Walk These Ways: Learning with Parametric Auxiliary Rewards for an Uncertain Future

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

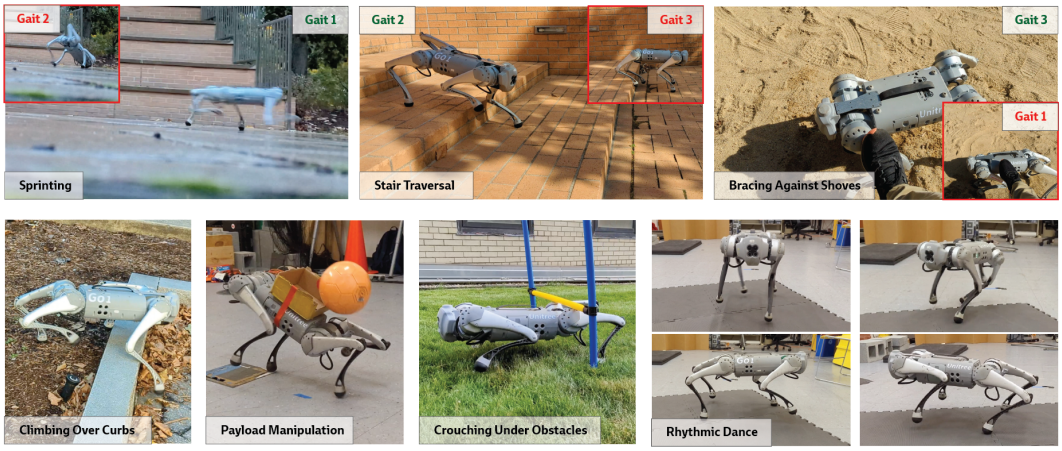

Figure 1: *Parametric auxiliary rewards* enable a human to pilot a quadruped expressively and perform online tuning, improving performance in environments outside the training distribution. A single policy embodies a large continuous space of distinct motion styles useful for downstream tasks. In the top row, the nested images illustrate how the monolithic auxiliary reward corresponding to Gait 1, 2, or 3 would fail in one of the environments shown despite succeeding in another.

Abstract: Learned locomotion policies can rapidly adapt to diverse environments similar to the ones the agent experienced during training, but lack a mechanism for fast tuning when they fail in an out-of-distribution test environment. This necessitates a slow, iterative cycle of reward and environment redesign to achieve good performance in sim-to-real transfer. As an alternative, we propose training a single policy with parametric auxiliary rewards to represent a structured family of locomotion strategies. The auxiliary reward parameters can be adjusted in real-time to improve performance in out-of-distribution environments without time-consuming retraining. We release a fast, robust open-source locomotion controller, *Walk These Ways*, that can trot, pronk, pace, and bound with variable footswing, posture, and speed, unlocking human demonstration of diverse downstream tasks including crouching, hopping, high-speed running, stair traversal, bracing against shoves, and rhythmic dance. Video and code release: https://sites.google.com/view/gait-conditioned-rl/.

Keywords: Locomotion, Reinforcement Learning, Task Specification

1 Introduction

Consider training a locomotion policy to be applied in all of the environments and tasks in Figure 1: climbing over stairs and curbs, sprinting fast on flat ground, bracing against shoves, crawling under obstacles, manipulating payloads, and executing choreographed dance. Online system identification approaches have been shown to rapidly adapt to in-distribution variations in the environment [1,

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

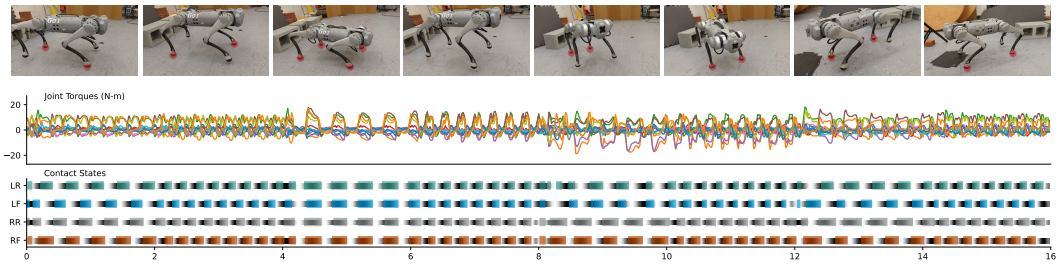

Figure 2: Transition between trotting, pronking, pacing, and bounding in place at alternating frequencies of 2Hz and 4Hz. Images show the robot achieving the contact phases of each gait, with stance feet highlighted in red. Black shading in the bottom plot reflects the timing reference variables $\mathbf{t}_t$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

2]. However, they have no fast mechanism for adjusting their behavior if they fail in an out-of-distribution environment or task. In practice, all of the environments we care about in sim-to-real reinforcement learning are out-of-distribution due to imperfect simulation. Therefore, sim-to-real practitioners implement application-specific biases towards generalization, such as constraints on contact schedule, body posture, energy consumption, or footswing height [1, 2, 3, 4, 5, 6]. These biases are commonly encoded through auxiliary rewards and tuned as hyperparameters of RL. This tuning process is slow because it requires training and deploying the policy repeatedly, and it results in a single tuned bias or preference among behaviors. Such a monolithic bias can be insufficient to yield success across diverse tasks. For example, consider sprinting, climbing stairs, and resisting shoves: each can be thought of as requiring a different tradeoff between robustness and agility. The tuning of the auxiliary reward impacts this tradeoff. We illustrate this in the nested red boxes in the top row of Figure 1; each shows an instance where a fixed choice of auxiliary reward would lead to failure for one task, despite working well for another.

Towards reducing the labor of tuning policies for open-world sim-to-real transfer, we propose and implement a task specification for learned quadrupedal locomotion based on *parametric auxiliary rewards*. Here, the auxiliary reward is defined by a set of parameters, which are sampled randomly at regular intervals during training and also directly observed by the policy. This incentivizes learning locomotion with different gaits, postures, and speeds that will be controllable by changing the input to the policy at deployment time. A language of gait parameters is common in the setting of analytical control, where it is usually necessary to define constraints, objectives, and models [7, 8, 9]. It is more often avoided in reinforcement learning, where it is possible for motion styles to emerge from a reward function without direct specification [1, 3, 4, 6].

We find that a single policy with parametric auxiliary rewards 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 real-time tuning of the locomotion behavior or gait for application to new downstream tasks.

2 Related Work

Most previous work on learning locomotion addresses the task of achieving a target body velocity [1, 2, 4, 5, 6, 10, 11, 12, 13]. 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 tuning auxiliary rewards. For example, penalties on energy expenditure [3, 14], body motion [4, 15], or footswing height [5] can serve to break ties in the simulated training environment, in favor of gaits that generalize to a target environment. 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 [5, 6, 10, 13]. Fu et al. [3] notably found 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 [16, 17, 18, 19, 20, 21]. Some approaches use an imitation objective to reward adherence to the reference motion [16, 20, 21], while others use adversarial techniques to reward stylistically similar movements [17, 18]. Most imitation learning approaches learn only one style of behavior per policy [16, 17]. Vollenweider et al. [18] provided a recent demonstration that a small number of discrete motion styles can be learned simultaneously. Shao et al. [21] notably generated a library of reference trajectories and trained a goal-conditioned policy to imitate them. However, using a library of full reference trajectories can penalize the agent for exploring behaviors that are difficult to procedurally generate. We prefer to specify the desired behavior more softly as an auxiliary reward, hoping this will help retain complex strategies such as dynamic body motion or slip recovery.

Generalized Diversity Objectives. Several prior methods explicitly optimize for a generalized metric of diversity during training. Quality diversity methods [22, 23, 24] learn a repertoire of high-performing discrete skills by maximizing a novelty objective defined among trajectories. Other work uses diversity as an unsupervised objective function for RL [25]. In contrast, parameterized auxilary reward is good for learning diverse behaviors corresponding to a set of structured parameters that are predefined using domain knowledge.

Parameterized-Reward 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 [26] and target foot placements [27]. Siekmann et al. [26] 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 [26] has few parameters: the timing offset between the two feet and the duration of the swing phase. This smaller space of command parameters was not shown to be useful for compositional tasks or out-of-distribution environments, but it inspires our work.

Hierarchical Control Using Gait Parameters. Several works have learned a high-level policy to modulate the gait parameters of a low-level model-based controller to accomplish downstream tasks. This approach has been applied to energy minimization [14] and vision-guided foot placement [15, 28]. 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. The tradeoffs between model-based control and reinforcement learning are well discussed in prior literature [6, 13]. Our work opens the door to revisiting hierarchical methods with learning at both levels.

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.

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	$(\boldsymbol{\phi} - \boldsymbol{\phi}^{\text{cmd}})^2$	−0.1
$r_{\boldsymbol{s}_y^{\text{cmd}}}$: raibert heuristic footswing tracking	$(\mathbf{p}_{x,y\,\text{foot}}^f - \mathbf{p}_{x,y\,\text{foot}}^f{}^{\text{cmd}}(\boldsymbol{s}_y^{\text{cmd}}))^2$	−0.2
$r_{\boldsymbol{h}_z^{f\,\text{cmd}}}$: footswing height tracking	$\sum_{\text{foot}}(\boldsymbol{h}_{z\,\text{foot}}^f - \boldsymbol{h}_z^{f\,\text{cmd}})^2 C_{\text{foot}}^{\text{cmd}}(\boldsymbol{\theta}^{\text{cmd}}, t)$	−0.6
z velocity	$\mathbf{v}_z^2$	−4e−4
roll-pitch velocity	$\lvert\boldsymbol{\omega}_{xy}\rvert^2$	−2e−5
foot slip	$\lvert\mathbf{v}_{xy}^{\text{foot}}\rvert^2$	−8e−4
thigh/calf collision	$\mathbb{1}_{\text{collision}}$	−0.02
joint limit violation	$\mathbb{1}_{q_i > q_{max} \lvert\lvert q_i < q_{min}}$	−0.2
joint torques	$\lvert\boldsymbol{\tau}\rvert^2$	−2e−5
joint velocities	$\lvert\dot{\mathbf{q}}\rvert^2$	−2e−5
joint accelerations	$\lvert\ddot{\mathbf{q}}\rvert^2$	−5e−9
action smoothing	$\lvert\mathbf{a}_{t-1} - \mathbf{a}_t\rvert^2$	−2e−3
action smoothing, 2nd order	$\lvert\mathbf{a}_{t-2} - 2\mathbf{a}_{t-1} + \mathbf{a}_t\rvert^2$	−2e−3

Task

Parametric Auxiliary

Fixed Auxiliary

Table 1: Reward structure: task rewards, parametric auxiliary rewards, and fixed auxiliary rewards.

- $\boldsymbol{h}_z^{\text{cmd}}$ 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 position rewards are simple functions of the squared error, weighted to account for the relative error magnitude. The desired foot position in the ground plane $\boldsymbol{p}_{x,y\,\text{foot}}^{f\,\text{cmd}}(\boldsymbol{s}_y^{\text{cmd}})$ is computed from stance width, body velocity and contact schedule using the Raibert heuristic [7]. 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 [26], with details given in the appendix. Stability and smoothness rewards express auxiliary locomotion objectives that are desired across all gaits. As in [5], 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 [4, 29] and train our locomotion policy using Proximal Policy Optimization [30].

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:

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

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

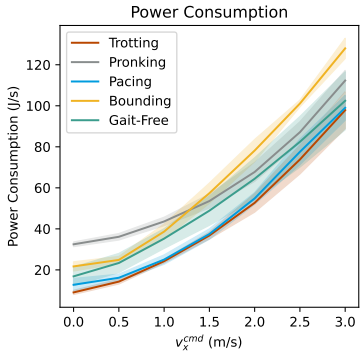

Table 3: Gait-conditioned policies enable interventional studies on the relationship between gait properties and performance criteria within a single policy. Here, we illustrate how power consumption ($\mathrm{J/s}$) varies across speeds for common quadrupedal gaits and for a baseline policy without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

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 [26] 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 $k_p = 20, k_d = 0.5$.

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. To simplify training and provide a case study in generalization, we only train on flat ground and do not randomize the terrain geometry. All randomization ranges are given in Appendix A.

Latency and Actuator Modeling. We perform system identification to reduce the sim-to-real gap in the robot dynamics. Following [13], we train an actuator network to capture the non-ideal relationship between PD error and realized torque. Separately, we identify a consistent latency of around $20\,\mathrm{ms}$ in our system and model this as a constant action delay during simulation.

Policy Architecture. We apply the Concurrent State Estimation architecture proposed by [5]. 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 [6]. 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.

The procedure for sampling a command $\mathbf{c}_t$ is as follows: First, one of the four major contact patterns (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 [6]. To let the policy learn to interpolate between major contact patterns, $(\boldsymbol{\theta}_1^{\mathrm{cmd}}, \boldsymbol{\theta}_2^{\mathrm{cmd}}, \boldsymbol{\theta}_3^{\mathrm{cmd}})$ 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 [31].

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

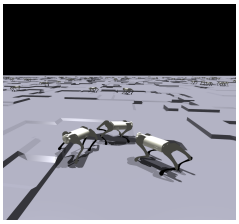

Table 4: Zero-shot robustness evaluation on platform terrain (visualized right). Pacing and trotting gaits yield the best survival time in zero-shot deployment, outperforming the gait-free baseline. Pronking attains the best velocity tracking performance, with similar survival time to the baseline. We 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.

Computing Architecture: An onboard Jetson TX2 NX computer runs our trained policy. We implement an interface based on Lightweight Communications and Marshalling (LCM) [32] 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.

4 Experimental Results

4.1 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 consistently 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. Videos (i)-(iv) on the project website demonstrate the modulation of each parameter.

4.2 Impact of Auxiliary Reward Modulation on Locomotion Performance

We probe the reward parameters of our trained model to quantify their role in quadrupedal locomotion performance. We compare our controller with various auxiliary reward settings to a baseline velocity-tracking controller (the "gait-free baseline"). The gait-free baseline is trained by the same method as our controller, but without the parameterized auxiliary reward terms for contact timing, body posture, and foot placement. 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. Trotting and pacing also consume less energy than the gait-free baseline at low and medium speeds. This indicates that tuning the auxiliary reward parameters can facilitate optimizing a new objective (energy efficiency) in the original training environment.

Robustness. We evaluate learned behaviors 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 (Section 3.2), so the platform terrain is out-of-distribution. In Table 4, we report the mean task reward on the platform terrain for each of the major gaits and

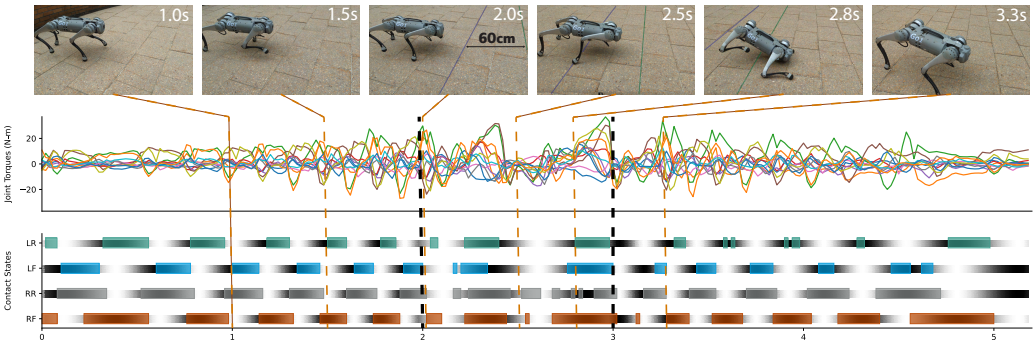

Figure 3: Gait parameter modulation supports the downstream task of crossing a gap wider than the robot's body length with a single leap.

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. Figure 9 in the appendix shows a similar analysis for footswing height. These results suggest that tuning parameterized auxiliary rewards is a viable mechanism for out-of-distribution adaptation.

Velocity Tracking and Top Speed. We report velocity tracking performance of multiple 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 parameterized auxiliary rewards do not hinder performance at low speeds, but are restrictive for combinations of high linear and angular velocity. We conclude that improved generalization via parametric auxiliary rewards can come with some cost to in-distribution task performance.

4.3 Case Studies: Achieving New Tasks with Auxiliary Reward 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 the controller in open-loop fashion.

Climbing over curbs. Modulating footswing height enables our controller to climb smoothly across stairs and curbs (Figure 1). Previous works demonstrating blind obstacle traversal on 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.

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

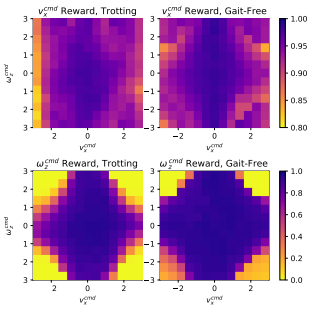

Table 5: Removing gait constraints results in improved velocity tracking task performance on flat ground. Heat maps (right) break down the mean task reward for each velocity command, revealing that the gait-free approach is most beneficial for combinations of high linear and angular velocity.

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, 5]. 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

Our experiments show that the benefits of adding parametric auxiliary rewards can come at a cost to in-distribution task performance, specifically limiting the robot's flat-ground sprinting performance (Section 4.2; Table 5). To quantify and control the tradeoff between task performance and reward shaping, some prior methods have been proposed [33, 34].

Some gaits suffer from performance degradation in the real world. For example, in the agile leap sequence (Figure 3), the robot's body sinks lower than the command height during the leap. Also, for some less stable gaits, such as low-frequency pacing with narrow stance and high body, the robot tends to drift to one side or stumble.

All behaviors we demonstrate rely on a human pilot to tune the auxiliary reward parameters during deployment. In the future, the parameter adaptation could be automated using behavioral cloning on real-world human demonstrations, or by learning a high-level policy with RL as in [14, 15, 28].

6 Conclusion

We have implemented and analyzed a novel method for learning quadrupedal locomotion with parametric auxiliary rewards. The resulting controller can accomplish a number of previously inaccessible downstream tasks and can be rapidly tuned in out-of-distribution environments without any retraining. Quantitative experiments suggest that modulating parametric auxiliary rewards is a practical mechanism to quickly improve RL policy performance 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.

Author Contributions

- **Gabriel B. Margolis** concieved, designed, and implemented the controller, ran all experiments, and played the primary role in paper writing.
- **Pulkit Agrawal** advised the project and contributed to its conceptual development, experimental design, positioning, and writing.

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] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] J. K. Pugh, L. B. Soros, and K. O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, page 40, 2016.

[23] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[24] B. Lim, L. Grillotti, L. Bernasconi, and A. Cully. Dynamics-aware quality-diversity for efficient learning of skill repertoires. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5360–5366. IEEE, 2022.

[25] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

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

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

[32] 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.

[33] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287, 1999.

[34] E. Chen, H. Zhang-Wei, J. Pajarinen, and P. Agrawal. Redeeming intrinsic rewards via constrained optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.