stage for learning both behaviours as well as locomotion.

In our experiments, we dedicate x% of the total terrain area as flat surface and (1-x) % for uneven surfaces composed of stairs and slopes. x % of the total robots initialised on the flat surface receive full scales of auxiliary rewards while the rest of the robot initialised on uneven ground receive a reduced scale of auxiliary rewards.

- ***Rationale***

By implementing asymmetric rewards, our approach differentiates between robots navigating flat surfaces and those traversing uneven terrain. On flat surfaces, the emphasis is placed on learning and accurately tracking behavioral commands.

Conversely, for robots navigating uneven terrain, the priority shifts to effectively tracking velocity and successfully traversing the challenging landscape, with less strict adherence to auxiliary commands. This tailored approach optimizes performance based on the specific demands of each surface type, enhancing overall adaptability and efficiency in locomotion tasks.

## Architecture Diagram

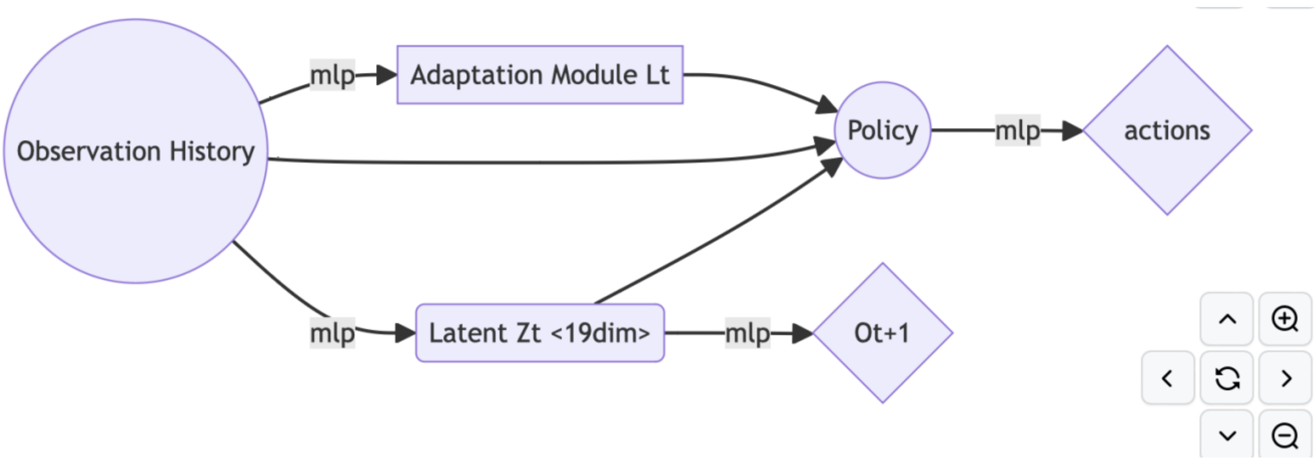


**Fig. 4**

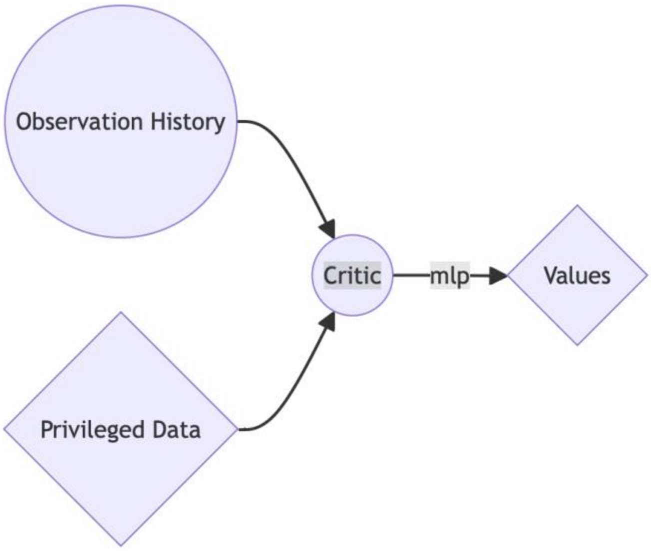
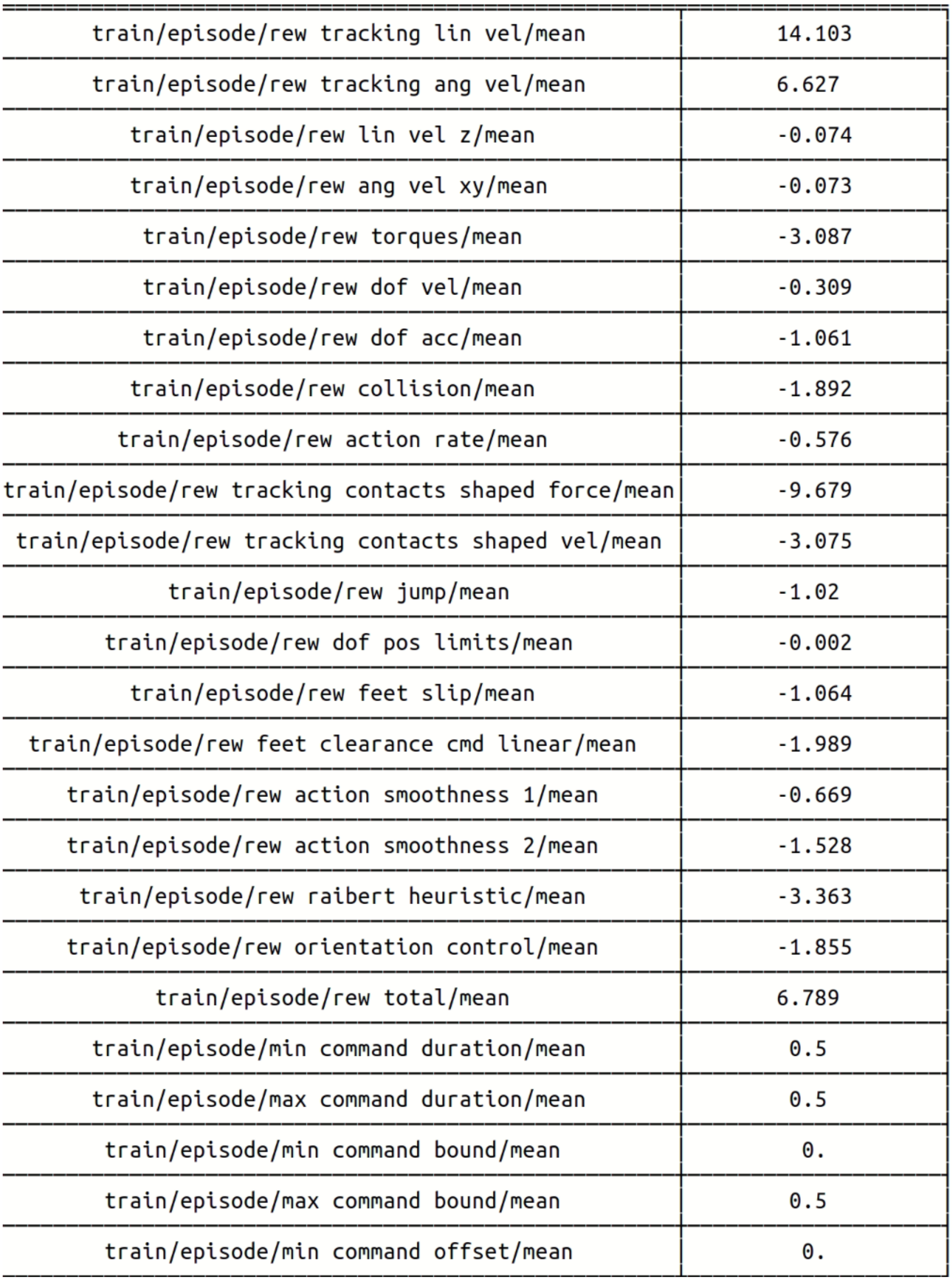


**Fig. 5**

# 5. RESULTS AND DISCUSSION



```
train/episode/rew tracking lin vel/mean                  14.103
train/episode/rew tracking ang vel/mean                  6.627
train/episode/rew lin vel z/mean                         -0.074
train/episode/rew ang vel xy/mean                        -0.073
train/episode/rew torques/mean                           -3.087
train/episode/rew dof vel/mean                           -0.309
train/episode/rew dof acc/mean                           -1.061
train/episode/rew collision/mean                         -1.892
train/episode/rew action rate/mean                       -0.576
train/episode/rew tracking contacts shaped force/mean    -9.679
train/episode/rew tracking contacts shaped vel/mean      -3.075
train/episode/rew jump/mean                              -1.02
train/episode/rew dof pos limits/mean                    -0.002
train/episode/rew feet slip/mean                         -1.064
train/episode/rew feet clearance cmd linear/mean         -1.989
train/episode/rew action smoothness 1/mean               -0.669
train/episode/rew action smoothness 2/mean               -1.528
train/episode/rew raibert heuristic/mean                 -3.363
train/episode/rew orientation control/mean               -1.855
train/episode/rew total/mean                             6.789
train/episode/min command duration/mean                  0.5
train/episode/max command duration/mean                  0.5
train/episode/min command bound/mean                     0.
train/episode/max command bound/mean                     0.5
train/episode/min command offset/mean                    0.
```

**Table 4.**

The above table shows the reward statistics after training for 3500 iterations.

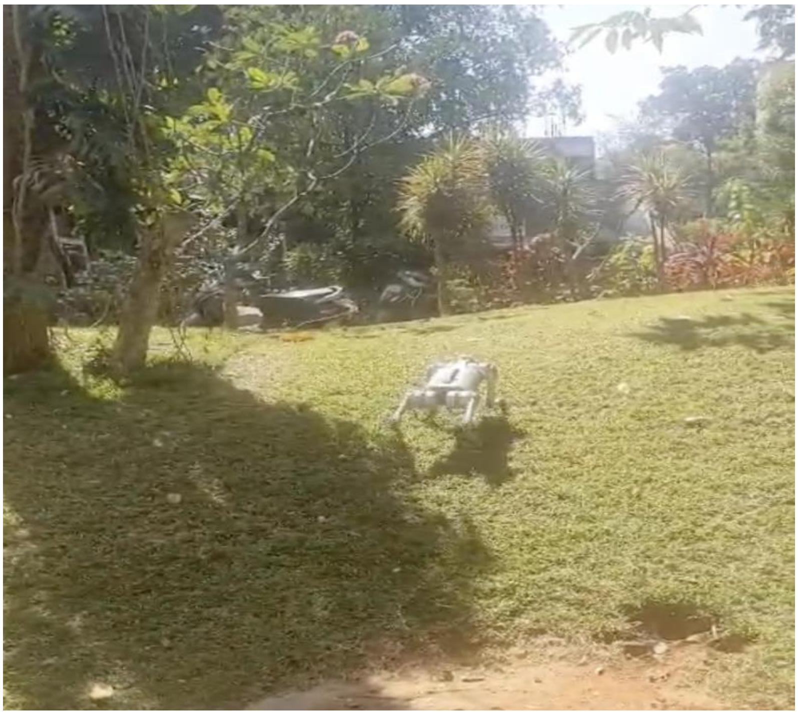


**Fig. 6**

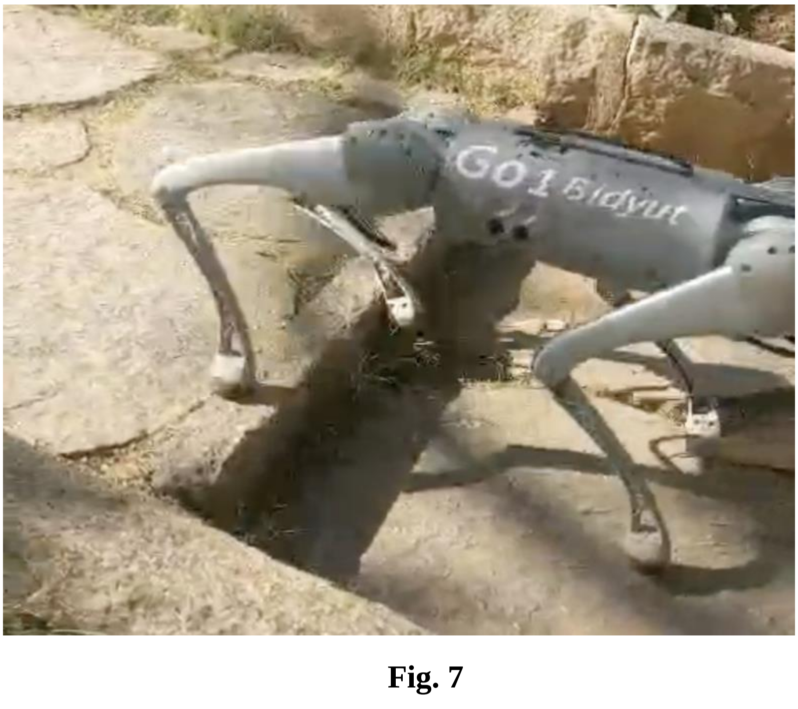


**Fig. 7**

The above figures show quadruped Unitree Go1 climbing stairs and slopes with our RL controller.

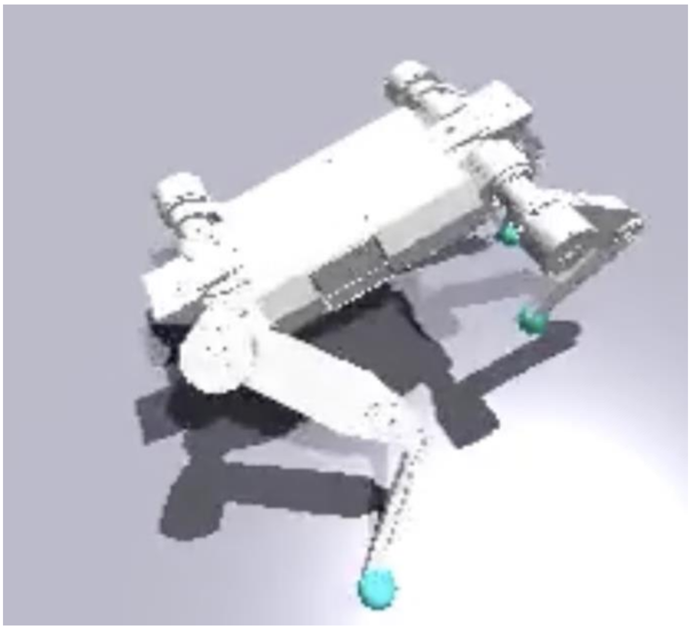


**Fig. 8**

Figure. 9 shows our RL controller implemented on our custom built quadruped “Mule” in the Isaac Gym simulation engine.

The observed overfitting of the Actor and Critic network weights on flat ground, coupled with the lack of generalization on uneven terrain, highlights a key challenge in locomotion control. The non-rich representations of flat ground observations limit the network's ability to adapt to diverse terrains, leading to suboptimal performance in real-world scenarios.

Training from scratch without asymmetric rewards exacerbates this issue, as evidenced by the training collapse observed around 3500 iterations. This underscores the importance of incorporating asymmetric rewards to encourage adaptive behavior across different surfaces, thereby promoting robustness and generalization in locomotion control.

The lack of smooth motion on stairs poses another challenge, indicating a dependency on the timer variable within the policy network. Our observations suggest that behaviors such as gaits, step height, and body height commands are heavily influenced by the timer variable, hindering the network's ability to learn skills independently of this dependency. Addressing this issue requires further exploration into reducing the network's reliance on the timer variable and promoting more diverse and adaptive locomotion strategies.

# 5. CONCLUSIONS AND FUTURE WORK

In this report, we have presented a novel approach to quadruped locomotion control that addresses the challenge of traversing diverse terrains. By integrating reinforcement learning (RL) techniques with proprioceptive observations, our methodology facilitates the emergence of adaptive gaits across varied terrain types. We have demonstrated the effectiveness of our approach in enabling quadrupeds to navigate uneven landscapes, while enforcing diverse behaviors such as gait selection, body height adjustments, and step height modulation.

The **asymmetric actor-critic** framework, coupled with an **asymmetric reward architecture**, enhances the adaptability and robustness of our learning process, simulating real-world partial observability scenarios and optimizing locomotion strategies across different terrains. Furthermore, the integration of **Control Barrier Function**-based rewards imbues our controller with less aggressive and more energy-efficient locomotion, enhancing adaptability and safety in complex environments.

Looking ahead, we envision leveraging **diffusion models** to further enhance our approach. By collecting data from existing techniques such as "**Walk These Ways**" and "**DreamWaQ**" and interpolating skills between them, diffusion models offer a promising avenue for learning and generalizing locomotion skills. This future step holds the potential to further improve the adaptability and efficiency of robotic locomotion systems, paving the way for agile and versatile quadrupedal locomotion in real-world scenarios.

# References

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

[2] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2016.

[3] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, “Learning to walk in minutes using massively parallel deep reinforcement learning,” in *Conference on Robot Learning*, pp. 91–100, PMLR, 2022.

[4] A. Kumar, Z. Fu, D. Pathak, and J. Malik, “RMA: Rapid Motor Adaptation for Legged Robots,” in *Proceedings of Robotics: Science and Systems*, (Virtual), July 2021.

[5] I. M. A. Nahrendra, B. Yu, and H. Myung, “Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5078–5084, IEEE, 2023.

[6] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning quadrupedal locomotion over challenging terrain,” *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[7] G. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, “Rapid Locomotion via Reinforcement Learning,” in *Proceedings of Robotics: Science and Systems*, (New York City, NY, USA), June 2022.

[8] G. B. Margolis and P. Agrawal, “Walk these ways: Tuning robot control for generalization with multiplicity of behavior,” in *Conference on Robot Learning*, pp. 22–31, PMLR, 2023.

[9] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

[10] J. Wu, Y. Xue, and C. Qi, “Learning multiple gaits within latent space for quadruped robots,” *arXiv preprint arXiv:2308.03014*, 2023.

[11] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, “Asymmetric actor critic for image-based robot learning,” in *Proceed- ings of Robotics: Science and Systems*, (Pittsburgh, Pennsylvania), June 2018.

[12] Nilaksh, A. Ranjan, S. Agrawal, A. Jain, P. Jagtap, and S. Kolathaya, “Barrier functions inspired reward shaping for reinforcement learning,” 2024.

[13] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, “Learning agile and dynamic motor skills for legged robots,” *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.