Walk These Ways: Gait-conditioned Policies Yield Diversified Quadrupedal Agility

Anonymous Author(s)
Affiliation
Address
email

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 at variable frequency, 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

Figure 1: Diverse gaits enable a human to pilot a quadruped expressively and respond to conditions that were not anticipated at training time. A single policy realizes crouching, hopping, high-speed running, agile leaps, and rhythmic dance. The controller is demonstrated on uneven, slippery, and granular terrain and can traverse down stairs, but not up.

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 deployments, 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.

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].

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

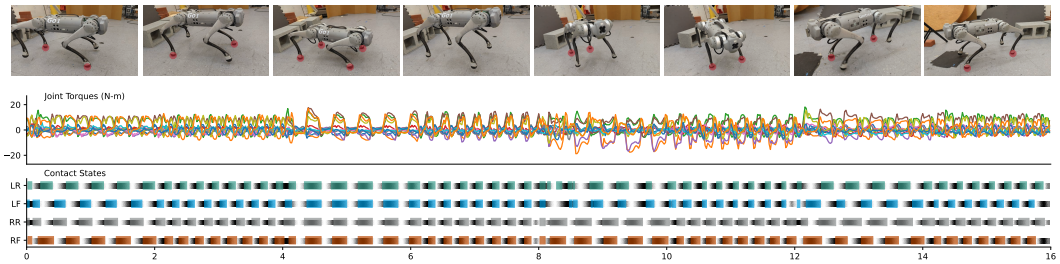

Figure 2: Transition between trotting, pronking, pacing, and bounding in place at alternating frequencies of 2Hz and 4Hz. Images show the robot achieving the contact phases of each gait, with stance feet highlighted in red. Black shading in the bottom plot reflects the timing reference variables $\mathbf{t}_t$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

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 properties and natural terrains. Our definition of gait parameters, which extends previous work by Siekmann et al. [13] from the bipedal setting, enables the composition of behaviors useful for multiple downstream tasks, including navigation through confined spaces, agile jumps longer than the robot's body length, and the execution of dynamic choreographed dance. No previously proposed reinforcement learning agent has been suitable to accomplish any of these tasks, much less to accomplish them all with a single policy.

2 Related Work

Most previous work on learning locomotion addresses the task of achieving a target body velocity [1, 2, 3, 11, 13, 14, 15, 12, 16]. 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 robot can move in order to achieve the same target velocity; for example, 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 implicit or explicit 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], body motion [17], or deviation from a nominal pose [16] 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, 15, 12, 16]. 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.

Imitation Learning. Another line of work uses reference trajectories to specify robot motions [5, 6, 7, 18, 19, 20]. Some approaches use an imitation objective to reward states similar to the reference motion [5, 20, 21], while others use adversarial techniques to reward stylistically similar movements [6, 7]. Shao et al. [20] notably generated a library of reference trajectories for a quadrupedal robot from a set of foot phase parameters 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 would 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.

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)	$[1 - C_{\text{foot}}^{\text{cmd}}(\boldsymbol{\theta}^{\text{cmd}}, t)] \exp\{-\lvert\mathbf{f}^{\text{foot}}\rvert^2/\sigma_{cf}\}$	−0.08
$r_{c_v^{\text{cmd}}}$: stance phase tracking (velocity)	$[C_{\text{foot}}^{\text{cmd}}(\boldsymbol{\theta}^{\text{cmd}}, t)] \exp\{-\lvert\mathbf{v}_{xy}^{\text{foot}}\rvert^2/\sigma_{cv}\}$	−0.08
$r_{h_z^{\text{cmd}}}$: body height tracking	$(\boldsymbol{z} - \boldsymbol{h}_z^{\text{cmd}})^2$	−0.2
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	$\mathbb{1}_{q_i \geq q_{max} \mid q_i < q_{min}}$	−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. Terms combined from [11, 16, 22].

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 [22] and target foot placements [23]. Our task reward is modeled on Siekmann et al. [22], which used gait-dependent reward terms to penalize foot motion during stance and contact force during swing phases. We extend [22] to a more diverse family of quadrupedal gaits and provide an expanded treatment in the setting of robust and high-speed locomotion.

3 Method

3.1 Gait Specification

We parameterize the common quadrupedal gaits by an 8-dimensional command vector,

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

- $\mathbf{v}_x^{\text{cmd}}, \mathbf{v}_y^{\text{cmd}}, \boldsymbol{\omega}_z^{\text{cmd}}$ 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. $\boldsymbol{\theta}^{\text{cmd}} = (0, 0, 0)$ corresponds to a pronking pattern with the contact timings of all four feet synchronized. $\boldsymbol{\theta}^{\text{cmd}} = (0.5, 0, 0)$ corresponds to a trotting pattern with diagonal pairs alternating contact. $\boldsymbol{\theta}^{\text{cmd}} = (0, 0.5, 0)$ represents a pacing pattern with left and right pairs alternating contact. $\boldsymbol{\theta}^{\text{cmd}} = (0, 0, 0.5)$ represents a bounding pattern with front and rear pairs alternating contact. Our policy also learns to continuously interpolate between the major patterns, enabling variants 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. For a visual illustration, refer to Figure 2 or the accompanying video.
- $\boldsymbol{f}^{\text{cmd}}$ 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.

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 pose tracking, and contact schedule tracking. To define the desired contact schedule, function $C_{\text{foot}}^{\text{cmd}}(\boldsymbol{\theta}^{\text{cmd}}, t)$ computes the desired contact state of

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	$\mathrm{m/s^2}$
$\mathbf{v}_x^{\mathrm{cmd}}$	–	–	m/s
$\mathbf{v}_x^{\mathrm{cmd}}$	-0.6	0.6	m/s
$\boldsymbol{\omega}_z^{\mathrm{cmd}}$	–	–	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

Table 2: Randomization ranges for dynamics parameters (top) and commands (bottom) during training. $\mathbf{v}_x^{\mathrm{cmd}}, \mathbf{v}_y^{\mathrm{cmd}}$ are adapted according to a curriculum.

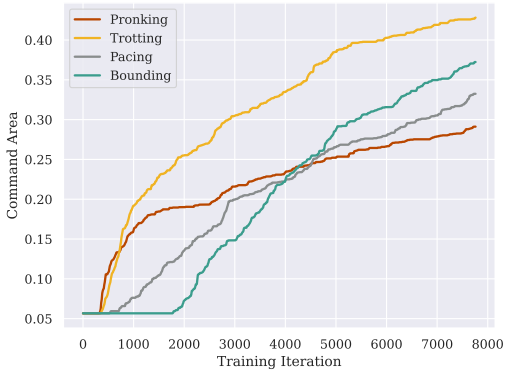

Figure 3: Pronking and trotting gaits are easier to learn and tend to dominate pacing and bounding early in training. However, when discovered, pacing and bounding gaits can yield good performance and later become preferred for some downstream tasks (Section 4.4).

each foot from the phase and timing variable, as described in [22], with details given in the appendix. Stability and smoothness rewards express auxiliary locomotion objectives that are desired across all gaits. As in [16], 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.

3.2 Learning Quadrupedal Gaits

We implement our training environment in the Isaac Gym simulator [24] and train our locomotion policy using Proximal Policy Optimization [25].

Observation Space. The observation $\mathbf{o}_t$ 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:

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

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 [22] to represent the quadrupedal gaits.

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

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 randomized parameters and ranges are given in Table 2.

Policy Architecture. We apply the Concurrent State Estimation architecture proposed by Ji et al. [16]. 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.

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 ($\boldsymbol{h}_z^{\mathrm{cmd}} = 40\,\mathrm{cm}$)	$10_{\pm 1}$	$23_{\pm 1}$	$52_{\pm 4}$	$95_{\pm 9}$

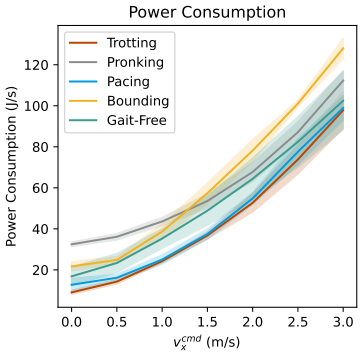

Table 4: Power consumption ($\mathrm{J/s}$) across speeds for common quadrupedal gaits and for policy without gait constraint. While efficiency varies across gaits and speeds, trotting and pacing gaits achieve efficiency competitive with unconstrained gait across speeds.

Task Curriculum. To enable the robot to both run and spin fast, we sample velocity commands using the grid adaptive curriculum strategy from Margolis et al. [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. Finally, the remaining command parameters $(\mathbf{v}_y^{\mathrm{cmd}}, \boldsymbol{f}^{\mathrm{cmd}}, \boldsymbol{h}_z^{\mathrm{cmd}})$ are sampled independently and uniformly. Their ranges are given in Table 2.

3.3 Deployment

We deploy our controller in the real world on the Unitree Go1 Edu robot [26]. 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) [27] to pass sensor data, motor commands, and joystick state between our code and the low-level control SDK provided by Unitree.

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.

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. Figure 3 shows the early progression of the velocity curriculum for each gait. Notably, the delayed progress of pacing and bounding suggests that these gaits may be more challenging to learn than trotting and pronking. This hypothesis is supported by the rarity of pacing and bounding in the reinforcement learning literature. Likewise, the high learnability of trotting aligns with its tendency to emerge from a variety of non-gait-specific auxiliary rewards [2, 11, 16, 17]. 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 all emerge as well.

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	$0.81^{(0.96)}_{\pm 0.03}$	$0.74^{(0.92)}_{\pm 0.06}$	–	–	$0.83^{(1.00)}_{\pm 0.01}$

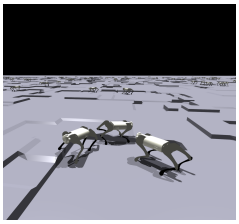

Table 5: Zero-shot robustness evaluation on discrete platform terrain (visualized right). The pacing and trotting gaits yield the best survival time during zero-shot deployment on this particular terrain, outperforming the gait-free baseline. Pronking attains the best velocity tracking performance of any gait, with a similar survival time to the gait-free baseline. Reward is reported as a fraction of the total possible episodic reward. Superscript reports performance in the flat training environment with no platforms. Subscript reports standard deviation across three random seeds.

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 the energy-efficiency, robustness, and speed of learned quadrupedal locomotion. 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.

(Intent of Analysis). The goal of our evaluation is *not* to endorse a specific gait for a specific scenario, for example to recommend pacing over pronking on one class of uneven terrain. The characteristics of specific gaits vary with different robot morphologies, training details, and environmental properties. We aim to illustrate that different 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. Table 4 reports the power consumption profile for each gait. The robot expends less energy while trotting or pacing than while pronking or bounding. One might hypothesize that the most energy-efficient gaits are also the easiest to learn, but our results go against this hypothesis (pronking emerges earlier than pacing; see Section 4.1). Additionally, fixed gaits 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 5). All policies in this work are trained only on flat ground, so the platform terrain is out-of-distribution. In Table 5, 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 ($10\,\text{s}$) before failure. The episode was terminated if the agent did not fall after 10 seconds. In this experiment, velocity commands were limited to $\mathbf{v}_x^{\text{cmd}} \in [-2, 2]$, $\boldsymbol{\omega}_z^{\text{cmd}} \in [-2, 2]$.

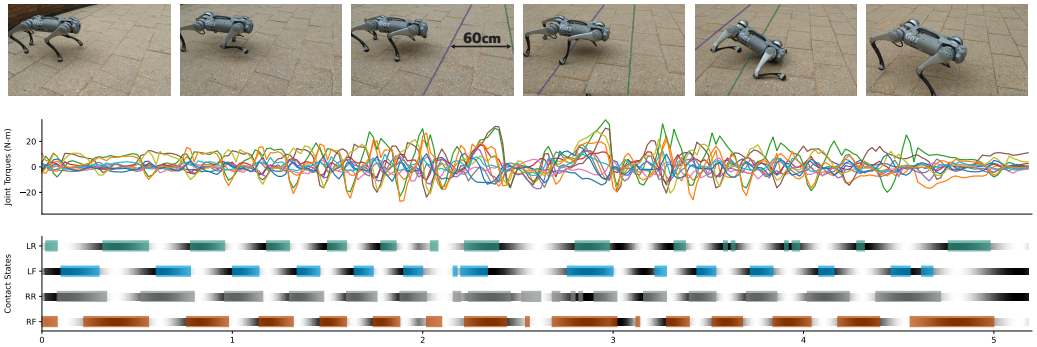

Figure 4: Gait parameter modulation supports the downstream task of crossing a gap wider than the robot's body length with a single leap. We show joint torques (top) and contact states (bottom) during an agile leap on pavement. The robot first accelerates to a target speed of $3\,\mathrm{m/s}$ at a trot 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.

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 the velocity tracking performance of multiple gaits and the gait-free baseline across a wide range of speeds including high-speed sprinting in Table 6. Removing gait constraints results in an improvement in velocity tracking task performance on flat ground. Heat maps break down the mean task reward for each velocity command, revealing that the gait-free approach does not offer much benefit at low speeds, but is better for achieving combinations of high linear and angular velocity.

4.4 The Downstream Utility of Gait Modulation

Navigation of confined spaces. Different gaits and postures 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 height is $13\,\mathrm{cm}$, leaving a maximum of $9\,\mathrm{cm}$ of clearance between the robot and the ground, not accounting for vertical motion of the body during locomotion. We also find that the robot can pass between between two narrowly spaced cinder blocks with a pacing gait, but gets stuck with the wider stance of the pronking gait.

Agile jumping. Modulation of contact schedule, velocity, and gait frequency at high speed can encode an agile jump. Figure 4 shows the contact states, joint torques, and estimated body velocity during a jump sequence. First, the robot is commanded to accelerate to $3\,\mathrm{m/s}$ at a trot while increasing its stepping frequency from $2\,\mathrm{Hz}$ to $4\,\mathrm{Hz}$. Second, a jump is initiated by commanding one second of pronking at $2\,\mathrm{Hz}$. Finally, the robot decelerates by reversing the acceleration sequence. During the jump phase, the distance from the location of the robot's front feet at takeoff to the location of the hind feet upon landing is measured to be $60\,\mathrm{cm}$.

Choreographed Dance. We program a sequence of gait parameters to generate a dance routine synchronized to a jazz song with a tempo of 90 beats per minute. 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 the body height and velocity in time

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	$0.94_{\pm 0.02}$	$0.76_{\pm 0.01}$	–	–

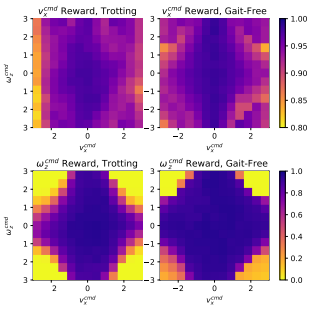

Table 6: Task reward reported as a fraction of the maximum possible reward for different gaits with ablations. Removing gait constraints results in an improvement in velocity tracking task performance on flat ground. Heat maps (right) break down the mean task reward for each velocity command, revealing that the gait-free approach is most beneficial for combinations of high linear and angular velocity.

with the music. An assistant script helps a human programmer procedurally generate a list of gait parameters at each timestep, which are fed into the neural network controller in open-loop fashion during deployment. The programmer is a robotics researcher who has no formal experience with dance, music theory, or choreography.

We believe ours to be the most flexible and dynamic published system for legged robotic dance to date. Notable prior works in legged robotic dance include Bi et al. [28], Shao et al. [20], and the unpublished demonstrations by Boston Dynamics [29]. Bi et al. [28] focused on automatic generation of dance choreography, but their model-based controller was restricted to carefully lifting one foot at a time. Shao et al. [20] defined a comparable gait modulation language to ours, but their use of a complete reference motion library, rather than a contact timing reward, produced behaviors that are less fast and dynamic. Boston Dynamics' dance demonstrations on Spot and Atlas [29] are based on an unpublished analytically designed controller.

5 Limitations

We implement a task parameterization and training procedure that allow a teleoperator or high-level planner to specify the locomotion behavior of a quadruped in diverse ways. Specifying the locomotion task reduces the system's degree of autonomy. Our experiments show that this hinders the robot's performance at very high speeds (Section 4.3; Table 6), suggesting that successful sprinting departs meaningfully from our parameterized family of gaits. To reparameterize the locomotion task without discouraging useful sprinting strategies is an open direction for future work.

Our work describes a blind locomotion controller that does not make use of exteroceptive sensor data from the robot's onboard cameras. This limits the system from anticipatively adapting to the terrain, which would be useful for tasks like upward stair traversal where reactive robustness is insufficient.

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 by a teleoperator 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, with the caveat that imposing such a parameterization can limit top-end sprinting performance.

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] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] R. Li, A. Jabri, T. Darrell, and P. Agrawal. Towards practical multi-object manipulation using relational reinforcement learning. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 4051–4058, Virtual, May 2020. doi:10.1109/ICRA40945.2020.9197468.

[22] 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.

[23] 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.

[24] 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.

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

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

[27] 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.

[28] T. Bi, P. Fankhauser, D. Bellicoso, and M. Hutter. Real-time dance generation to music for a legged robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1038–1044. IEEE, 2018.

[29] Boston Dynamics, Spot, 2022, `https://www.bostondynamics.com/spot`, [Online; accessed Jun. 2022].