

A DETAILED EXPERIMENT SETUP

A.1 REWARD

In this section, we detail the reward function and the weights. The reward function is comprised of several critical components, each serving a distinct purpose. The exponential function is represented by $\exp(\cdot)$, and the variance function is denoted by $\text{var}(\cdot)$. The terms $(\cdot)^{\text{des}}$ and $(\cdot)^{\text{cmd}}$ are used to denote the desired and commanded values, respectively. The robot’s body frame is defined by the coordinates x , y , and z , with x and z oriented in the forward and upward directions. The rotation angles of the robot’s coordinate system are given by roll, yaw, and pitch. $P_{f(t)}$, $I_{d(t)}$, $I_{c(t)}$, T_{air} , v , ω , h , d_f , d_k , g , θ , τ represent the height of the foot at time t , the phase at time t in the gait cycle, the contact status of the foot at time t , the aerial time of the foot, the linear velocity of the robot’s root, yaw rate, height, foot distance, knee distance, the projection of the gravity vector onto the robot’s body frame, joint position, and joint torque.

Table 4: Setup of reward function and scales.

Reward	Equation (r_i)	Scale (w_i)
Feet clearance	$(p_{f(t)}^{\text{des}} - p_{f(t)})^2 \cdot (1 - I_{d(t)})$	-0.01
Feet air time	$T_{\text{air}} \cdot I_{c(t)}$	-0.001
Follow gait phase	$(1 - I_{d(t)}) \cdot I_{c(t)}$	-0.001
Feet slip	$\omega \cdot I_{c(t)}$	-0.005
Feet&Knee distance	$\frac{\exp\{-100 \cdot 0.3 - d_{f,k} \} + \exp\{-100 \cdot 0.125 - d_{f,k} \}}{2}$	0.4
Lin. velocity tracking	$\exp\left\{-4 \left(\mathbf{v}_{xy}^{\text{cmd}} - \mathbf{v}_{xy}\right)^2\right\}$	2.4
Ang. velocity tracking	$\exp\left\{-4 \left(\omega_{\text{yaw}}^{\text{cmd}} - \omega_{\text{yaw}}\right)^2\right\}$	1.1
Velocity mismatch	$\frac{\exp\{-10(-v_z)^2\} + \exp\{-5(-\omega_{\text{roll, pitch}})^2\}}{2}$	0.5
Orientation	$ \mathbf{g} ^2$	1.0
Feet orientation	$ \mathbf{g}_{\text{feet}} ^2$	1.0
Default joint	$\exp\left\{-2(\theta - \theta_{\text{zero}})^2\right\}$	0.5
Body height	$(h^{\text{des}} - 0.6505)^2$	-1.0
Root accelerations	$\exp\left\{-\left(\ddot{\theta}_{\text{root}}\right)^3\right\}$	0.2
Joint accelerations	$\ddot{\theta}^2$	-1×10^{-6}
Joint velocity	$\dot{\theta}^2$	-5×10^{-3}
Joint power	τ^2	-1×10^{-5}
Action rate	$(\mathbf{a}_t - \mathbf{a}_{t-1})^2$	-0.01
Smoothness	$(\mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2})^2$	-0.01
Joint position tracking	$\exp\left\{-2(\theta - \theta_{\text{target}})^2\right\}$	3.2

A.2 DOMAIN RANDOMIZATION

We leverage domain randomization during training to narrow the reality gap. Specifically, we set the range of parameters as shown in Table 5, mainly consisting of delay of action and torque, randomization of position, velocity, friction, KP/KD factor, and CoM.

A.3 IMPLEMENTATION DETAILS

Our humanoid robot, named N1, is equipped with a total of 18 degrees of freedom. In this work, we have immobilized the 8 joints associated with the arms, focusing exclusively on the 10 joints related

Table 5: Overview of Domain Randomization. Presented are the domain randomization terms and the associated parameter ranges. Additive randomization increments the parameter by a value within the specified range while scaling randomization adjusts it by a multiplicative factor from the same range.

Parameter	Unit	Range	Operator	Type
Joint Position	rad	[-0.05, 0.05]	additive	Gaussian (lo)
Joint Velocity	rad/s	[-1.5, 1.5]	additive	Gaussian (lo)
Angular Velocity	rad/s	[-0.2, 0.2]	additive	Gaussian (lo)
Linear Velocity	m/s	[-0.1, 0.1]	additive	Gaussian (lo)
Euler Angle	rad	[-0.06, 0.06]	additive	Gaussian (lo)
Action Delay	ms	[0, 10]	-	Uniform
Torque Delay	ms	[0, 10]	-	Uniform
Friction	-	[0.1, 2.0]	-	Uniform
Kp factor	%	[80, 120]	scaling	Gaussian (lo)
Kd factor	%	[80, 120]	scaling	Gaussian (lo)
Motor Strength	%	[80, 120]	scaling	Gaussian (lo)
Payload	kg	[-5, 5]	additive	Gaussian (lo)
CoM	m	[-0.02, 0.02]	additive	Gaussian (lo)
Link Mass	%	[0.9, 1.1]	scaling	Gaussian (lo)

339 to the legs. The motors’ hip (pitch) and knee joint torque can reach up to 150 Nm, while the motors’
340 torque of the foot joints is 36 Nm. This robot’s total height and weight are 0.95 m and 23 kg.

341 Our RL control strategy operates at 100 Hz, coupled with an internal PD controller that runs at 1000
342 Hz. It ensures synchronization with the operational frequency of the actual hardware. We employ
343 Isaac Gym for training and conduct sim-to-sim validation in various simulation environments, in-
344 cluding MuJoCo, PyBullet, and Gazebo. This multi-environment approach ensures the robustness
345 and adaptability of our models. We utilize the Proximal Policy Optimization (PPO) algorithm [30].
346 The details of our training parameters are presented in Table 6, where we outline the specifics that
347 contribute to the enhanced performance of our model.

Table 6: Training hyperparameters

Parameter	Value
Number of environments	4096
Training epochs	2
Learning rate	10^{-5}
Gamma γ	0.995
Lambda λ	0.95
Batch size	24
Episode Length	3000
Backbone hidden layers	[512, 512, 128]
Encoder hidden layers	[768, 256, 64]
Activation function	ELU
Decoder observation	19
Number of Observation frame stack	5
Number of Privileged frame stack	3
Number of Aggregated Observation	219
Number of Aggregated Privileged Observation	846

348 B ALGORITHM

349 We present our complete algorithmic process in the following algorithm, where four networks are
 350 continuously updated through sampling from the simulation based on the process outlined below.

Input: Encoder network est_ϕ , decoder network dec_φ , policy network π_θ , value function V_ψ , environment, number of epochs E , mini-batch size M

Output: Optimized encoder network est_ϕ , policy network π_θ

Initialize encoder network est_ϕ , decoder network dec_φ , policy network π_θ and value function V_ψ

Initialize advantages $A_t = 0$ and value targets $V_t = 0$

for each epoch $e = 1$ to E do

for each mini-batch $b = 1$ to M do

 Initialize observation o_0 , obtain frame stack observation o_0^H and collect rollouts

for each step $t = 1$ to T do

 Compute linear velocity v_t and latent features $z_t \sim est_\phi(\cdot | o_t^H)$

 Compute action $a_t \sim \pi_\theta(\cdot | o_t^H, v_t, z_t)$

351 Execute a_t in the environment and observe $o_{t+1}, o_{t+1}^H, r_t, d_t$

 Compute advantage estimate \hat{A}_t , value target \hat{V}_t linear velocity target \hat{v}_t and next step observation target \hat{o}_{t+1} ; Update advantages $A_t = \rho_t A_{t-1} + \hat{A}_t$ and value targets $V_t = \gamma V_{t-1} + \hat{V}_t$

end

 Perform multiple PPO updates using A_t and V_t to optimize π_θ and V_ψ and estimator updates using o_{t+1}, z_t and v_t to optimize est_ϕ and dec_φ ;

end

end

return the optimized policy network π_θ and the encoder policy network est_ϕ ;

Algorithm 1: Training algorithm

352 C GENERALIZATION

353 This section presents a series of experimental images that vividly illustrate our algorithm’s intricate
 354 implementation details and robust generalization capabilities across various initial poses for robotic
 355 walking tasks, shown in the Fig. 6. We have achieved smooth locomotion across a spectrum of initial postures, as evidenced by the images depicting the robot’s initial stance.

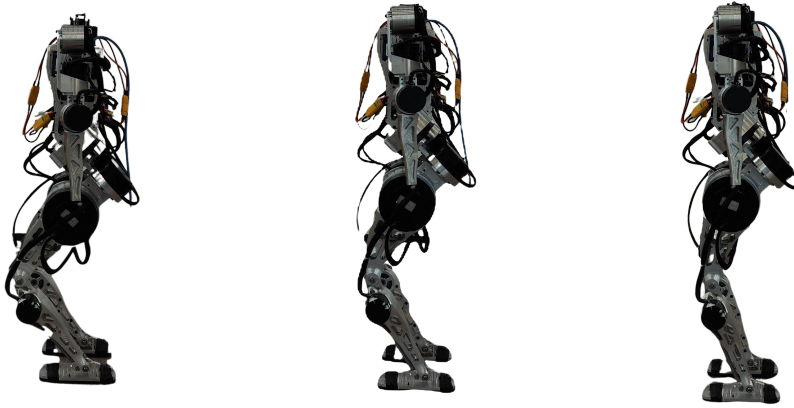


Figure 6: Training with diverse initial poses: An illustrative analysis of robotic locomotion across varied starting configurations.

356

357 Within the simulated environment, the Fig. 7 display the robot's upright and stable gait, underscoring
 358 the algorithm's exceptional precision in control within the virtual platform. We have further sub-
 359 stantiated the algorithm's reliability, shown in Fig. 8, and practical utility by advancing to real-world
 360 scenarios. The deployment of our algorithm on an actual robot has endowed it with the ability to
 361 navigate in various straight-legged postures, as illustrated by the images portraying the robot's com-
 362 mendable equilibrium and stability. This rigorous process is a testament to the algorithm's resilience
 363 and adaptability.

364 These outcomes furnish compelling validation for our algorithm's ongoing refinement and appli-
 365 cation, simultaneously presenting innovative perspectives and methodologies that hold significant
 366 promise for future research within related domains.

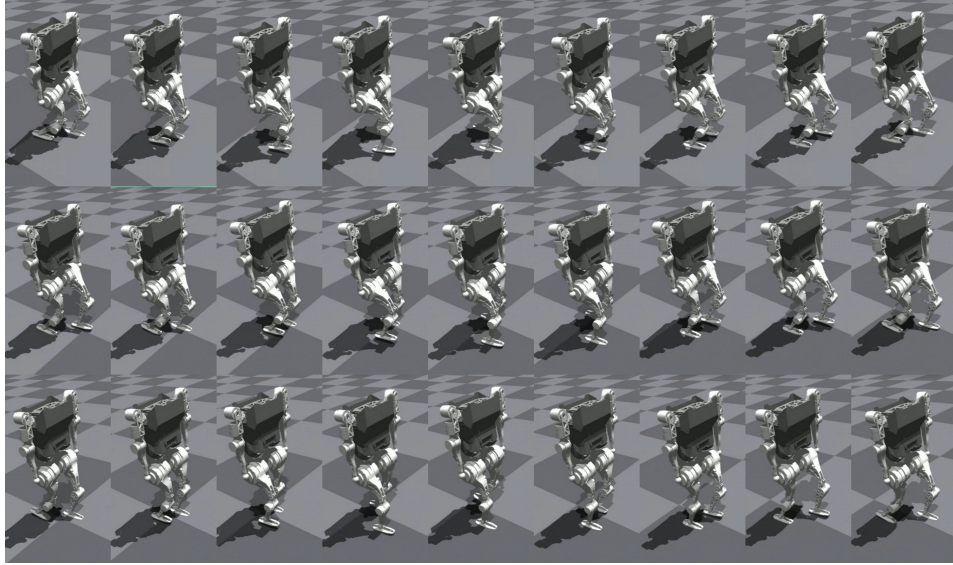


Figure 7: Diverse initial poses in simulation: A testament to our algorithm's robustness and stability.



Figure 8: Simulated locomotion across diverse initial poses: Demonstrating algorithmic adaptability in sim-to-real transitions.

D REAL-MACHINE EXPERIMENTATION

We conducted multiple experiments on the actual machine, testing our policy on various terrains, including grasslands, wire-strewn ground, slopes, and stairs. Our robot's traversal across grasslands is a testament to its ability to handle the soft and unpredictable ground, while its passage through areas with wireless showcases its resilience against obstacles that could impede movement. The robot's ascent on slopes highlights its dynamic balance and the algorithm's capacity to adjust to inclines that require precise foot placement and torque control. Most notably, the robot's ability to climb and descend stairs indicates our algorithm's advanced control mechanisms, ensuring that each step is calculated for maximum efficiency and safety. The images reveal a robot that is not just mobile but one that can adapt to and stabilize on a wide array of environmental conditions, thereby proving the algorithm's robustness and stability in a comprehensive sim-to-real context.

These visual records are more than just demonstrations of our robot's physical capabilities; they are evidence of the sophisticated algorithms that enable it to interact intelligently with its surroundings, providing a solid foundation for further research and development in robotics.



Figure 9: The sequence of images presented illustrates the diverse terrains our robot navigates with proficiency, ranging from the soft contours of grasslands to the challenging unevenness of wire-strewn areas, the inclines of varying gradients, and the ascents and descents of stairs. Each scenario, captured in a vertical progression from top to bottom, demonstrates not only the robot's adaptability but also its ability to maintain equilibrium on surfaces that demand different levels of traction and stability.