

Ablation	$r_{v_x^{\text{cmd}}}$	$r_{\omega_z^{\text{cmd}}}$	$r_{c_f^{\text{cmd}}}$	$r_{c_v^{\text{cmd}}}$
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	—	—

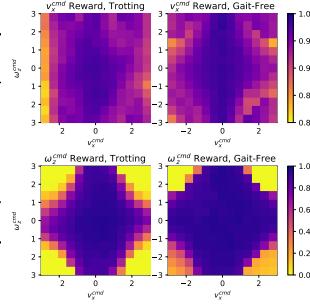


Table 5: Removing gait constraints results in improved velocity tracking task performance on flat ground. Heat maps (right) break down the mean task reward for each velocity command, revealing that the gait-free approach is most beneficial for combinations of high linear and angular velocity.

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	m/s ²
v_x^{cmd}	—	—	m/s
v_y^{cmd}	-0.6	0.6	m/s
ω_z^{cmd}	—	—	m/s
f^{cmd}	1.5	4.0	Hz
$\theta_1^{\text{cmd}}, \theta_2^{\text{cmd}}, \theta_3^{\text{cmd}}$	0.0	1.0	—
h_z^{cmd}	0.10	0.45	m
ϕ^{cmd}	-0.4	0.4	rad
s_y^{cmd}	0.05	0.45	m
h_z^{cmd}	0.03	0.25	m

Table 6: Randomization ranges for dynamics parameters (top) and commands (bottom) during training. $v_x^{\text{cmd}}, \omega_z^{\text{cmd}}$ are adapted according to a curriculum.

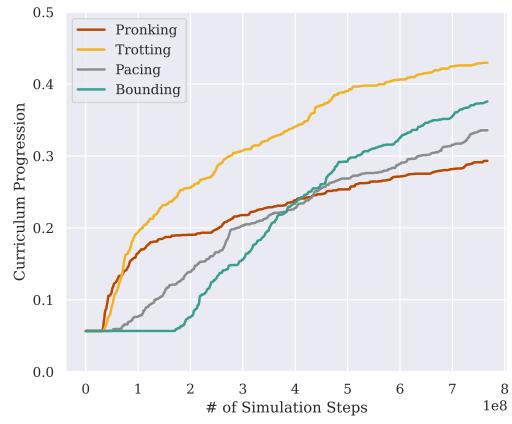


Figure 4: Pronking and trotting gaits are easier to learn and tend to dominate pacing and bounding early in training. However, when discovered, pacing and bounding gaits can yield good performance and later become preferred for some downstream tasks (Section 4.2).

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 [3]; 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 Interface

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.



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.

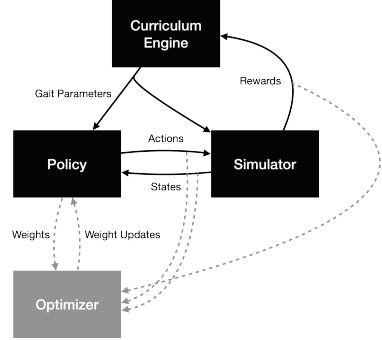


Figure 6: **Training architecture.** The policy computes the action as a function of the gait parameters and state. The simulator computes the reward and state as a function of the gait parameters and actions. The curriculum engine periodically resamples gait parameters based on the reward.

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 in Table 11 for pronking, pacing, and bounding gaits to supplement the trotting and gait-free heatmaps provided in Table 5.

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 θ^{cmd} as follows. First, we increment the global timing variable t by $\frac{f^{\text{cmd}}}{f_\pi}$ where f^{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:

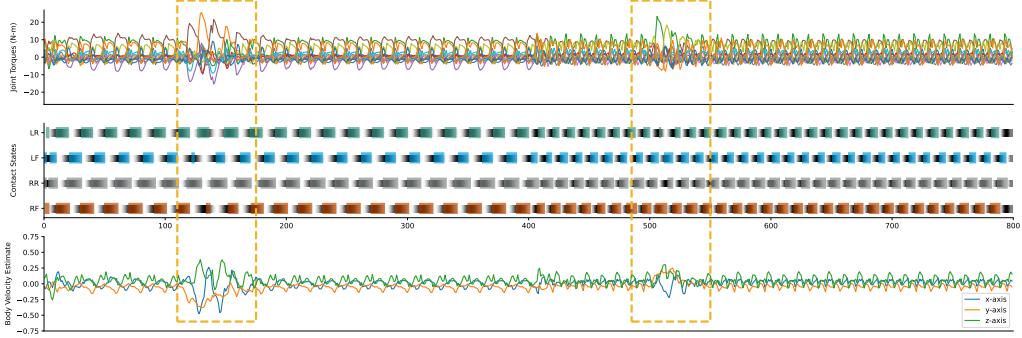


Figure 7: **Shove Robustness Test.** Joint torques (top), contact states (middle), and velocity estimate (bottom) during trotting in the laboratory setting. Dashed boxes indicate shove events. The robot was first shoved to the right during trotting at low frequency, and then shoved to the left during 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.

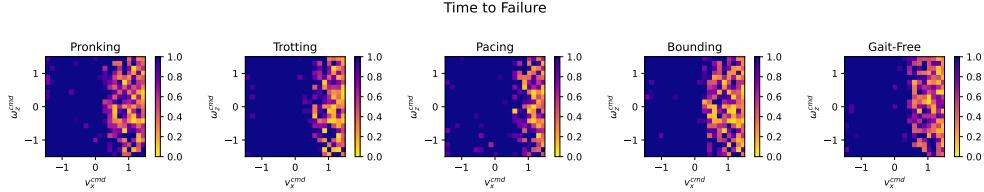


Figure 8: **Forward vs Backward Walking on Platforms.** Time to failure for different gaits and velocities in the random platforms environment (zero-shot test). The temperature bar unit is the mean fraction of a 20s episode elapsed before failure, between zero and one. The gait frequency is 3Hz. The 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.

$$[t^{\text{FR}}, t^{\text{FL}}, t^{\text{RR}}, t^{\text{RL}}] = \text{clip}([t + \theta_2^{\text{cmd}} + \theta_3^{\text{cmd}}, t + \theta_1^{\text{cmd}} + \theta_3^{\text{cmd}}, t + \theta_1^{\text{cmd}}, t + \theta_2^{\text{cmd}}], 0, 1)$$

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:

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

which is an approximation to the Von Mises distribution used in [8] 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
v_x^{cmd} initial	[-1.0, 1.0]	m/s
ω_z^{cmd} initial	[-1.0, 1.0]	rad/s
v_x^{cmd} max	[-3.0, 3.0]	m/s
ω_z^{cmd} max	[-5.0, 5.0]	rad/s
v_x^{cmd} bin size	0.5	m/s
ω_z^{cmd} bin size	0.5	rad/s
$r_{v_x^{\text{cmd}}, y}$ threshold	0.8	–
$r_{\omega_z^{\text{cmd}}}$ threshold	0.7	–
$r_{e_f^{\text{cmd}}}$ threshold	0.95	–
$r_{e_v^{\text{cmd}}}$ threshold	0.95	–

Table 8: Curriculum parameters.

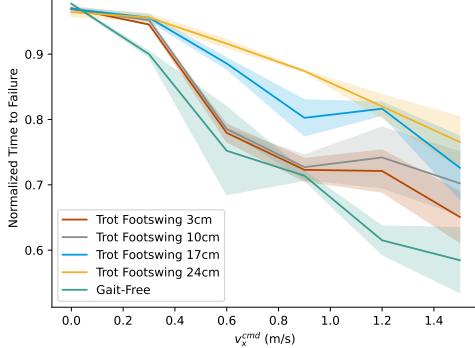


Figure 9: **Footswing Height vs Robustness:** Impact of footswing height on time to failure on the platform terrain (Section 4.2). Increased footswing height yields better generalization to uneven terrain from flat terrain compared to the gait-free policy.

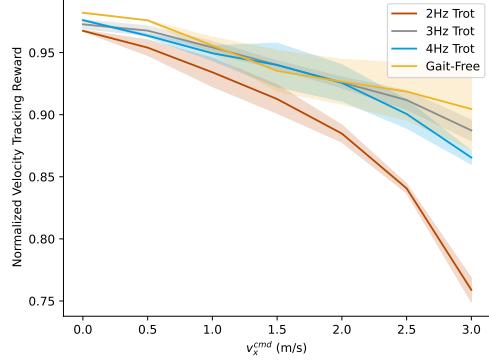


Figure 10: **Frequency vs Speed:** Impact of trotting frequency on flat-ground velocity tracking reward across speeds (Section 4.2). Enforcing low frequency (2Hz) makes high speeds less attainable. The gait-free policy offers slightly better performance at the lowest and highest speeds.

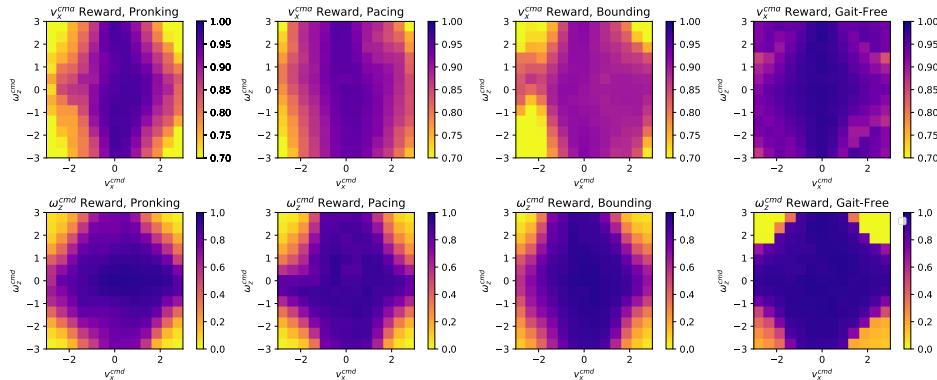


Figure 11: **Flat Ground Velocity Tracking Heatmaps:** We provide heatmaps as in Table 5 for the other major gaits: pronking, pacing, and bounding. In all cases, the policy forgoes performance on the training task of flat-ground velocity tracking to achieve diversity that will help accomplish new tasks.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