Learning Quadrupedal Locomotion using *Trust Region Policy Optimization* and *Gait Priors*

Ankit Sinha (Group 1)

Introduction - TRPO

- Trust Region Policy Optimization (TRPO) is a **policy gradient** algorithm in Reinforcement Learning.
- It guarantees stable and monotonic policy improvement during training.

$$L_{\pi_{old}}(\pi_\theta) = \mathbb{E}_{s\sim\rho_{\pi_{old}},\, a\sim\pi_{old}}\left[\frac{\pi_\theta(a|s)}{\pi_{old}(a|s)}A(s,a)\right]$$

$$\max L_{\pi_{old}}(\pi_\theta)$$

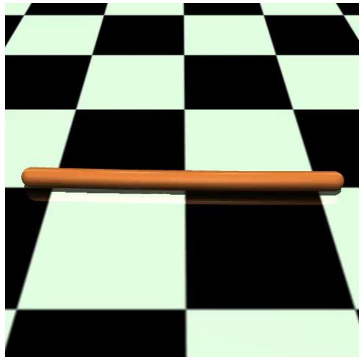
$$\text{subject to } \mathbb{D}_{KL}^{\max}(\pi_{old}||\pi_\theta) \leq \delta$$

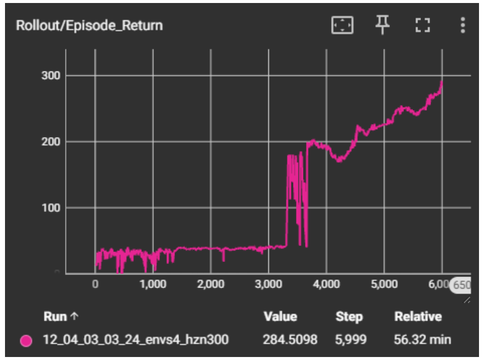
- This algorithm was published by *John Schulmann* and colleagues at ICML 2015.
- The predecessor NPG algorithm is a special case of TRPO with fixed step-size in the policy update step.

TRPO Algorithm

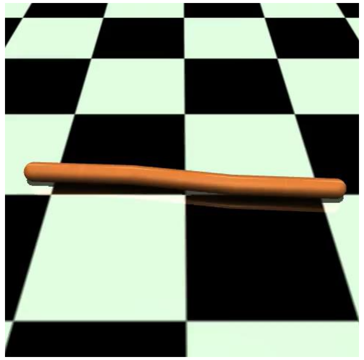
1. Collect trajectories using current policy $\pi_{\theta\,old}$
    1. Run policy in environment for N episodes or T total timesteps
    2. Store experience tuples $\{(s_t, a_t, r_t, s_{t+1})\}_{t=0}^{T}$
2. Compute rewards-to-go and *advantages*:
    1. Estimate value function $V(s_t)$ using a separate Critic or Monte-Carlo returns
    2. Compute *Generalized Advantage Estimates* (GAE) –
        1. $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ and $A_t = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \delta_{t+l}$
3. Define the surrogate objective to maximize:
    1. $L(\theta) = \frac{1}{T} \sum_{t=0}^{T-1} \rho_t(\theta) A_t$ where $\rho_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$
4. Compute the gradient $g = \nabla_\theta L(\theta)|_{\theta=\theta_{old}}$
5. Compute the Fisher-vector product function $Fv \rightarrow F(\theta_{old})v$
6. Solve $Fx = g$ using Conjugate Gradient (CG) algorithm
7. Compute the shrinkage factor to satisfy the KL constraint: $\Delta\theta = \sqrt{2\delta / x^T g} \cdot x$
8. Backtracking line search on step size α to ensure improvement and KL constraint holds.

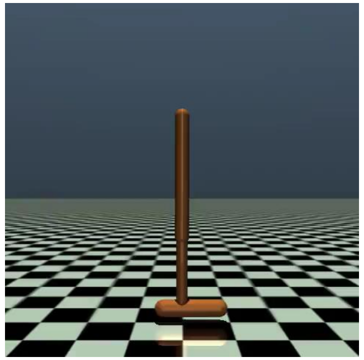
My TRPO Results – Mujoco tasks

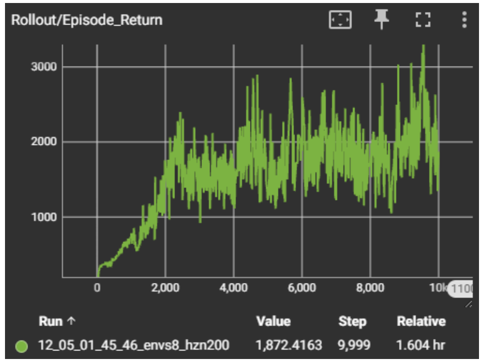




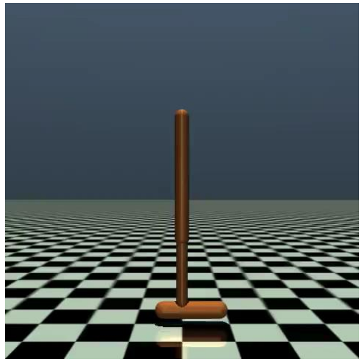
Run ↑	Value	Step	Relative
● 12_04_03_03_24_envs4_hzn300	284.5098	5,999	56.32 min

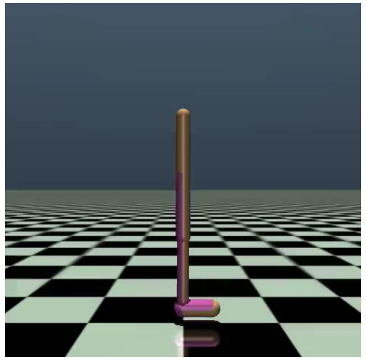


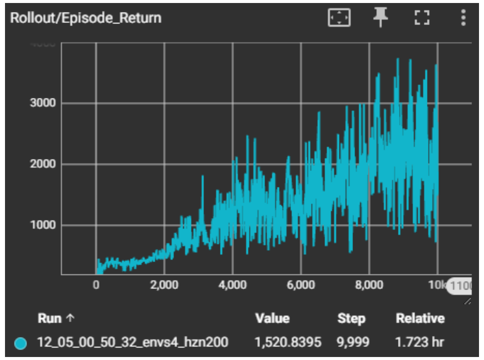




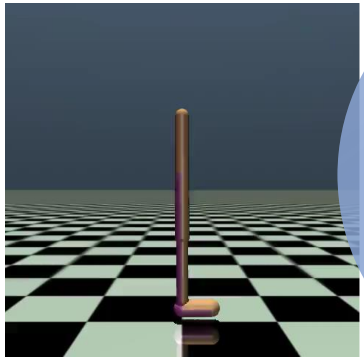
Run ↑	Value	Step	Relative
● 12_05_01_45_46_envs8_hzn200	1,872.4163	9,999	1.604 hr



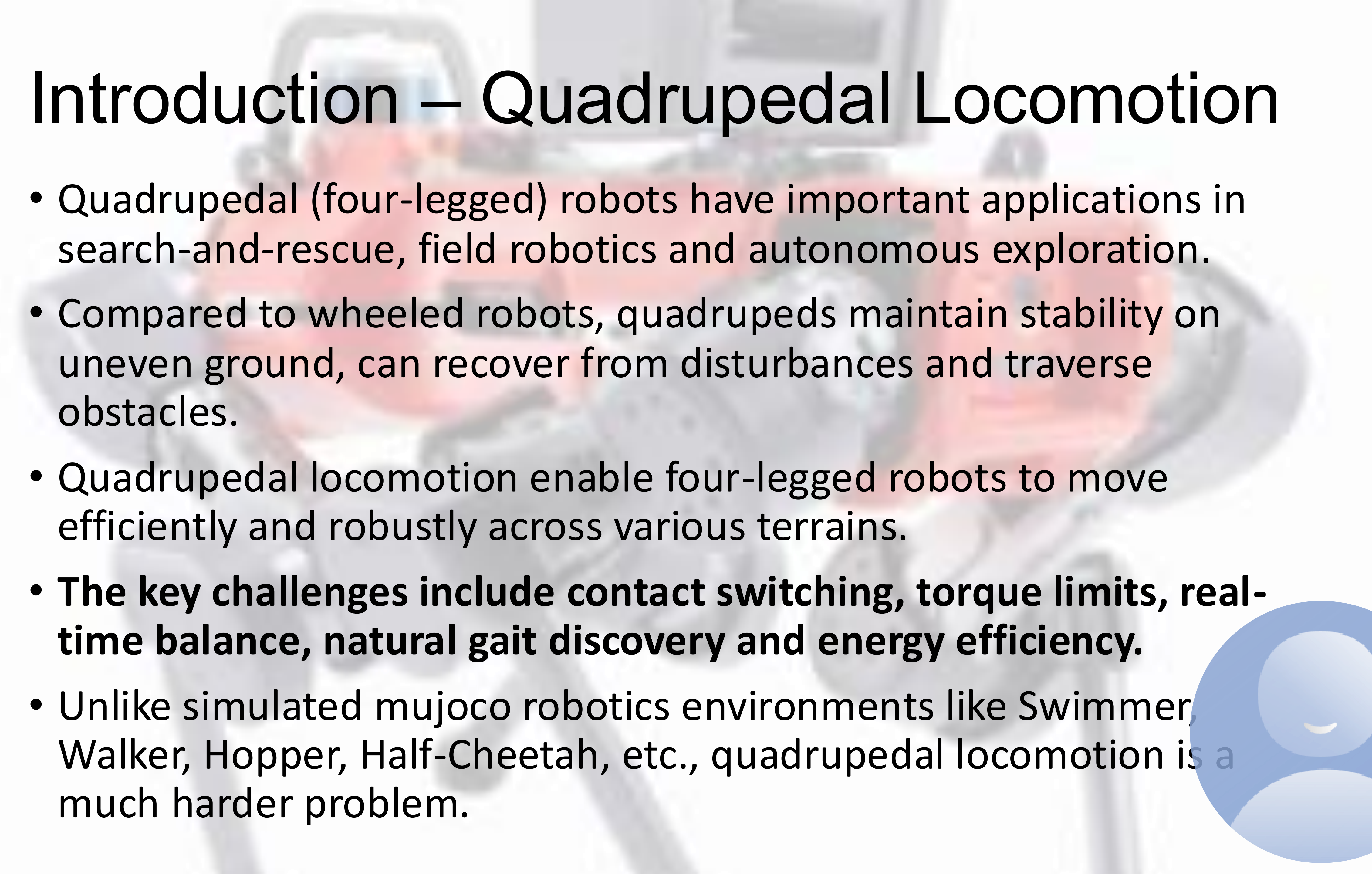




Run ↑	Value	Step	Relative
● 12_05_00_50_32_envs4_hzn200	1,520.8395	9,999	1.723 hr



Introduction – Quadrupedal Locomotion



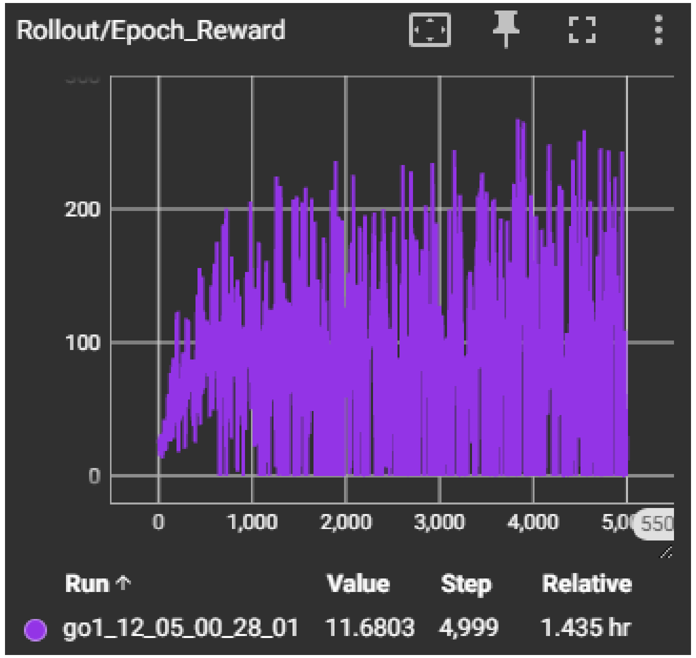
- Quadrupedal (four-legged) robots have important applications in search-and-rescue, field robotics and autonomous exploration.
- Compared to wheeled robots, quadrupeds maintain stability on uneven ground, can recover from disturbances and traverse obstacles.
- Quadrupedal locomotion enable four-legged robots to move efficiently and robustly across various terrains.
- **The key challenges include contact switching, torque limits, real-time balance, natural gait discovery and energy efficiency.**
- Unlike simulated mujoco robotics environments like Swimmer, Walker, Hopper, Half-Cheetah, etc., quadrupedal locomotion is a much harder problem.

Experimental Setup

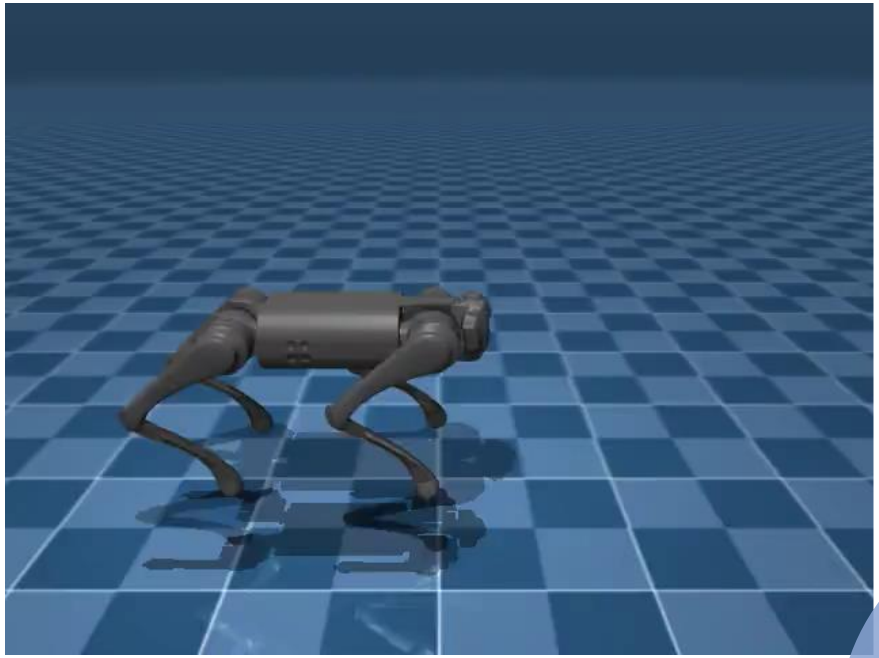
- **<u>Objective:</u>** Teach a quadrupedal robot to walk with a *natural*, *periodic*, *symmetric* and *smooth* gait!
- I used the MuJoCo physics simulator and built a gymnasium-style environment from scratch.
- The Go1 robot from the mujoco *menagerie* repository has been used for all experiments.
  - The robot has 12 active degrees of freedom.
  - All of its joints are position-controlled.
- The action space of the policy is 12 D (1d corresponding to each joint).
- The observation space consists of the (i) *joint positions* (ii) *joint velocities* (iii) *base orientation* (iv) *base linear velocity* (v) *base angular velocity* → 34 D.
- Reward Functions: (i) $r_{\text{vel}} = 2 \cdot exp\left(-\frac{(v-0.40)^2}{0.30^2}\right)$ (ii) $r_{alive} = \mathbb{1}$ (if episode does not terminate)
- Termination Criteria: If the robot's root height falls below 10 cm (i.e. robot base touches the ground).

Baseline: TRPO

- Here, the policy does not have any notion of periodicity or symmetry in leg movement.
- It tries to learn a forward motion from scratch.



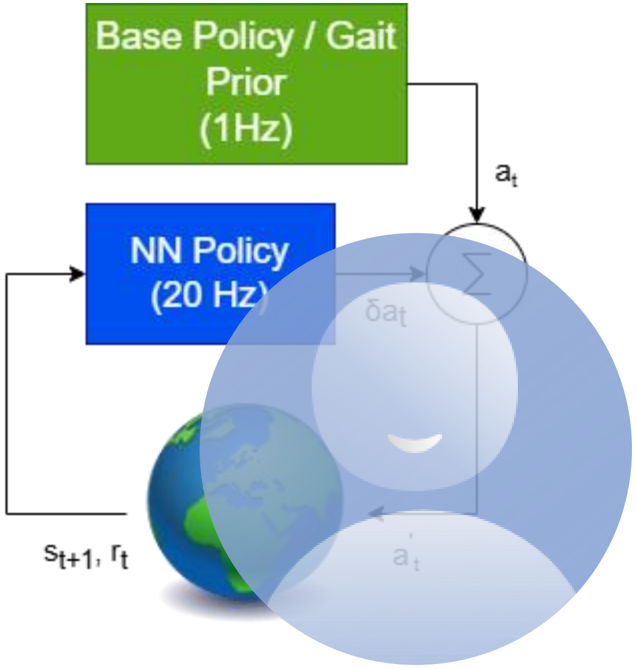
Run ↑	Value	Step	Relative
● go1_12_05_00_28_01	11.6803	4,999	1.435 hr



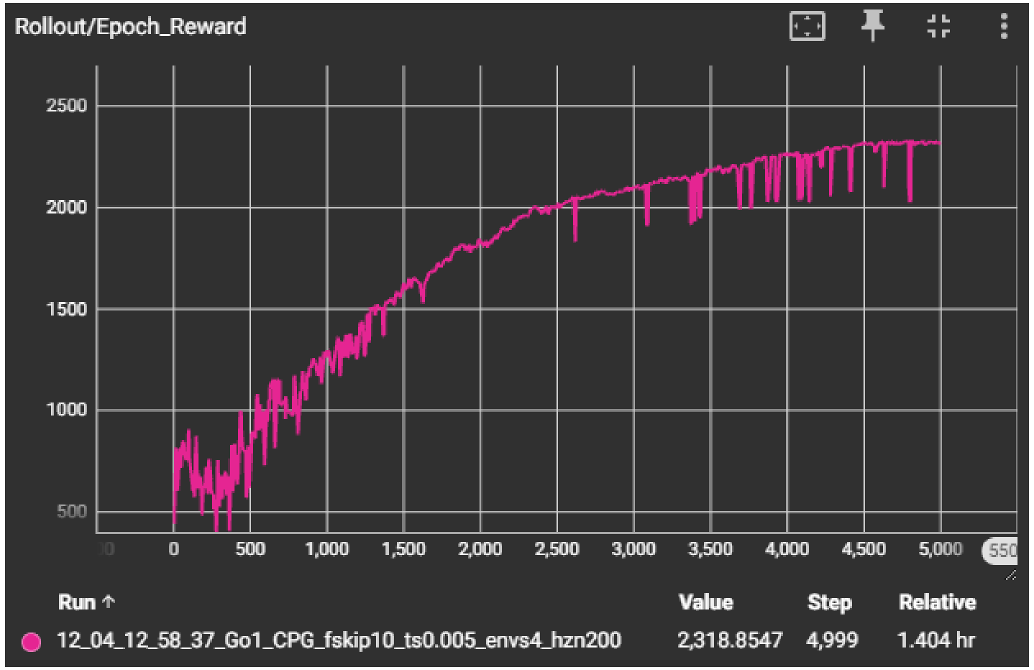
- Weird, unnatural and jerky motion, inability to move forward.
- The robot only learns to avoid falling flat on the ground and terminating the episode.

Proposed method: TRPO with Gait Priors

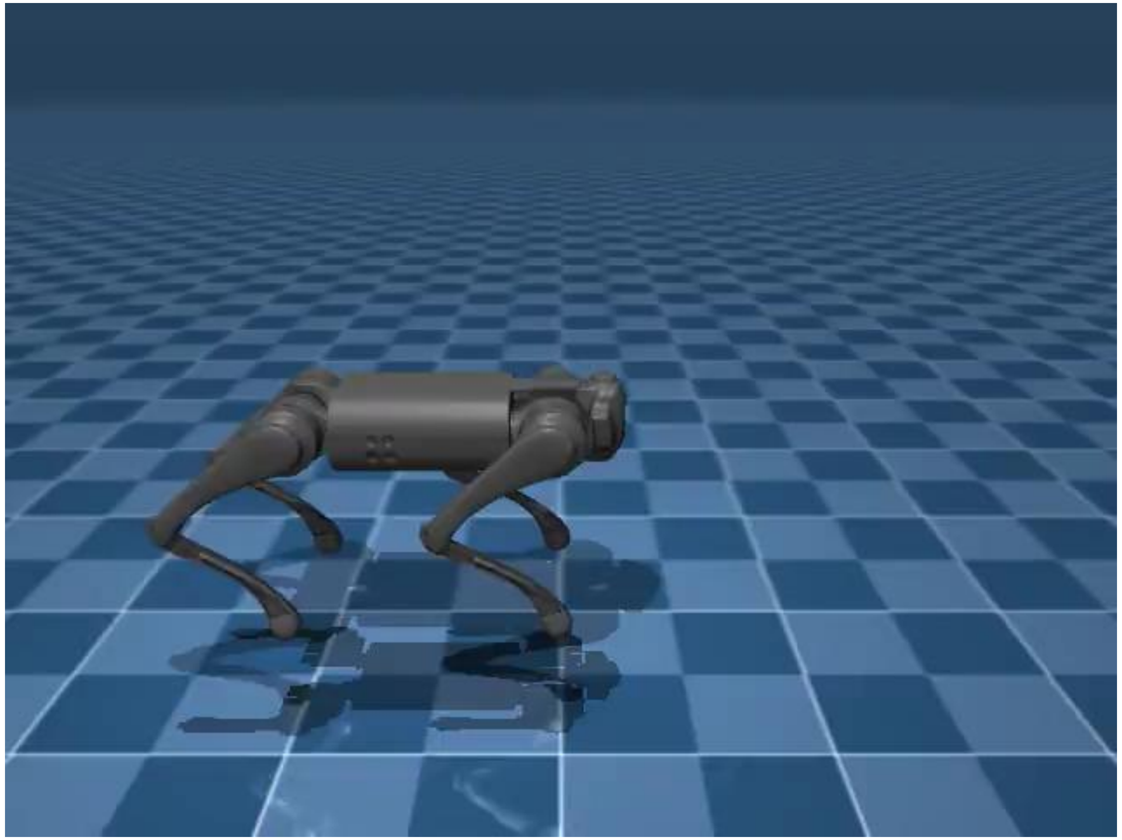
- Learning locomotion without any prior knowledge is hard!
- Instead, we can introduce a *gait prior* which is heuristic-designed for a quadrupedal robot consisting of abduction, hip and knee joints.
- I use a prior - “periodic signal” for a trotting gait such that the front left (FL) and rear right (RR) joints are in phase while the front right (FR) and rear left (RL) are 180 deg out of phase.
- The prior signal acts as a base **feed-forward** controller operating at a frequency of 1 Hz.
- Whereas a neural network policy is trained with **TRPO** to act as a *residual feedback* controller.
- The residual policy operates at 20 Hz. The simulation frequency is 200 Hz.



Final Results



Run ↑	Value	Step	Relative
● 12_04_12_58_37_Go1_CPG_fskip10_ts0.005_envs4_hzn200	2,318.8547	4,999	1.404 hr



- Learning curve – policy converges to the maximum possible reward.
- Play video – symmetric, natural and periodic gait.
- The robot learns to move forward!

Thank You for listening!