Walk These Ways: Learning Diverse Gaits for an Uncertain Future

Gabriel B. Margolis **Pulkit Agrawal**
Improbable AI Lab, Massachusetts Institute of Technology
{gmargo, pulkitag}@mit.edu

Figure 1: Diverse gaits enable a human to pilot a quadruped expressively and perform online tuning to tasks that were not anticipated at training time. A single policy embodies a large continuous space of distinct motion styles useful for downstream tasks like high-speed running, stair traversal, perturbation resistance, and payload manipulation. Transition between behaviors accomplishes compound tasks such as diverse outdoor terrain traversal, agile running leaps, and rhythmic dance.

Abstract: We investigate the utility of structuring learned quadrupedal gaits with dynamically specified gait parameters. We present a fast and robust locomotion controller that can trot, pronk, pace, and bound with variable footswing, posture, and speed. Modulating these parameters at runtime enables a human operator or high-level planner to execute varied sequences of agile behavior useful for downstream tasks. A single policy realizes crouching, hopping, high-speed running, agile leaps, and rhythmic dance. Our analyses suggest new perspectives on the role of gait in learned locomotion. Video is available at https://sites.google.com/view/gait-conditioned-rl/.

Keywords: Locomotion, Reinforcement Learning, Task Specification

1 Introduction

Much work on learning quadrupedal locomotion targets a static locomotion style objective, such as robustness to disturbances [1, 2, 3], energy efficiency [4], or similarity to a particular reference motion [5, 6, 7]. While successful, these approaches offer the user meager control over the robot's locomotion style during deployment. In practical use cases, the end user may prefer to have a choice among a large number of locomotion styles in the hope that some will suit their needs at any given moment. The user's needs might range from interaction with unanticipated environments to the generation of stylish motions for entertainment.

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

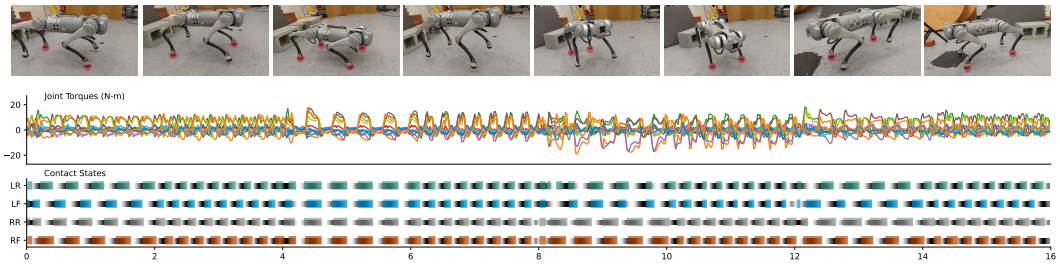

Figure 2: Transition between trotting, pronking, pacing, and bounding in place at alternating frequencies of 2Hz and 4Hz. Images show the robot achieving the contact phases of each gait, with stance feet highlighted in red. Black shading in the bottom plot reflects the timing reference variables $\mathbf{t}_t$ for each foot; colored bars report the contact states measured by foot sensors.

One way to place more control in the hands of the user is to provide a set of flexible and intuitive motion primitives that they can compose to produce the locomotion behavior of their preferred style or which suits their downstream task. To this end, we implement and evaluate a task specification language for learned quadrupedal locomotion which enables the online composition of motion sequences combining diverse gaits, postures, and speeds. Such language is common in the setting of analytical control, where it is often necessary to define constraints, objectives, and models [8, 9, 10]. It is often avoided in reinforcement learning, where it is possible for motion styles to emerge from a reward function without direct specification [1, 4, 11, 12].

We find that a single *gait-conditioned policy* can execute diverse and controllable motion styles while retaining a high degree of robustness against variations in robot and terrain dynamics. Crucially, our method not only learns to locomote with diverse contact schedules, body postures, and foot postures, but also learns to transition between them online. This enables diverse real-time adaptive behaviors useful for diverse downstream tasks, including navigation through confined spaces, agile forward leaps longer than the robot's body length, and the execution of dynamic choreographed dance. Additional case studies show how dynamic gait adaptation can enable fast obstacle traversal, object manipulation, and robustness to shoves. The combinatorial diversity of behaviors we demonstrate in a single locomotion policy goes beyond previous results in multi-task learned locomotion. This strong base policy invites future work to build high-level controllers for diverse tasks. Our method employs shaped reward, but it is simple and extensible compared to other works employing discrete task-specific heuristics or reference trajectory libraries [7, 13, 14, 15].

Practitioners of learning-based legged locomotion have reported limits on the behavioral expressiveness of a single policy, and hypothesized that specialized architectures or learning procedures would be necessary [1, 4, 14, 16]. We demonstrate that given appropriate design choices, RL yields extremely diverse controllable quadrupedal gaits in a small, fully-connected policy network. We show comprehensive real-world validation of our claims in the setting of robust high-speed locomotion.

2 Related Work

Most previous work on learning locomotion addresses the task of achieving a target body velocity [1, 2, 3, 11, 13, 17, 16, 12, 18]. In these works, the operator provides the target velocity as input to a control policy, and the policy coordinates leg motions to track that velocity as closely as possible. There are many ways a legged robot can move in order to achieve the same target velocity; it may select different foot contact patterns or carry its body in different poses while progressing forward at the same speed. For this reason, we claim that the velocity tracking task is generally *underspecified*. In response, several forms of specification have been used in the prior literature:

Auxiliary Rewards. Some prior works overcome underspecification by defining auxiliary rewards. For example, penalties on energy expenditure [4, 19], body motion [20], or deviation from a nominal pose [18] may distinguish one locomotion gait as preferred over another. In theory, an objective such

as energy minimization may cause substantially different gaits to be preferred at different speeds. In practice, all works that learn joint-space policies with auxiliary rewards have ended up learning relatively static contact patterns and body motion [3, 16, 12, 18]. Fu et al. [4] notably reported that different gaits emerge at different speeds when applying an auxiliary energy minimization reward, but they achieved this by training a separate policy at each speed and combining them through a manual distillation step. Other works learned a high-level policy to modulate the gait of a low-level controller to achieve energy minimization [19] or vision-guided gap crossing [20, 21]. All opted for a model-predictive low-level controller to execute the different gaits; a learned controller with suitable gait parameters did not yet exist. Our work demonstrates such a controller and will enable revisiting these hierarchical frameworks with learning at both levels.

Imitation Learning. Another line of work uses reference trajectories to specify robot motions [5, 6, 7, 22, 23, 15]. Some approaches use an imitation objective to reward states similar to the reference motion [5, 15, 23], while others use adversarial techniques to reward stylistically similar movements [6, 7]. Most imitation learning approaches learn only one style of behavior per policy [5, 6]. Vollenweider et al. [7] provided a recent demonstration that a small number of discrete motion styles can be learned simultaneously. Shao et al. [15] notably generated a library of reference trajectories and trained a goal-conditioned policy to imitate them. However, using a library of detailed reference trajectories penalizes the agent for exploring behaviors that are difficult to procedurally generate. We prefer to specify the gait in a less explicit way, to allow the policy to discover complex strategies such as dynamic body motion, slip recovery, and resistance to perturbation.

Reward-based Task Specification. The prior works most closely related to ours explicitly include gait parameters in the task specification through additional reward terms, including tracking rewards for contact patterns [24] and target foot placements [25]. Siekmann et al. [24] used gait-dependent reward terms to penalize foot motion during stance and contact force during swing phases. Compared to quadrupeds, a biped has fewer feet and its body has fewer practical degrees of freedom. As such, the multi-gait policy in [24] has few parameters: the timing offset between the two feet and the duration of the swing phase. The authors also reported difficulty eliciting multiple well-formed gaits from one policy, and implemented additional reward shaping to achieve specific gaits.

3 Method

3.1 Gait Specification

We parameterize the common quadrupedal gaits by an 11-dimensional command vector,

$$\mathbf{c}_t = [\mathbf{v}_x^{\text{cmd}}, \mathbf{v}_y^{\text{cmd}}, \boldsymbol{\omega}_z^{\text{cmd}}, \boldsymbol{\theta}_1^{\text{cmd}}, \boldsymbol{\theta}_2^{\text{cmd}}, \boldsymbol{\theta}_3^{\text{cmd}}, \boldsymbol{f}^{\text{cmd}}, \boldsymbol{h}_z^{\text{cmd}}, \boldsymbol{\phi}^{\text{cmd}}, \boldsymbol{s}_y^{\text{cmd}}, \boldsymbol{h}_z^{f\,\text{cmd}}],$$

- $\mathbf{v}_x^{\text{cmd}}, \mathbf{v}_y^{\text{cmd}}, \boldsymbol{\omega}_z^{\text{cmd}}$ are the linear velocities in the body-frame x- and y- axes, and the angular velocity in the yaw axis.
- $\boldsymbol{\theta}^{\text{cmd}} = (\boldsymbol{\theta}_1^{\text{cmd}}, \boldsymbol{\theta}_2^{\text{cmd}}, \boldsymbol{\theta}_3^{\text{cmd}})$ are the timing offsets between pairs of feet. These express gaits including pronking ($\boldsymbol{\theta}^{\text{cmd}} = (0.0, 0, 0)$), trotting ($\boldsymbol{\theta}^{\text{cmd}} = (0.5, 0, 0)$), bounding, ($\boldsymbol{\theta}^{\text{cmd}} = (0, 0.5, 0)$), pacing ($\boldsymbol{\theta}^{\text{cmd}} = (0, 0, 0.5)$), as well as their continuous interpolations such as galloping ($\boldsymbol{\theta}^{\text{cmd}} = (0.25, 0, 0)$). Taken together, the parameters $\boldsymbol{\theta}^{\text{cmd}}$ can express all two-beat quadrupedal contact patterns. Figure 2 provides a visual illustration.
- $\boldsymbol{f}^{\text{cmd}}$ is the stepping frequency expressed in Hz. Commanding $\boldsymbol{f}^{\text{cmd}} = 3\,\text{Hz}$ will result in each foot making contact three times per second.
- $\boldsymbol{h}_z^{\text{cmd}}$ is the body height command; $\boldsymbol{\phi}^{\text{cmd}}$ is the body pitch command.
- $\boldsymbol{s}_y^{\text{cmd}}$ is the foot stance width command; $\boldsymbol{h}_z^{f\,\text{cmd}}$ is the footswing height command.

Reward function. All reward terms are given in Table 1. Task rewards are defined as functions of the command vector $\mathbf{c}_t$ for body velocity tracking, body posture tracking, foot posture tracking, and contact schedule tracking. Body velocity, body posture, and foot posture rewards are simple

Term	Equation	Weight
$r_{v_{x,y}^{\text{cmd}}}$: xy velocity tracking	$\exp\{-\lvert\mathbf{v}_{xy} - \mathbf{v}_{xy}^{\text{cmd}}\rvert^2/\sigma_{vxy}\}$	0.02
$r_{\omega_z^{\text{cmd}}}$: yaw velocity tracking	$\exp\{-(\boldsymbol{\omega}_z - \boldsymbol{\omega}_z^{\text{cmd}})^2/\sigma_{\omega z}\}$	0.01
$r_{c_f^{\text{cmd}}}$: swing phase tracking (force)	$\sum_{\text{foot}}[1 - C_{\text{foot}}^{\text{cmd}}(\boldsymbol{\theta}^{\text{cmd}}, t)] \exp\{-\lvert\mathbf{f}^{\text{foot}}\rvert^2/\sigma_{cf}\}$	−0.08
$r_{c_v^{\text{cmd}}}$: stance phase tracking (velocity)	$\sum_{\text{foot}}[C_{\text{foot}}^{\text{cmd}}(\boldsymbol{\theta}^{\text{cmd}}, t)] \exp\{-\lvert\mathbf{v}_{xy}^{\text{foot}}\rvert^2/\sigma_{cv}\}$	−0.08
$r_{\boldsymbol{h}_z^{\text{cmd}}}$: body height tracking	$(\boldsymbol{h_z} - \boldsymbol{h}_z^{\text{cmd}})^2$	−0.2
$r_{\boldsymbol{\phi}^{\text{cmd}}}$: body pitch tracking	$(\boldsymbol{\phi} - \boldsymbol{\phi}^{\text{cmd}})^2$	−0.08
$r_{\boldsymbol{s}_y^{\text{cmd}}}$: raibert heuristic footswing tracking	$(\boldsymbol{s_y} - \boldsymbol{s}_y^{\text{cmd}})^2$	−0.08
$r_{\boldsymbol{h}_z^{f\,\text{cmd}}}$: footswing height tracking	$\sum_{\text{foot}}(\boldsymbol{h}_{z,\text{foot}}^{f} - \boldsymbol{h}_z^{f\,\text{cmd}})^2 C_{\text{foot}}^{\text{cmd}}(\boldsymbol{\theta}^{\text{cmd}}, t)$	−0.08
z velocity	$\mathbf{v}_z^2$	$-4\text{e}-4$
roll-pitch velocity	$\lvert\boldsymbol{\omega}_{xy}\rvert^2$	$-2\text{e}-5$
foot slip	$\lvert\mathbf{v}_{xy}^{\text{foot}}\rvert^2$	$-8\text{e}-4$
thigh/calf collision	$\mathbb{1}_{\text{collision}}$	−0.02
joint limit violation	$\mathbb{1}_{q_i > q_{max} \lvert q_i \leq q_{min}}$	−0.2
joint torques	$\lvert\boldsymbol{\tau}\rvert^2$	$-2\text{e}-5$
joint positions	$\lvert\mathbf{q} - \hat{\mathbf{q}}\rvert^2$	$-1\text{e}-3$
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	$\lvert\mathbf{a}_{t-2} - 2\mathbf{a}_{t-1} + \mathbf{a}_t\rvert^2$	$-2\text{e}-3$

Task

Stability

Smoothness

Table 1: Reward terms for task, stability, and smoothness.

functions of the squared error, weighted to account for the relative error magnitude. To define the desired contact schedule, function $C_{\text{foot}}^{\text{cmd}}(\boldsymbol{\theta}^{\text{cmd}}, t)$ computes the desired contact state of each foot from the phase and timing variable, as described in [24], with details given in the appendix. Stability and smoothness rewards express auxiliary locomotion objectives that are desired across all gaits. As in [18], we force the total reward to be positive by computing it as $r_{\text{pos}} \exp(0.02 r_{\text{neg}})$ where r_{pos} is the sum of all positive reward terms and r_{neg} is the sum of all negative reward terms.

To demonstrate that our method is not detrimentally sensitive to reward tuning, we first tune the hyperparameters (reward weights, optimizer parameters) for a policy with no rewards for body pitch, stance width, or footswing height control ($\boldsymbol{\phi}^{\text{cmd}}, \boldsymbol{s}_y^{\text{cmd}}, \boldsymbol{h}_z^{f\,\text{cmd}}$). Then we introduce the rewards for ($\boldsymbol{\phi}^{\text{cmd}}, \boldsymbol{s}_y^{\text{cmd}}, \boldsymbol{h}_z^{f\,\text{cmd}}$) and tune only their weights. The original reward terms and weights do not need any re-tuning to add new gait parameters and capabilities to our proposed framework.

3.2 Learning Quadrupedal Gaits

We implement our training environment in the Isaac Gym simulator [26] and train our locomotion policy using Proximal Policy Optimization [27].

Observation Space. The policy observes history $\mathbf{o}_{t-H\ldots t}$ where $\mathbf{o}_t$ consists of command $\mathbf{c}_t$, previous action $\mathbf{a}_{t-1}$, sensor data $\mathbf{s}_t$, and timing reference variables $\mathbf{t}_t$. The sensor data $\mathbf{s}_t$ includes joint positions and velocities $\mathbf{q}_t, \dot{\mathbf{q}}_t$ (measured by joint encoders) and the gravity vector in the body frame $\mathbf{g}_t$ (measured by accelerometer). Timing reference variables $\mathbf{t}_t$ are sine functions aligned to the contact offset of each foot:

$$[t^{\text{FR}}, t^{\text{FL}}, t^{\text{RR}}, t^{\text{RL}}] = [t + \boldsymbol{\theta}_2^{\text{cmd}} + \boldsymbol{\theta}_3^{\text{cmd}}, t + \boldsymbol{\theta}_1^{\text{cmd}} + \boldsymbol{\theta}_3^{\text{cmd}}, t + \boldsymbol{\theta}_1^{\text{cmd}}, t + \boldsymbol{\theta}_2^{\text{cmd}}]$$
$$[\mathbf{t}_t^{\text{FR}}, \mathbf{t}_t^{\text{FL}}, \mathbf{t}_t^{\text{RR}}, \mathbf{t}_t^{\text{RL}}] = [\sin(2\pi t^{\text{FR}}), \sin(2\pi t^{\text{FL}}), \sin(2\pi t^{\text{RR}}), \sin(2\pi t^{\text{RL}})]$$

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. These timing variables are adapted from [24] to represent the quadrupedal gaits.

Action Space. The action $\mathbf{a}_t$ consists of position targets for each of the twelve joints. A zero action corresponds to the nominal joint position, $\hat{\mathbf{q}}$. The position targets are tracked using a proportional-derivative controller with $k_p = 20, k_d = 0.5$.

Gait	$0.0\,\mathrm{m/s}$	$1.0\,\mathrm{m/s}$	$2.0\,\mathrm{m/s}$	$3.0\,\mathrm{m/s}$
Trotting	$9_{\pm 1}$	$24_{\pm 1}$	$53_{\pm 5}$	$98_{\pm 9}$
Pronking	$32_{\pm 1}$	$43_{\pm 2}$	$68_{\pm 5}$	$112_{\pm 5}$
Pacing	$13_{\pm 3}$	$25_{\pm 2}$	$55_{\pm 3}$	$99_{\pm 6}$
Bounding	$22_{\pm 2}$	$39_{\pm 4}$	$78_{\pm 5}$	$127_{\pm 35}$
Gait-free Baseline	$17_{\pm 5}$	$35_{\pm 5}$	$64_{\pm 10}$	$102_{\pm 14}$
Trotting ($\boldsymbol{f}^{\mathrm{cmd}} = 2\,\mathrm{Hz}$)	$11_{\pm 2}$	$25_{\pm 1}$	$55_{\pm 4}$	$104_{\pm 8}$
Trotting ($\boldsymbol{f}^{\mathrm{cmd}} = 3\,\mathrm{Hz}$)	$9_{\pm 1}$	$24_{\pm 1}$	$53_{\pm 5}$	$98_{\pm 9}$
Trotting ($\boldsymbol{f}^{\mathrm{cmd}} = 4\,\mathrm{Hz}$)	$9_{\pm 1}$	$26_{\pm 0}$	$60_{\pm 4}$	$114_{\pm 12}$
Trotting ($\boldsymbol{h}_z^{\mathrm{cmd}} = 20\,\mathrm{cm}$)	$9_{\pm 1}$	$26_{\pm 1}$	$56_{\pm 3}$	$102_{\pm 8}$
Trotting ($\boldsymbol{h}_z^{\mathrm{cmd}} = 30\,\mathrm{cm}$)	$9_{\pm 1}$	$24_{\pm 1}$	$53_{\pm 5}$	$98_{\pm 9}$
Trotting ($\boldsymbol{h}_z^{\mathrm{cmd}} = 40\,\mathrm{cm}$)	$10_{\pm 1}$	$23_{\pm 1}$	$52_{\pm 4}$	$95_{\pm 9}$

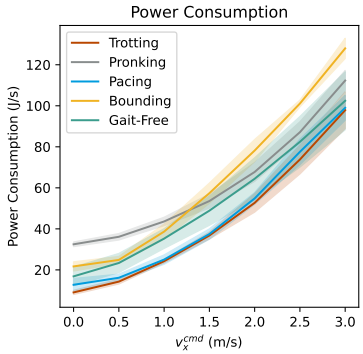

Table 3: Gait-conditioned policies enable interventional studies on the relationship between gait properties and performance criteria within a single policy. Here, we illustrate how power consumption ($\mathrm{J/s}$) varies across speeds for common quadrupedal gaits and for a baseline policy without gait constraint. While efficiency varies across gaits and speeds, trotting and pacing gaits achieve efficiency competitive with unconstrained gait across all speeds.

Domain Randomization. We randomize the robot's body mass, motor strength, and joint position calibration, the friction and restitution of the terrain, and the orientation and magnitude of gravity. This randomization facilitates sim-to-real transfer and makes the controller robust to different terrain geometries and material properties. All randomization ranges are given in Appendix A.

Policy Architecture. We apply the Concurrent State Estimation architecture proposed by [18]. The "estimator module", a feedforward neural network, is trained using supervised learning to predict the mass, velocity, and height of the robot body, the friction and restitution of the ground, and the gravity vector. The input to this estimator module is a 15-step history of observations. The estimator output is concatenated with the state history as input to the policy body. The policy body is an MLP with hidden layer sizes $[512, 256, 128]$; the estimator module is an MLP with hidden layer sizes $[256, 128]$; both use ELU activations.

Task Curriculum. To enable the robot to both run and spin fast, we sample velocity commands using the grid adaptive curriculum strategy from [12]. A discrete grid tabulates successful combinations of linear and angular velocity thus far. Velocity commands for the policy are sampled from a small neighborhood of the successful region. We grow a separate grid adaptive curriculum for each of the four major gaits: pronking, trotting, bounding, and pacing.

The procedure for sampling a command $\mathbf{c}_t$ is as follows: First, one of the four major gaits (pronking, trotting, bounding, or pacing) is selected with equal probability. Then, a candidate velocity command $(\mathbf{v}_x^{\mathrm{cmd}}, \boldsymbol{\omega}_z^{\mathrm{cmd}})$ for the chosen gait is sampled following [12]. To let the policy learn to interpolate between major gaits, $(\boldsymbol{\theta}_1^{\mathrm{cmd}}, \boldsymbol{\theta}_2^{\mathrm{cmd}}, \boldsymbol{\theta}_3^{\mathrm{cmd}})$ are sampled from a Gaussian distribution that is centered at the chosen major gait. The remaining command parameters $(\mathbf{v}_y^{\mathrm{cmd}}, \boldsymbol{f}^{\mathrm{cmd}}, \boldsymbol{h}_z^{\mathrm{cmd}}, \boldsymbol{\phi}^{\mathrm{cmd}}, \boldsymbol{h}_z^{f\,\mathrm{cmd}}, \boldsymbol{s}_y^{\mathrm{cmd}})$ are sampled independently and uniformly. Their ranges are given in Table 6.

3.3 Deployment

We deploy our controller in the real world on the Unitree Go1 Edu robot [28]. The Go1 is a commercially available and relatively low-cost robot quadruped.

Computing Architecture: An onboard Jetson TX2 NX computer runs our trained policy. We implement an interface based on Lightweight Communications and Marshalling (LCM) [29] to pass sensor data, motor commands, and joystick state between our code and the low-level control SDK provided by Unitree. For both training and deployment, the control frequency is 50Hz.

Sensors: In addition to joint position encoders and an inertial measurement unit, the robot is equipped with foot contact sensors. We log the detected contact states for our analysis, but they are not provided to our controller as we did not find that they improve performance.

Gait	$r_{v_{x,y}^{\text{cmd}}}$	$r_{\omega_z^{\text{cmd}}}$	$r_{c_f^{\text{cmd}}}$	$r_{c_v^{\text{cmd}}}$	Survival
Trotting	$0.80^{(0.95)}_{\pm 0.01}$	$0.76^{(0.89)}_{\pm 0.00}$	$0.95^{(0.97)}_{\pm 0.00}$	$0.98^{(0.98)}_{\pm 0.00}$	$0.88^{(1.00)}_{\pm 0.01}$
Pronking	$0.84^{(0.94)}_{\pm 0.01}$	$0.77^{(0.85)}_{\pm 0.01}$	$0.96^{(0.96)}_{\pm 0.00}$	$0.97^{(0.98)}_{\pm 0.00}$	$0.82^{(1.00)}_{\pm 0.02}$
Pacing	$0.76^{(0.91)}_{\pm 0.01}$	$0.76^{(0.81)}_{\pm 0.01}$	$0.94^{(0.96)}_{\pm 0.00}$	$0.98^{(0.98)}_{\pm 0.00}$	$0.87^{(1.00)}_{\pm 0.02}$
Bounding	$0.80^{(0.88)}_{\pm 0.01}$	$0.73^{(0.86)}_{\pm 0.01}$	$0.94^{(0.96)}_{\pm 0.00}$	$0.98^{(0.98)}_{\pm 0.00}$	$0.82^{(1.00)}_{\pm 0.01}$
Gait-free	$0.81^{(0.96)}_{\pm 0.03}$	$0.74^{(0.92)}_{\pm 0.06}$	–	–	$0.83^{(1.00)}_{\pm 0.01}$

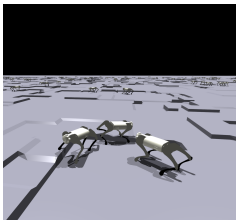

Table 4: Zero-shot robustness evaluation on platform terrain (visualized right). Pacing and trotting gaits yield the best survival time in zero-shot deployment, outperforming the gait-free baseline. Pronking attains the best velocity tracking performance, with similar survival time to the baseline. We report the fraction of maximum episodic reward. Superscript reports performance in flat training environment with no platforms. Subscript reports standard deviation across three random seeds.

4 Experimental Results

4.1 Learning High-Speed Quadrupedal Gaits

Our method enables a single learned controller to run and spin using all common quadrupedal gaits. Trotting is learned faster, followed by pronking, then pacing and bounding. The high learnability of trotting aligns with its tendency to emerge from a variety of non-gait-specific auxiliary rewards [2, 11, 18, 20]. However, because our task sampling strategy provides explicit incentive to achieve all gaits, we find that fast and robust control strategies for pacing, bounding, and pronking emerge as well. Details about the relative learning progression of gaits can be found in the appendix.

4.2 Sim-to-Real Transfer and Gait Switching

We deploy our controller in zero-shot sim-to-real manner using the robot and software architecture described in Section 3.3. We find that all gait parameters are correctly tracked after sim-to-real transfer. Figure 2 shows torques and contact states during transition between trotting, pronking, bounding, and pacing in place while alternating $\boldsymbol{f}^{\text{cmd}}$ between $2\,\text{Hz}$ and $4\,\text{Hz}$. In the bottom subfigure, good alignment between the colored rectangles, representing true contacts, and the dark shaded regions, representing desired contacts, reflects successful tracking across gaits and frequencies.

4.3 Impact of Gait Specification on Locomotion Performance

We probe the gait parameters of our trained model to quantify their role in quadrupedal locomotion performance. We compare our gait-conditioned controller to a baseline velocity-tracking controller (the "gait-free baseline"). The gait-free baseline is trained by the same method as our gait-conditioned controller, but without task reward terms for contact timing or body posture. We aim to illustrate that varied styles of locomotion can have different performance characteristics on new tasks, and that by training a model that can achieve multiple styles, we grant the user a useful degree of freedom to exploit for any new task without task-specific retraining.

Energy Efficiency. We measure the power consumption (J/s) of symmetric quadrupedal gaits and the gait-free baseline. We consider mechanical power consumption measured by summing the product of joint velocity and torque at each of the 12 motors: $\sum_i \max(\boldsymbol{\tau}_i \dot{\mathbf{q}}_i, 0)$. Table 3 reports the power consumption profile for each gait. The robot expends less energy while trotting or pacing than while pronking or bounding. Fixed trotting and pacing achieve equal or lower energy consumption compared to the gait-free baseline at low and medium speeds. This suggests that policies trained using standard reward structure on flat ground are not achieving a gain in energy efficiency via uncommon adaptation of contact pattern, gait frequency, or body posture.

Robustness. We evaluate quadrupedal gaits on a non-flat test terrain consisting of randomly arranged platforms up to 8cm in height (the "platform terrain", pictured beside Table 4). All policies in this work are trained only on flat ground, so the platform terrain is out-of-distribution. In Table

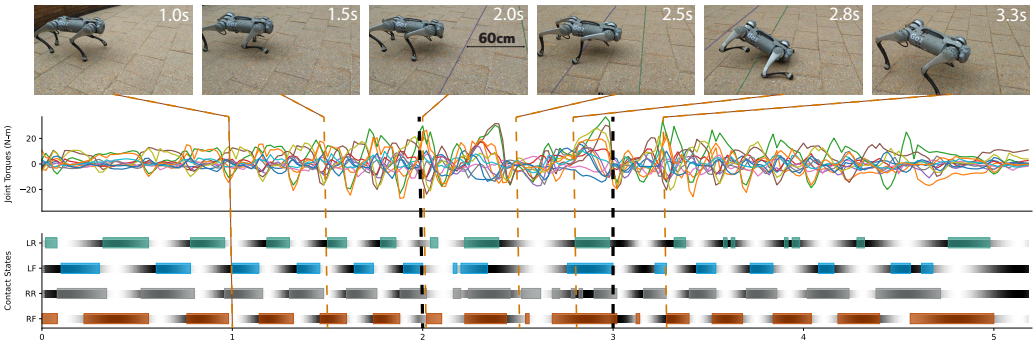

Figure 3: Gait parameter modulation supports the downstream task of crossing a gap wider than the robot's body length with a single leap.

4, we report the mean task reward on the platform terrain for each of the major gaits and the gait-free baseline. We also report the survival time as the mean fraction of the maximum episode length ($10\,\mathrm{s}$) before failure. The episode was terminated if the agent did not fall after 10 seconds. In this experiment, velocity commands were limited to $\mathbf{v}_x^{\mathrm{cmd}} \in [-2, 2]$, $\boldsymbol{\omega}_z^{\mathrm{cmd}} \in [-2, 2]$.

Different gaits yield different robustness characteristics on the platform terrain. Pacing has the highest survival time, higher than the gait-free policy, which indicates that it was able to remain stable on some terrains where the emergent gait on flat ground falls. However, pacing tracked the target velocity less accurately than other gaits. Pronking has the closest characteristics to the gait-free policy in terms of velocity tracking, and with a similar survival time.

In the real world, we demonstrate robustness with different gaits while traversing uneven, slippery, and granular terrain as well as stairs. The robot can traverse down stairs with a variety of gaits, but not up. Video showing a number of real world deployment scenarios is at the project website.

Velocity Tracking and Top Speed. We report velocity tracking performance of multiple 3Hz gaits and the gait-free baseline across a wide range of speeds including high-speed sprinting in Table 5. Removing gait constraints results in improved velocity tracking task performance on flat ground. Heat maps break down the mean task reward for each velocity, revealing that the gait-free approach does not help at low speeds, but is better for combinations of high linear and angular velocity.

4.4 Case Studies: Downstream Utility of Gait Modulation

Navigating confined spaces. Different body postures and foot stances are useful for navigating confined spaces with limited width or height. We demonstrate crawling beneath an increasingly lowered bar. The robot was able to crawl under a $22\,\mathrm{cm}$ bar, and the robot body thickness is $13\,\mathrm{cm}$, leaving maximum $9\,\mathrm{cm}$ of clearance between the robot and the ground.

Agile forward leap. Modulation of contact schedule, velocity, and gait frequency at high speed encode an agile forward leap (Figure 3). The robot first accelerates to a target speed of $3\,\mathrm{m/s}$ at a trot while increasing its step frequency from $2\,\mathrm{Hz}$ to $4\,\mathrm{Hz}$, then switches to pronking at $2\,\mathrm{Hz}$ for one second, then decelerates to a standstill while trotting. During the leap phase, the distance from the location of the robot's front feet at takeoff to the location of the hind feet upon landing is $60\,\mathrm{cm}$.

Choreographed dance. We program a sequence of gait parameters to generate a dance routine synchronized to a jazz song with a tempo of 90 bpm. At this tempo, combinations of phases 0, 0.25, and 0.5 with frequencies of $1.5\,\mathrm{Hz}$ and $3\,\mathrm{Hz}$ yield eighth, quarter, half, and full beat gaps between consecutive footsteps. We also modulate body height and velocity in time with the music. An assistant script procedurally generates gait parameters, fed into controller in open-loop fashion.

Climbing over curbs. Modulating footswing height enables our controller to climb smoothly across 12-cm elevated platforms (Figure 1, left). Previous works demonstrating blind obstacle traversal on

Ablation	$r_{v_{x,y}^{\text{cmd}}}$	$r_{\omega_z^{\text{cmd}}}$	$r_{c_f^{\text{cmd}}}$	$r_{c_v^{\text{cmd}}}$
Trotting	$0.92_{\pm 0.01}$	$0.70_{\pm 0.04}$	$0.98_{\pm 0.00}$	$0.95_{\pm 0.00}$
Pronking	$0.85_{\pm 0.01}$	$0.64_{\pm 0.05}$	$0.98_{\pm 0.00}$	$0.94_{\pm 0.00}$
Pacing	$0.83_{\pm 0.02}$	$0.66_{\pm 0.04}$	$0.96_{\pm 0.01}$	$0.94_{\pm 0.01}$
Bounding	$0.88_{\pm 0.01}$	$0.63_{\pm 0.05}$	$0.96_{\pm 0.01}$	$0.95_{\pm 0.00}$
Gait-free Baseline	$0.94_{\pm 0.02}$	$0.76_{\pm 0.01}$	–	–

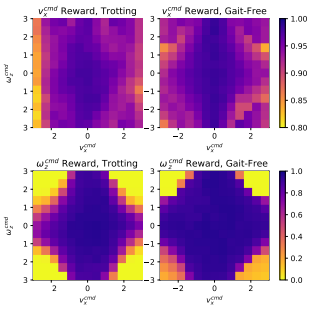

Table 5: Removing gait constraints results in improved velocity tracking task performance on flat ground. Heat maps (right) break down the mean task reward for each velocity command, revealing that the gait-free approach is most beneficial for combinations of high linear and angular velocity.

a quadruped [1, 2] learned a foot-trapping reflex where the robot first trips, then raises its foot over the obstacle. In contrast, our gait-conditioned policy with high footswing command enables fast and smooth obstacle traversal without tripping, and is trained on a simpler flat-ground terrain.

Payload manipulation. Body posture control enables manipulation of objects. In one example, the robot carries a ball in a payload box tied to its back. The operator pilots the robot to the delivery location, then pitches the body backward to dump its payload. Next, the robot runs to a resting ball and moves it into a goal by ducking and pushing with its shoulders (Figure 1, second left). In-motion posture control is also useful when a mounted camera needs to be aimed at a high or low target.

Bracing against shoves. Learned locomotion controllers often keep the feet nominally below the hips as incentivized by auxiliary rewards [2, 18]. Widening the stance can make a quadruped more robust to perturbations, while narrowing it can facilitate better running efficiency or navigation of tight areas. We shove the robot while modulating between wide and narrow stance. The robot remains upright in both cases, but its body motion is significantly less in the wide stance.

5 Limitations

Specifying the locomotion task reduces the system's degree of autonomy, sacrificing some task performance. Our experiments show this hinders the robot's flat-ground sprinting performance (Section 4.3; Table 5), suggesting the best sprinting departs from our parameterized family of gaits. Constraining the gait may also lead to suboptimality in desiderata besides those we analyzed, such as wear on the robot, footstep noise, or foot dragging. To further understand the effect of locomotion task parameterization on implicit or explicit task criteria is an open direction for future work.

Some gaits suffer from performance degradation in the real world. For example, in the agile leap sequence, the robot's body sinks lower than the command height during the leap. We also note that when assuming an unstable gait, such as low-frequency pacing with a narrow stance and high body, the robot tends to drift to one side or stumble.

Our work relies on a human or high-level policy to compose behaviors. Future work can integrate gait planning methods such as [19, 20, 21] to give the system high-level autonomous behaviors.

6 Conclusion

We have implemented and analyzed a novel combination of flexible gait specification and robust training procedure for learning quadrupedal locomotion. The resulting controller was modulated to accomplish a number of previously inaccessible downstream tasks without task-specific retraining. This controller also exhibits a high degree of robustness with multiple gaits on uneven, slippery, and granular terrain, as well as stairs. Quantitative experiments suggest that online modulation of gait parameters can benefit RL policies in out-of-distribution environments.

Acknowledgments

We thank the members of the Improbable AI lab for the helpful discussions and feedback on the paper. We are grateful to MIT Supercloud and the Lincoln Laboratory Supercomputing Center for providing HPC resources. This research was supported by the DARPA Machine Common Sense Program, the MIT-IBM Watson AI Lab, and the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/). This research was also sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

References

[1] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.*, 5(47):eabc5986, Oct. 2020. doi:10.1126/scirobotics.abc5986.

[2] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid motor adaptation for legged robots. In *Proc. Robot.: Sci. and Syst. (RSS)*, Virtual, July 2021. doi:10.48550/arXiv.2107.04034.

[3] T. Miki, J. Lee, J. Hwanbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.*, 7(62):abk2822, Jan. 2022. doi:10.1126/scirobotics.abk2822.

[4] Z. Fu, A. Kumar, J. Malik, and D. Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. In *Proc. Conf. Robot Learn. (CoRL)*, pages 928–937, London, UK, Nov. 2021. doi:10.48550/arXiv.2111.01674.

[5] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

[6] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel. Adversarial motion priors make good substitutes for complex reward functions. *arXiv preprint arXiv:2203.15103*, 2022.

[7] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter. Advanced skills through multiple adversarial motion priors in reinforcement learning. *arXiv preprint arXiv:2203.14912*, 2022.

[8] M. H. Raibert. *Legged Robots That Balance*. MIT press, 1986.

[9] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Auton. Robot.*, 40(3):429–455, July 2015. doi:10.1007/s10514-015-9479-3.

[10] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. *arXiv preprint*, 2019. doi:10.48550/arXiv.1909.06586.

[11] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Proc. Conf. Robot Learn. (CoRL)*, pages 91–100, London, UK, Nov. 2021. doi:10.48550/arXiv.2109.11978.

[12] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *Proc. Robot.: Sci. and Syst. (RSS)*, June 2022.

[13] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. In *Proc. Robot.: Sci. and Syst. (RSS)*, Virtual, July 2021. doi:10.48550/arXiv.2105.08328.

[14] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li. Multi-expert learning of adaptive legged locomotion. *Sci. Robot.*, 5(49):eabb2174, Dec. 2020. doi:10.1126/scirobotics.abb2174.

[15] Y. Shao, Y. Jin, X. Liu, W. He, H. Wang, and W. Yang. Learning free gait transition for quadruped robots via phase-guided controller. *IEEE Robotics and Automation Letters*, 7(2): 1230–1237, 2021.

[16] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Sci. Robot.*, 4(26):aau5872, Jan. 2019. doi:10.1126/scirobotics.aau5872.

[17] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In *Proc. Robot.: Sci. and Syst. (RSS)*, pages 1–9, Pittsburgh, Pennsylvania, USA, June 2018. doi:10.15607/RSS.2018.XIV.010.

[18] G. Ji, J. Mun, H. Kim, and J. Hwangbo. Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robot. Automat. Lett. (RA-L)*, 7 (2):4630 – 4637, Apr. 2022. doi:10.1109/LRA.2022.3151396.

[19] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots. Fast and efficient locomotion via learned gait transitions. In *Proc. Conf. Robot Learn. (CoRL)*, pages 773–783, Nov. 2022. doi:10.48550/arXiv.2104.04644.

[20] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. Kim, and P. Agrawal. Learning to jump from pixels. In *Proc. Conf. Robot Learn. (CoRL)*, pages 1025–1034, London, UK, Nov. 2021. doi:10.48550/arXiv.2110.15344.

[21] W. Yu, D. Jain, A. Escontrela, A. Iscen, P. Xu, E. Coumans, S. Ha, J. Tan, and T. Zhang. Visual-Locomotion: Learning to Walk on Complex Terrains with Vision. In *Proc. Conf. Robot Learn. (CoRL)*, Oct. 2021.

[22] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, and M. van de Panne. Learning locomotion skills for Cassie: Iterative design and sim-to-real. In *Proc. Conf. Robot Learn. (CoRL)*, pages 1–13, Osaka, Japan, Nov. 2020. doi:10.48550/arXiv.1903.09537.

[23] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

[24] J. Siekmann, Y. Godse, A. Fern, and J. Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 7309–7315, Xi'an, China, June 2021. doi:10.1109/ICRA48506.2021.9561814.

[25] H. Duan, A. Malik, J. Dao, A. Saxena, K. Green, J. Siekmann, A. Fern, and J. Hurst. Sim-to-real learning of footstep-constrained bipedal dynamic walking. *arXiv preprint arXiv:2203.07589*, 2022.

[26] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac Gym: High performance GPU-based physics simulation for robot learning. *arXiv preprint*, 2021. doi:10.48550/arXiv.2108.10470.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint*, 2017. doi:10.48550/arXiv.1707.06347.

[28] Unitree Robotics, Go1, 2022, https://www.unitree.com/products/go1, [Online; accessed Jun. 2022].

[29] A. S. Huang, E. Olson, and D. C. Moore. Lcm: Lightweight communications and marshalling. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4057–4062. IEEE, 2010.

Term	Minimum	Maximum	Units
Payload Mass	-1.0	3.0	kg
Motor Strength	90	110	%
Joint Calibration	-0.02	0.02	rad
Ground Friction	0.40	1.00	–
Ground Restitution	0.00	1.00	–
Gravity Offset	-1.0	1.0	$\mathrm{m/s^2}$
$\mathbf{v}_x^{\mathrm{cmd}}$	–	–	$\mathrm{m/s}$
$\mathbf{v}_y^{\mathrm{cmd}}$	-0.6	0.6	$\mathrm{m/s}$
$\boldsymbol{\omega}_z^{\mathrm{cmd}}$	–	–	$\mathrm{m/s}$
$\boldsymbol{f}^{\mathrm{cmd}}$	1.5	4.0	Hz
$\boldsymbol{\theta}_1^{\mathrm{cmd}}, \boldsymbol{\theta}_2^{\mathrm{cmd}}, \boldsymbol{\theta}_3^{\mathrm{cmd}}$	0.0	1.0	–
$\boldsymbol{h}_z^{\mathrm{cmd}}$	0.10	0.45	m
$\boldsymbol{\phi}^{\mathrm{cmd}}$	-0.4	0.4	rad
$\boldsymbol{s}_y^{\mathrm{cmd}}$	0.05	0.45	m
$\boldsymbol{h}_z^{f\,\mathrm{cmd}}$	0.03	0.25	m

Table 6: Randomization ranges for dynamics parameters (top) and commands (bottom) during training. $\mathbf{v}_x^{\mathrm{cmd}}, \boldsymbol{\omega}_z^{\mathrm{cmd}}$ are adapted according to a curriculum.

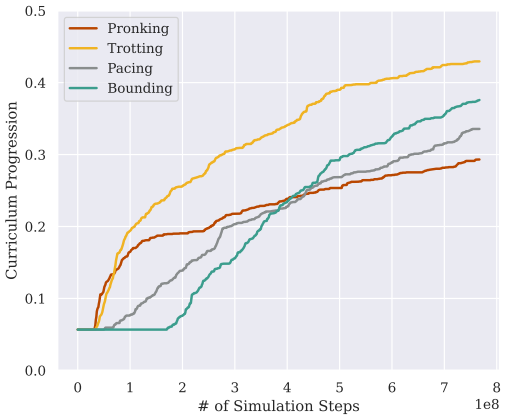

Figure 4: Pronking and trotting gaits are easier to learn and tend to dominate pacing and bounding early in training. However, when discovered, pacing and bounding gaits can yield good performance and later become preferred for some downstream tasks (Section 4.4).

A Training Details

The ranges used for domain and command randomization are provided in Table 6. The hyperparameters used for PPO are provided in Table 7. The hyperparameters used in the curriculum are provided in Table 8.

Figure 6 illustrates data flow during training. The curriculum engine first samples from a Gaussian distribution centered at one of the four main gaits (trotting, pronking, bounding, pacing); then it samples velocity commands from a grid distribution according to the method of [12]; then finally it samples the stepping frequency and body height uniformly. The policy and simulator are rolled out, and the reward is computed as a function of the gait parameters. Then the curriculum is updated if the episodic reward meets the thresholds given in Table 8.

B Teleoperation

Figure 5 illustrates the mapping from remote control inputs to gait parameters used during teleoperation. The front bumpers on the top toggle between control modes to accommodate mixing our large number of gait parameters. Preprogrammed sequences such as dancing and leaping can be assigned to the rear bumpers.

C Extended Performance Analysis

Impact of Gait Frequency on High-speed Running. We evaluate them impact of gait frequency on performance of the robot at high speeds. Figure 10 reports our result that higher gait frequency is necessary to yield good tracking performance for higher-speed running.

Impact of Footswing Height on Platform Terrain Performance. We evaluate them impact of footswing height on performance of the robot on the out-of-distribution platform terrain. Figure 9 reports our result that higher swing heights yield improved platform traversal, outperforming the gait-free policy.

Flat Ground Velocity Heatmaps for More Gaits. We provide velocity heatmaps as in Table 5 for pronking, pacing, and bounding gaits to supplement the trotting and gait-free heatmaps provided in Table 5.

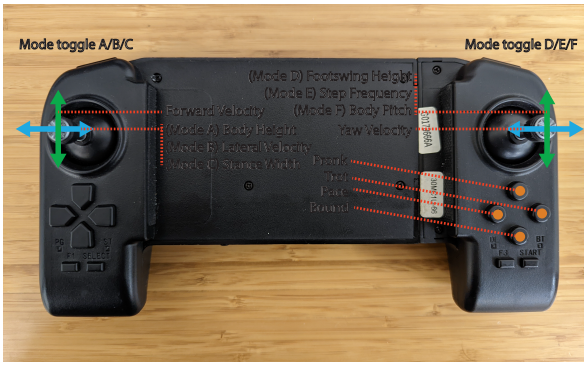

Figure 5: **Controller mapping.** Mapping of remote control inputs to gait parameters during robot teleoperation. The user can change between gaits at any time. Continuous interpolation between contact patterns is supported by our policy, but not mapped here. Lateral velocity is also supported by the controller but excluded from the mapping.

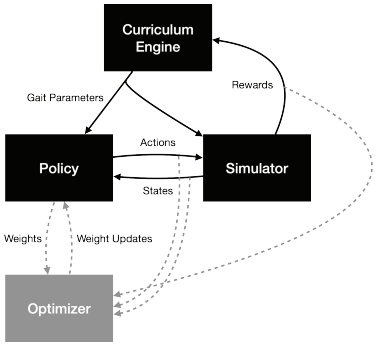

Figure 6: **Training architecture.** The policy computes the action as a function of the gait parameters and state. The simulator computes the reward and state as a function of the gait parameters and actions. The curriculum engine periodically resamples gait parameters based on the reward.

Forward and Backward Locomotion. During evaluation in the random platforms environment, we found that walking backward leads to fewer failures than walking forward. Figure 8 illustrates this phenomenon by plotting the mean failure rate of each gait at each test velocity. Possible explanations include (i) recovery strategies that are dependent on knee orientation and (ii) the weight distribution of the robot.

Real-world Robustness Demonstrations. We conducted several hours of real-world testing of different gaits across a variety of laboratory and outdoor environments. A selection of this footage is available on the project website. The robot was able to traverse down stairs, up and down granular and slippery terrain, and to respond to external perturbations. To provide insight into the robot's disturbance response, we plot the joint torques, contact states, and learned state estimate of the robot in Figure 7. The robot trots at two different frequencies and is shoved twice by the operator, once from each side. The state estimator correctly predicts the direction of the lateral velocity increase (orange), and adapts the joint torques and contact schedule to correct.

D Contact Schedule Parameterization

At each control timestep, we compute the desired contact from the gait parameters $\boldsymbol{\theta}^{\text{cmd}}$ as follows. First, we increment the global timing variable t by $\frac{\boldsymbol{f}^{\text{cmd}}}{f_\pi}$ where $\boldsymbol{f}^{\text{cmd}}$ is the commanded stepping frequency and f_π is the control frequency. Then, we compute separate timing variables for each foot, clipped between 0 and 1:

$$[t^{\text{FR}}, t^{\text{FL}}, t^{\text{RR}}, t^{\text{RL}}] = \text{clip}([t + \boldsymbol{\theta}_2^{\text{cmd}} + \boldsymbol{\theta}_3^{\text{cmd}}, t + \boldsymbol{\theta}_1^{\text{cmd}} + \boldsymbol{\theta}_3^{\text{cmd}}, t + \boldsymbol{\theta}_1^{\text{cmd}}, t + \boldsymbol{\theta}_2^{\text{cmd}}], 0, 1)$$

From these, we can directly compute the desired contact states:

$$C_{\text{foot}}^{\text{cmd}}(t^{\text{foot}}(\boldsymbol{\theta}^{\text{cmd}}, t)) = \Phi(t^{\text{foot}}, \sigma) * (1 - \Phi(t^{\text{foot}} - 0.5, \sigma)) + \Phi(t^{\text{foot}} - 1, \sigma) * (1 - \Phi(t^{\text{foot}} - 1.5, \sigma))$$

where $\Phi(x; \sigma)$ is the cumulative density function of the normal distribution:

$$\Phi(x; \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x}{\sigma})^2}$$

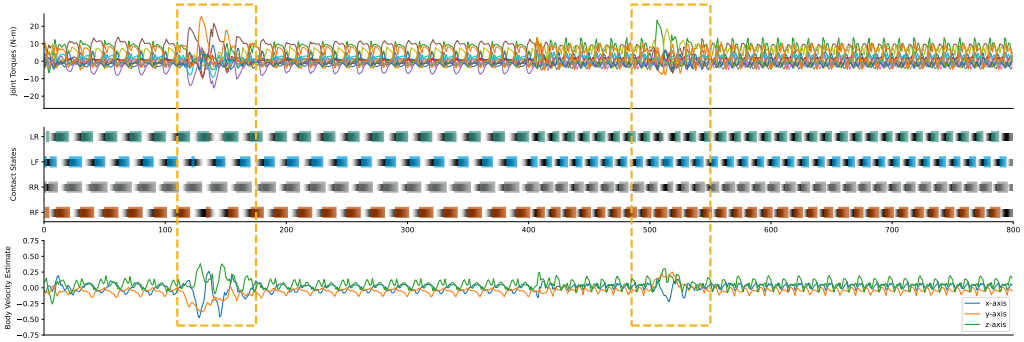

Figure 7: **Shove Robustness Test.** Joint torques (top), contact states (middle), and velocity estimate (bottom) during trotting in the laboratory setting. Dashed boxes indicate shove events. The robot was first shoved to the right during trotting at low frequency, and then shoved to the left during trotting at high frequency. The learned state estimator correctly infers the change in lateral velocity and adjusts the joint torques and foot contacts to stabilize the robot.

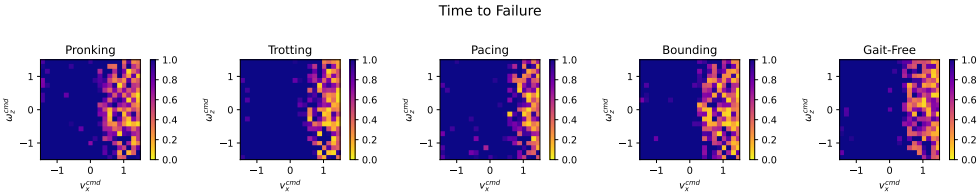

Figure 8: **Forward vs Backward Walking on Platforms.** Time to failure for different gaits and velocities in the random platforms environment (zero-shot test). The temperature bar unit is the mean fraction of a 20s episode elapsed before failure, between zero and one. The gait frequency is 3Hz. The pacing gait boasts the longest mean survival time, possibly due to higher footswings or more practice with recovery strategies during training. Interestingly, almost all failures occur during forward locomotion, and the robot is much more robust when moving backward. Possible explanations include (i) recovery strategies that are dependent on knee orientation and (ii) the weight distribution of the robot.

which is an approximation to the Von Mises distribution used in [13] to form a smooth transition between stance and swing.

Hyperparameter	Value
discount factor	0.99
GAE parameter	0.95
# timesteps per rollout	21
# epochs per rollout	5
# minibatches per epoch	4
entropy bonus (α_2)	0.01
value loss coefficient (α_1)	1.0
clip range	0.2
reward normalization	yes
learning rate	1e-3
# environments	4096
# total timesteps	2.58B
optimizer	Adam

Table 7: PPO hyperparameters.

Parameter	Value	Units
$\mathbf{v}_x^{\text{cmd}}$ initial	[-1.0, 1.0]	m/s
$\boldsymbol{\omega}_z^{\text{cmd}}$ initial	[-1.0, 1.0]	rad/s
$\mathbf{v}_x^{\text{cmd}}$ max	[-3.0, 3.0]	m/s
$\boldsymbol{\omega}_z^{\text{cmd}}$ max	[-5.0, 5.0]	rad/s
$\mathbf{v}_x^{\text{cmd}}$ bin size	0.5	m/s
$\boldsymbol{\omega}_z^{\text{cmd}}$ bin size	0.5	rad/s
$r_{v_{x,y}^{\text{cmd}}}$ threshold	0.8	–
$r_{\omega_z^{\text{cmd}}}$ threshold	0.7	–
$r_{c_f^{\text{cmd}}}$ threshold	0.95	–
$r_{c_v^{\text{cmd}}}$ threshold	0.95	–

Table 8: Curriculum parameters.

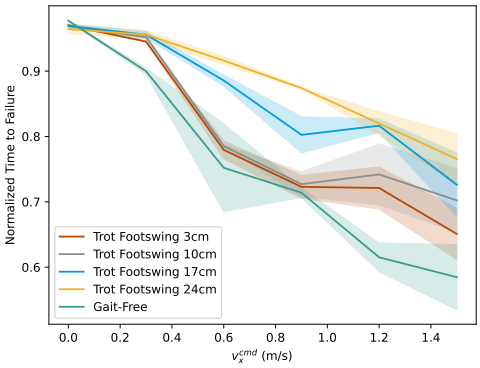

Figure 9: **Footswing Height vs Robustness**: Impact of footswing height on time to failure on the platform terrain (Section 4.3). Increased footswing height yields better generalization to uneven terrain from flat terrain compared to the gait-free policy.

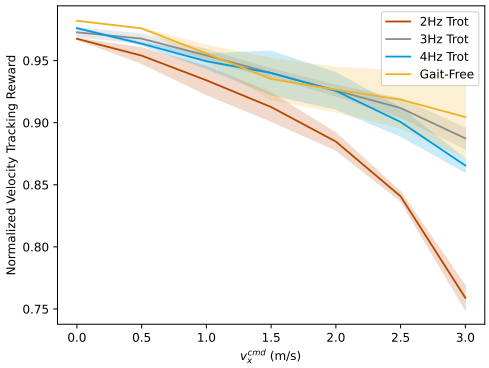

Figure 10: **Frequency vs Speed**: Impact of trotting frequency on flat-ground velocity tracking reward across speeds (Section 4.3). Enforcing low frequency (2Hz) makes high speeds less attainable. The gait-free policy offers slightly better performance at the lowest and highest speeds.

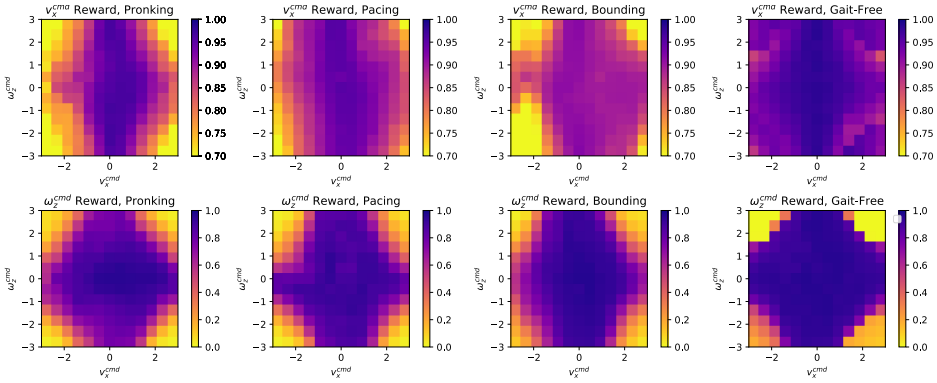

Figure 11: **Flat Ground Velocity Tracking Heatmaps**: We provide heatmaps as in Table 5 for the other major gaits: pronking, pacing, and bounding. In all cases, the policy forgoes performance on the training task of flat-ground velocity tracking to achieve diversity that will help accomplish new tasks.