

Architecture Is All You Need: Diversity-Enabled Sweet Spots for Robust Humanoid Locomotion

*Blake Werner, *Lizhi Yang, Aaron D. Ames

Abstract—Robust humanoid locomotion in unstructured environments requires architectures that balance fast low-level stabilization with slower perceptual decision-making. We show that a simple layered control architecture (LCA), a proprioceptive stabilizer running at high rate, coupled with a compact low-rate perceptual policy, enables substantially more robust performance than monolithic end-to-end designs, even when using minimal perception encoders. Through a two-stage training curriculum (blind stabilizer pretraining followed by perceptual fine-tuning), we demonstrate that layered policies consistently outperform one-stage alternatives in both simulation and hardware. On a Unitree G1 humanoid, our approach succeeds across stair and ledge tasks where one-stage perceptual policies fail. These results highlight that architectural separation of timescales, rather than network scale or complexity, is the key enabler for robust perception-conditioned locomotion.

I. INTRODUCTION

Robust humanoid locomotion over mixed and unstructured terrain is a task as old as the platform itself, while still an unsolved problem. Sensing of terrain is partial and noisy, contact events are discontinuous, and controllers must react faster than perception can resolve in detail. Decades of practice in guidance–navigation–control (GNC) suggest a simple lesson: robustness emerges when fast, low-level stabilization is paired with slower, longer-horizon navigation. The canonical example is aerospace GNC [1], [2]: a slow, semantic guidance layer chooses where to go; an intermediate-rate trajectory-generation layer turns goals into feasible references; and a fast feedback control layer tracks those references and rejects disturbances. The same pattern, “slow and flexible” above “fast and rigid,” with well-defined interfaces, recurs across robotics and biological sensorimotor systems [3], [4].

This work takes this observation to its logical extreme and argues that, for high-dimensional perception-conditioned control problems, layered control architecture (LCA) [5]–[9] is the primary driver of robustness. Sophisticated models, learned world representations, or intricate reward shaping help to get the maximum absolute performance, but are not necessary for task success when the stack itself is well-posed. In a well-posed LCA, information flows through narrow interfaces: references descend (planner → controller) while tracking error or status ascends (controller → planner). Crucially, layers operate at different time scales—a design that

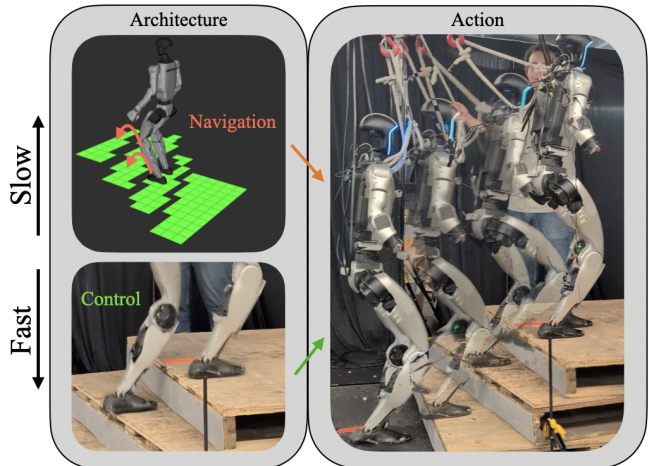


Fig. 1. A humanoid robot trained to traverse complex terrain through use of a combination perception information and fast proprioception information. Using this input effectively requires the use of structured architecture in order to produce performant and robust results.

both reduces computational burden and improves robustness by letting each layer specialize where it is most effective [5].

The separation of layers in a LCA, together with heterogeneous objectives and information, enables “diversity-enabled sweet spots” (DeSS) [10]: the combined stack can outperform any single monolithic component tuned in isolation. For perception-conditioned humanoid walking, the LCA framing implies a minimal yet sufficient stack: (i) a compact, local-perception navigation encoder that updates at moderate rate to construct an latent space that reflects long-horizon terrain geometry, and (ii) a fast stabilizer that uses proprioception to condition upon this geometry and contend with contact variability. Our method instantiates exactly this two-layer core, with the guidance layer assumed given, aligning with the quantitative architectural principles in [5].

A. Contributions

This paper makes two central claims. First, robust locomotion necessarily requires a layered, multi-rate design: a reflexive controller that stabilizes with proprioception at high rate, and a navigation layer that updates more slowly from exteroceptive cues to set short-horizon trajectories. Second, there exists a minimal instantiation of the LCA that can perform complex robust locomotion tasks without the use of heavy machinery: no complex environment estimators, no mixed-integer footstep search, no world models, and

* denotes equal contribution. All authors affiliated with Caltech MCE. This research is supported in part by the Technology Innovation Institute (TII), BP p.l.c., and by Dow via project #227027AW.

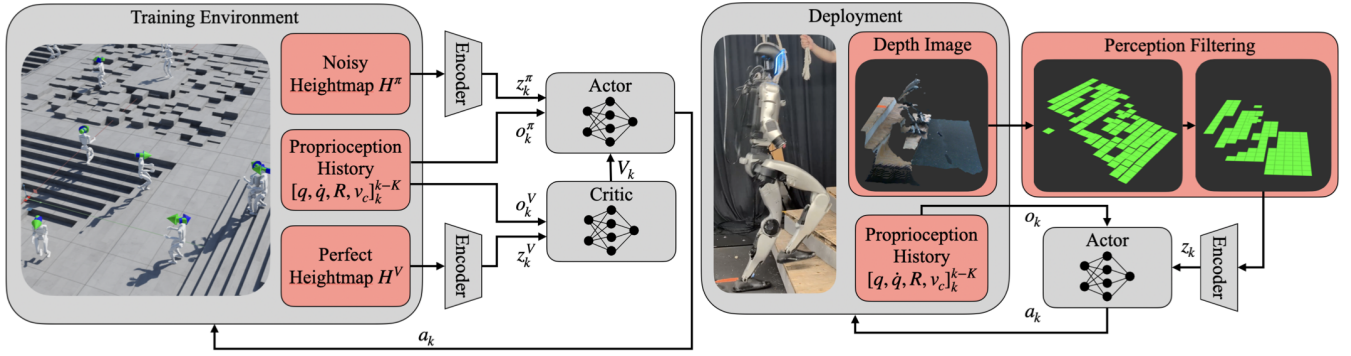


Fig. 2. Training and Deployment Overview: both actor and critic are two-stage architectures each with their own perception encoder. The actor receives noisy heightmap information, while the critic receives perfect information, and each receive proprioception history. During deployment, a depth image is filtered and passed through the trained encoder, and the actor combines this with the proprioception history to determine action.

no complex network architectures. The performance “sweet spot” arises from different parts of the control architecture taking information at different rates and information budgets rather than from any single sophisticated component.

Concretely, this paper realizes the smallest useful LCA for humanoids: (i) a fast low-level stabilizer (joint-space tracking with largely standard locomotion RL rewards) that runs purely on proprioception, and (ii) a slow navigation policy that consumes a compact local heightmap and allows the low-level to condition itself upon longer-horizon information. Training follows a two-stage curriculum: a blind phase (perception zeroed) that emphasizes stabilization, followed by perception phase that allows for more intelligent longer-horizon planning. This architecture is intentionally plain by design, yet we show it closes the gap to recent methods that rely on richer models or elaborate perception stacks.

Our contributions are as follows:

- **Architecture over complexity** We argue and empirically validate that robust humanoid locomotion *requires* a layered, multi-rate stack; the particular choice of sophisticated models is secondary.
- **Minimal LCA for robust humanoid locomotion** We instantiate a two-layer, two-stage pipeline with standard rewards and a compact local perception interface that performs well on complex locomotion tasks in unstructured terrain.
- **Architecture-isolating ablations** We vary network architectures and training curriculums. Results show that while model details produce only small performance differences, removing the layered structure causes large drops in success and tracking metrics.

B. Related Work

Two-stage training pipelines. Two-stage curricula appear in several locomotion settings, but for different architectural reasons. On sparse or precarious supports, works emphasize contact selection and balance under limited footholds, effectively prioritizing longer-horizon foot placement behavior before refining stabilization [11], [12]. In contrast, other works on challenging terrain follow a “blind-then-vision” strategy: first learn a robust proprioceptive stabilizer, then

condition that stabilizer on exteroceptive cues via a slower vision module [13], [14]. Both fall naturally into the LCA view: the first stage trains one layer in isolation (navigation or stabilization), while the second introduces the complementary layer and its interface. Blind stair-traversal and in general rough terrain works [15]–[20], can be seen as extreme instantiations where the navigation/perception layer is absent: such works are often very strong at stabilization but limited in foresight, relying on overfitting to a terrain type from training, and implicitly switching to this ‘mode’ when encountering this obstacle during deployment. Our design follows the second category, choosing to emphasize proprioceptive stabilization first before adding a conditioning vision module; however, doing so with only a minimal architecture consisting of just those two components.

Perception encoders. Across humanoid pipelines, perception does not feed torques directly; instead, visual depth or heightmaps are first encoded, then fused with proprioception downstream [21]. This preserves rate separation and prevents slow, noisy exteroception from contaminating fast feedback. Examples include perceptive internal models that fuse vision and state estimates for improved foothold selection [22], [23] and world-model approaches that learn latent representations to inform mid-rate decision making [24]. Our design follows the same pattern but keeps the encoder intentionally compact, simple, and local to maintain a narrow interface between the navigation layer and the stabilizer.

Model-based stepping and hybrid stacks. Classical “perceptive” footstep planners select contacts via mixed-integer optimization using sensed terrain [25]. Hybrid pipelines integrate such planners with model-free RL, letting the planner handle discrete contact choices while RL handles low-level tracking and robustness [26]. Whole-body methods with sequential contacts and adaptive motion optimization for dexterous humanoids also embody this decomposition: a mid-rate generator proposes feasible references, while a high-rate controller enforces stability and feasibility [27]. In all cases, the pipeline is explicitly layered: planning (navigation) up top, fast feedback below, with narrow reference/feedback channels—precisely the LCA pattern.

Student-teacher and distillation. Teacher-student pipelines leverage privileged information and rich supervision to train a capable teacher, then distill a deployable student with restricted observations [28]–[33]. From an LCA standpoint, such methods can partially sidestep architectural constraints during training by allowing the teacher to approximate harder, more global solutions before compressing capability into a smaller runtime policy. While highly effective, analyzing their architectural equivalence (e.g., whether the distilled student implicitly embeds a multi-rate decomposition) remains open; we regard this as complementary and leave a deeper treatment for future work.

II. METHODS

A. Optimization Analysis

To analyze the complex problem of perception-informed robot control, consider the following optimization:

$$\theta^* = \max_{\theta} \mathbb{E} \left[\sum_k \gamma^k r(s_k, a_k | \theta) \right], \quad (1)$$

where θ are the network parameters. This is the classical one-stage formulation of the reinforcement learning pipeline. Note that while this attempts to solve the global optimal control problem, it suffers from significant sensitivity to initial conditions [34] [35].

From an optimization perspective, solving the problems in sequence performs a different optimization, with less of the specified sensitivity. Let the parameters of the networks be divided as $\theta = [\theta_x, \theta_y]^T$, with ‘slow’ network parameters θ_y and ‘fast’ parameters θ_x . Note that in practice, these rates are more frequencies of the signals themselves, rather than the frequency of the controller (as they are all one network running at one speed). By solving the fast-rate optimization first, we solve

$$\theta_x^\dagger = \max_{\theta_x} \mathbb{E} \left[\sum_k \gamma^k r(s_k, a_k | \theta_x, \theta_{y,0}) \right], \quad (2)$$

wherein we maximize the reward conditioned on the fast rate controller parameters θ_x subject to an initial setting of the slow parameters $\theta_{y,0}$. In practice, since we will be removing perception from the optimization in stage one of the training, we remove the dependence on $\theta_{y,0}$, allowing for a simpler optimization more likely to find a satisfactory local maxima. In the second stage of the optimization then, we solve

$$\theta_x^*, \theta_y^* = \max_{\theta_x, \theta_y} \mathbb{E} \left[\sum_k \gamma^k r(s_k, a_k | \theta_x, \theta_y) \right] \quad (3)$$

$$\text{s.t. } \theta_{x,0} = \theta_x^\dagger. \quad (4)$$

Since we are optimizing over both variables, we perform the same optimization as the one-stage, so in sufficiently regular cases such as strictly concave reward landscapes we are guaranteed the same solution, i.e. $\theta_{\text{one-stage}}^* = \theta_{\text{two-stage}}^*$. However, in the highly nonconcave reward landscape that we work with, we note that by choosing a good initial condition from the first optimization, we are less susceptible to bad

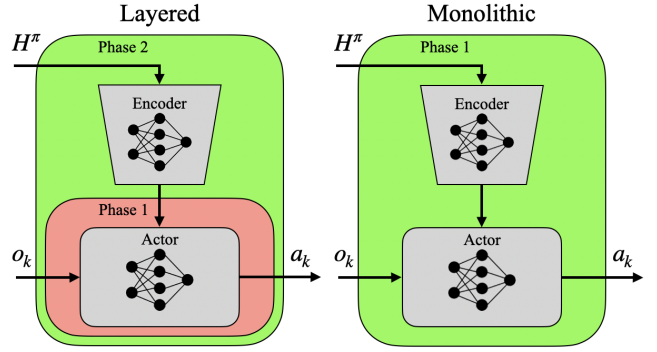


Fig. 3. Layered versus monolithic architectures: while the network architecture may be identical, training in two phases allows them to assume the layered control structure.

local maxima with large regions of convergence that would otherwise attract the optimization algorithm.

B. Observations and Normalization

We propose a minimal robust humanoid locomotion pipeline with the goal of illustrating our LCA hypothesis. Let q be joint positions, $\dot{q}$ joint velocities, g_b the projected gravity direction in body frame, ω_b the base angular velocity in body frame, a_{k-1} the last applied action, and u the commanded planar velocity. Let o_k be a K -length history of robot state information along with current velocity command and last action: $o_k = [q_k, q_{k-1}, \dots, q_{k-K}, \dot{q}_k, \dots, \dot{q}_{k-K}, \omega_{b,k}, \dots, \omega_{b,k-K}, g_{b,k}, \dots, g_{b,k-K}, u, a_{k-1}]$.

a) *Actor observation:* Our actor observation is a concatenation of a the K -length history of robot state information and the current velocity command and heightmap: $o_k^\pi = [o_k, H^\pi]$, where $H^\pi \in \mathbb{R}^{11 \times 11}$ is a *noisy, sparse* heightmap covering 1.0, m $\times$ 1.0, m around the robot (robot-centric frame). Note that with a nominal max velocity range of ± 0.6 m/s, and the set step period of 0.4s, the robot will take about two steps to move from its current location to that at the edge of the map. Therefore, the map encodes some temporal information (ground height where the robot will be in the future) despite the use of only current heightmap information. Additionally, we do not incorporate perception delays or latency for simplicity and consistency with other methods, but we note that the heightmap signal changes relatively much more slowly than the proprioceptive information. Finally, by using a history of states, we can capture transient and higher-order behaviors than would be allowed by strictly using the current state.

b) *Critic observation:* To construct our critic observation, we use similar information to the actor with the addition of the world-frame body velocities and a larger and more accurate heightmap with zero noise covering 1.5m $\times$ 1.5m around the robot. Giving the critic correct ground height information allows for a more accurate advantage function estimate, and the larger heightmap size allows the critic to see further into the ‘future’. In total, the observation is $o_k^V: [o_k, v_{\text{base}}, H^V]$.

Both heightmaps are normalized by subtracting the grid mean (cellwise) and clipping to $[-1, 1]$. Zero-centering re-

moves steady-state sim-to-real offsets such as those caused by changed camera mounting and compliance or different motor characteristics and simplifies biases in the MLP during the two-stage training.

C. Network Architecture

Our network architecture consists of two main components: the perception encoder, a network that takes the perception information and encodes it in a latent representation usable by the main actor network, and the primary actor network that uses a combination of the latent perception information and the standard robot proprioception to determine the robot's actions. Note that in our studies, we consider multiple choices for both the encoder network and the actor network to show the minimal underlying benefits of the actual implementation, instead highlighting the benefit of the layered architecture itself.

Our choice of encoder is either a small CNN or MLP mapping $H \in \mathbb{R}^{N \times N}$ to an embedding $z_H \in \mathbb{R}^{d_H}$. By ablating this to an MLP, we see if the spatial encoding characteristic of a CNN performs better than a simpler model, even on the small scale of the 11x11 heightmap. A similar network is used to encode the perception information sent to the critic network, the only difference being the larger size of the input.

The actor network, where we consider both the LSTM and MLP network architectures takes a concatenation of proprioception and perception information and outputs actions a_t as position setpoints, which are then tracked by joint-level PD controllers.

D. Rewards

We construct a number of rewards designed for our specific task to guide training, allowing for feasible performance on all of the proposed architectures, where we keep these rewards consistent. We use the following notation: we notate feet $i \in \{L, R\}$, the contact indicator as $C_i(k) = \mathbf{1}(\max_j \|\mathbf{F}_i^{(j)}(k)\|_2 > F_*)$ with $F_*=1\text{N}$, the planar foot velocity as $\mathbf{v}_i^{xy}(k)$, foot pitch as $\theta_{i,\text{pitch}}(t)$, foot height as $z_i(k)$, and phase as $\phi_i(k)$.

a) Phase-contact consistency reward: For $\tau=0.55$, $\varepsilon=5 \times 10^{-3}$, let the stance intent be

$$s_i(t) = \mathbf{1}(\phi_i(k) < \tau \vee \|\mathbf{u}_{\text{cmd}}(k)\|_2 < \varepsilon), .$$

Detected contact is $c_i(t) = C_i(k)$. The reward is the XNOR agreement:

$$\begin{aligned} r_{\text{phase}}(k) &= \sum_{i \in \{L, R\}} \mathbf{1}(c_i(k) = s_i(k)) \\ &= 2 - \sum_{i \in \{L, R\}} |c_i(k) - s_i(k)|. \end{aligned}$$

Standing case. When $\|\mathbf{u}_{\text{cmd}}(t)\|_2 < \varepsilon$, we have $s_L = s_R = 1$, so r_{phase} rewards double support and acts as the standing reward. We therefore do not include a separate standing term.

b) Foot-strike cost: Penalize lateral ground-reaction forces (scuffs, edge kicks):

$$r_{\text{strike}} = \sum_{i \in \{L, R\}} \|\mathbf{F}_i^{xy}\|_2, \quad \mathbf{F}_i^{xy} = \begin{bmatrix} F_i^x & F_i^y \end{bmatrix}.$$

c) Feet sliding cost: Suppress planar slip during stance:

$$r_{\text{slide}} = \sum_{i \in \{L, R\}} C_i(k) \|\mathbf{v}_i^{xy}(k)\|_2^2.$$

d) Feet orientation (flatness) cost: Encourage flat feet in contact with smooth saturation:

$$r_{\text{orient}} = 1 - \exp\left(-k_\theta \sum_{i \in \{L, R\}} C_i(k) |\theta_{i,\text{pitch}}(k)|\right), \quad k_\theta=25.$$

e) Feet clearance (swing height) cost: Penalize deviation from target swing height only when *not* in contact:

$$r_{\text{clear}} = \sum_{i \in \{L, R\}} (1 - C_i(k)) \left(\frac{z_i(k) - h_i^*(k)}{h_{\text{scale}}} \right)^2 g_i(k),$$

where $h_i^*(k)$ is the nominal swing height (collapsing to foot thickness near zero command), $g_i(k) = \tanh(\kappa \|\mathbf{v}_i^{xy}(k)\|)$ gates by step activity, and h_{scale} normalizes units.

f) Total reward: We combine the terms with positive weights and subtract the penalties from the standard locomotion rewards:

$$r = r_{\text{locomotion}} + 0.5 r_{\text{phase}} - r_{\text{strike}} - 0.2 r_{\text{slide}} - r_{\text{orient}} + r_{\text{clear}}.$$

E. Two-Stage Curriculum

Our training curriculum consists of two stages, wherein the robot first learns to traverse complex terrains without perception information, yielding a good baseline along with stabilization capabilities in order to deal with unseen obstacles or perturbations, then is given heightfield information, allowing the robot to learn longer-horizon behavior.

a) Stage 1 (blind stabilization): We set $H \equiv 0$ for the actor (though the critic is still given full information), and train in an environment made up of a quarter respectively of up-stairs, down-stairs, uneven terrain, and flat terrain tasks.

b) Stage 2 (perception-critical): Re-enable H for the actor. This allows the robot to make longer-horizon plans based on the local terrain, or put differently, condition the blind policy on the perceived surroundings.

F. Perception Filtering

During deployment, we perform a few stages of filtering for our perception stack in order to curate our data in a way that is usable by the policy. Note that while other works have used more complex perception filtering pipelines such as U-Nets and Transformers [23] [13], ours is intentionally simple, robust, and requires minimal tuning. Our input is a noisy, dense, depth image from the concatenation of two depth images. We then perform the following steps.

a) *Downsampling*: We aggregate the dense pointcloud to an 11×11 grid over $1.0, \text{m} \times 1.0, \text{m}$ by taking the minimum of each of the valid point heights in each cell. While this method does make the downsampling more sensitive to noise, the minimum value approximation allows for a more correct height estimate in situations such as stair occlusions, where the higher stair occludes the lower one, but the occluded values should be mapped to the lower stair height.

b) *Outlier Rejection*: We then compute the mean μ_z and standard deviation σ_z of the grid heights, then clamp outliers to the mean value. Here, we consider outliers to be cells (g_x, g_y, g_z) such that

$$|g_z - \mu_z| \geq \gamma \sigma_z \quad (5)$$

where γ is a tuned parameter. While the prior step deals with disturbances and outliers at the point level, this helps to deal with outliers at the grid cell level, such as the robot’s own legs and small anomalies in the terrain.

c) *Zero-mean and Quantize*: We then subtract μ_z from all the heightmap values in order to zero-center them and clip each to $[-1, 1]$ as the observation requires. Finally, we quantize to buckets corresponding to ‘steps’ of 5cm. This is the smallest ground height perturbation in simulation, and the stabilization of the policy seems robust to smaller perturbations.

III. SIMULATION EXPERIMENTS

A. Training

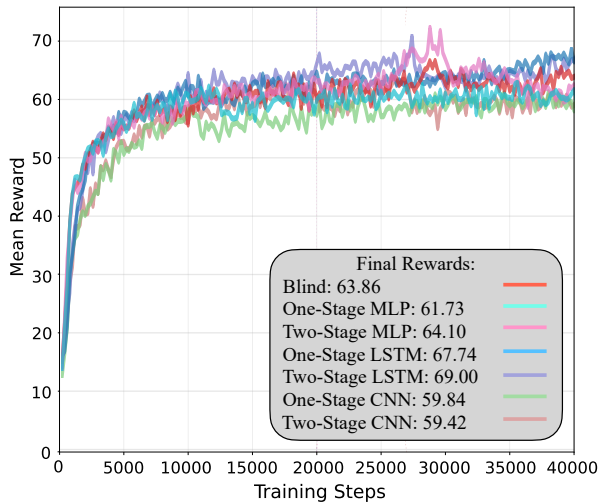


Fig. 4. Training rewards from each of the policies. We see that in training, all the policies perform largely identically; we believe that the small deviations may be a function of the network architecture of that component, such as the LSTM’s ability to store a hidden state or the CNN’s ability to reason more spatially, but may also simply be products of randomness.

a) *Setup*: We train all policies in IsaacSim on a single RTX 4090 with 4096 parallel environments. Each training batch is drawn from a balanced mixture of tasks: stair ascent (25%), stair descent (25%), uneven terrain (25%), and flat ground (25%), all trained using the asymmetric actor critic algorithm [36] to 40000 steps.

b) *Policies*: We compare seven variants: a blind baseline (no exteroception throughout), three one-stage perception-informed models (vision available for the full curriculum), and three two-stage models (blind in the first half of training, perception introduced in the second half). All actors share the same backbone sizes and differ only in network and encoder architectures: the actor is either an MLP or an LSTM with hidden layers $\{512, 256, 128\}$; the perception encoder is either a CNN (3×3 , stride 1) or an MLP, both with hidden layers $\{256, 256\}$. The critic is an MLP with the same hidden sizes and receives privileged inputs: a height scan of $1.5 \times 1.5 \text{m}$ at 0.1m resolution, joint states, base orientation, and CoM velocities. Rewards and observation normalizations are held fixed across policies so that differences reflect architecture and curriculum, not reward shaping.

c) *Metrics*: We report: (i) Success rate—the fraction of episodes that time out (task completed) without a fall or intervention; (ii) Contacts per step—the number of high-force lateral foot–environment impacts per robot step (threshold $> 100 \text{N}$), normalized by step count; and (iii) Tracking error—the mean ℓ_2 difference between commanded and measured base velocity, reported in cm/s.

d) *Protocol*: For each policy we roll out 500 simulated units, each performing 3 episodes of 1000 steps. Stair risers are uniformly randomized and treads set per condition; uneven terrain uses block fields with specified height ranges; and uniform noise of fixed magnitude is added to the heightfield observation. Parameter ranges are summarized in Table II (units in meters).

e) *Results*: On the medium (in-distribution) setting, all policies achieve near-parity; residual differences are within run-to-run variability. However, out-of-distribution (OOD) effects are more revealing:

Stairs (OOD) All models remain reasonably strong—stairs are structured and thus easier to “overfit.” However, the two-stage variants exhibit roughly $3 \times$ lower contacts/step than their one-stage counterparts and are closer to the blind baseline in this metric. A plausible mechanism is that, under noisy exteroception, two-stage policies fall back to the robust blind stabilizer learned in stage 1, whereas one-stage policies rely more heavily on the heightfield for both planning and stabilization, leading to occasional poor foot placements. Tracking error shows a smaller but consistent improvement in the same direction.

Uneven terrain (OOD) Here the differences are pronounced: the two-stage policies outperform one-stage by ≈ 10 percentage points in success on average, with corresponding reductions in contacts/step. Because the terrain is unstructured and harder to memorize, robustness requires both fast stabilization and longer-horizon placement—capabilities that are explicitly separated and co-trained in the two-stage pipeline but entangled in the one-stage models.

TABLE I. Simulation results across two tasks. Metrics: success rate ($\uparrow$), contacts per step ($\downarrow$), tracking error in cm ($\downarrow$).

Policy	Stairs			Uneven Terrain		
	R_{succ} (% , $\uparrow$)	Contacts/step (% , $\downarrow$)	Track (cm/s, $\downarrow$)	R_{succ} (% , $\uparrow$)	Contacts/step (% , $\downarrow$)	Track (cm/s, $\downarrow$)
<i>Medium difficulty</i>						
Blind	98.30	0.85 ± 0.20	13.40 ± 1.43	97.86	2.28 ± 0.34	13.90 ± 1.44
One-Stage MLP	99.00	0.64 ± 0.16	14.10 ± 1.39	98.00	1.66 ± 0.31	15.30 ± 1.45
Two-Stage MLP	97.90	1.04 ± 0.18	13.30 ± 1.42	98.60	0.80 ± 0.15	13.40 ± 1.35
One-Stage LSTM	97.40	0.55 ± 0.20	13.10 ± 1.36	98.80	0.54 ± 0.15	13.30 ± 1.37
Two-Stage LSTM	99.40	0.26 ± 0.11	13.50 ± 1.29	99.40	0.34 ± 0.12	14.40 ± 1.32
One-Stage CNN	98.90	0.99 ± 0.23	13.50 ± 1.49	97.80	2.74 ± 0.38	14.80 ± 1.54
Two-Stage CNN	98.00	0.79 ± 0.26	13.60 ± 1.34	98.50	2.79 ± 0.39	15.70 ± 1.54
<i>Hard difficulty</i>						
Blind	93.08	0.81 ± 0.20	15.10 ± 1.56	64.63	2.79 ± 0.43	17.70 ± 2.05
One-Stage MLP	92.80	1.59 ± 0.45	16.10 ± 1.60	61.60	3.36 ± 0.54	18.20 ± 1.92
Two-Stage MLP	90.48	0.56 $\pm$ 0.14	13.40 $\pm$ 1.41	70.96	3.10 $\pm$ 0.55	17.70 $\pm$ 2.00
One-Stage LSTM	85.33	3.24 ± 0.47	17.20 ± 1.96	58.02	3.10 ± 0.52	18.10 ± 2.01
Two-Stage LSTM	95.01	0.78 $\pm$ 0.25	16.10 $\pm$ 1.64	72.36	3.06 $\pm$ 0.51	17.60 $\pm$ 1.92
One-Stage CNN	96.10	2.15 ± 0.39	15.30 ± 1.68	63.93	5.45 ± 0.69	19.10 ± 2.18
Two-Stage CNN	98.40	0.83 $\pm$ 0.20	14.47 $\pm$ 1.42	71.95	5.68 $\pm$ 0.86	19.10 $\pm$ 2.12

TABLE II. Environment parameters for stairs and uneven terrain.

Difficulty	Stairs			Uneven Terrain	
	Height	Depth	Noise	Height	Noise
Medium	0.14–0.18	0.31	0.10	0.05–0.20	0.10
Hard	0.16–0.20	0.23	0.30	0.05–0.40	0.30

TABLE III. Proprioception observation terms o_{name} with noise, scaling, and history length applied.

Observation	Formula / Description
$o_{\text{base ang vel}}$	$\omega_b \in \mathbb{R}^3$, base angular velocity in body frame. Noise $\mathcal{U}(-0.2, 0.2)$, scale 0.25.
$o_{\text{projected gravity}}$	$g_b \in \mathbb{R}^3$, gravity vector projected in body frame. Noise $\mathcal{U}(-0.05, 0.05)$.
$o_{\text{velocity commands}}$	$u = (u_x, u_y, u_\omega)$, commanded base velocity, scale (2.0, 2.0, 0.25).
$o_{\text{joint pos}}$	$q - q^{\text{default}}$, joint positions relative to defaults. Noise $\mathcal{U}(-0.01, 0.01)$.
$o_{\text{joint vel}}$	$\dot{q} - \dot{q}^{\text{default}}$, joint velocities relative to defaults. Noise $\mathcal{U}(-1.5, 1.5)$, scale 0.05.
o_{actions}	a_{k-1} , last applied action.
$o_{\text{phase obs}}$	gait phase $\phi \in [0, 1]$ with standing detection, period 0.8.

B. Hardware Experiments

To verify our hypothesis on hardware, we deploy a subset of our policies on the G1 Humanoid robot. The perception stack is run on board using the robot’s Jetson Orin NX 16GB, and the RL controller is run on a Framework Laptop 13 with AMD Ryzen 7 7840u, which can either be off-board or strapped to the robot for a complete self-contained hardware stack. Perception data is provided by two Intel Realsense D435 cameras, one on the back of the hips, and one mounted on the chest, both pointing down. The lower hip camera allows for vision between the legs and behind the robot, while the upper chest camera allows for vision further in front of the robot. Together, they have a near-complete field of view of a 1.4m square around the robot, except for small

TABLE IV. Nominal reward terms and weights for humanoid locomotion.

Reward	Formula
$r_{\text{track lin vel xy exp}}$	$1.0 \cdot \exp\left(-\frac{\ v_{xy}^{\text{base}} - v_{xy}^{\text{command}}\ ^2}{0.25}\right)$
$r_{\text{track ang vel z exp}}$	$1.0 \cdot \exp\left(-\frac{(u_z - \omega_z)^2}{0.25}\right)$
$r_{\text{lin vel z l2}}$	$-2.0 \cdot v_z^2$
$r_{\text{ang vel xy l2}}$	$-0.05 \cdot (\omega_x^2 + \omega_y^2)$
$r_{\text{dof torques l2}}$	$-2.0 \times 10^{-5} \cdot \sum_j \dot{q}_j \tau_j $
$r_{\text{dof acc l2}}$	$-2.5 \times 10^{-7} \cdot \sum_j \dot{q}_j^2$
$r_{\text{dof vel l2}}$	$-1.0 \times 10^{-3} \cdot \sum_j \dot{q}_j^2$
$r_{\text{action rate l2}}$	$-0.01 \cdot \sum_a (a_t - a_{t-1})^2$
$r_{\text{undesired contacts}}$	$-1.0 \cdot \sum_{b \in B} \mathbf{1}(\max_t \ F_{t,b}\ > \theta)$
$r_{\text{contact no vel}}$	$-0.2 \cdot \sum_{b \in A} \ v_b\ ^2 \mathbf{1}(\text{contact}_b)$
$r_{\text{joint deviation hip}}$	$-1.0 \cdot \sum_{j \in J_{\text{hip}}} q_j - q_j^{\text{def}} $
$r_{\text{joint deviation arms}}$	$-0.5 \cdot \sum_{j \in J_{\text{arms}}} q_j - q_j^{\text{def}} $
$r_{\text{joint deviation torso}}$	$-1.0 \cdot q_{\text{waist}} - q_{\text{waist}}^{\text{def}} $
$r_{\text{height torso}}$	$-50.0 \cdot (z_{\text{root}} - 0.77)^2$
$r_{\text{feet clearance}}$	$+1.0 \cdot \sum_f (z_f - h_{\text{target}}(s_f))^2 \cdot (1 - \text{contact}_f)$
$r_{\text{feet slide}}$	$-0.2 \cdot \sum_f \ v_{f,xy}\ ^2 \mathbf{1}(\text{contact}_f)$
$r_{\text{phase contact}}$	$+0.5 \cdot \sum_f \mathbf{1}(\text{contact}_f = \text{stance}_f)$
$r_{\text{stand still}}$	$-0.1 \cdot \sum_j q_j - q_j^{\text{def}} \mathbf{1}(\ u\ < \epsilon)$
$r_{\text{feet flat}}$	$-1.0 \cdot \left(1 - e^{-25(\theta_{\text{pitch}}^L c_L + \theta_{\text{pitch}}^R c_R)}\right)$
$r_{\text{flat orientation l2}}$	$-1.0 \cdot (g_x^2 + g_y^2)$
$r_{\text{dof pos limits}}$	$-5.0 \cdot \sum_j \text{violation}(q_j)$
r_{alive}	$+0.15 \cdot \mathbf{1}(\neg \text{terminated})$
$r_{\text{termination penalty}}$	$-200.0 \cdot \mathbf{1}(\text{terminated})$

holes where the legs shadow the camera view. Due to the edge warping, we crop the center 1m x 1m area for depth. The cameras send depth images which are merged into a combined point cloud at a rate of 30Hz. The policy runs at 50hz, sending position setpoints to PD controllers at the joints operating at 1kHz.

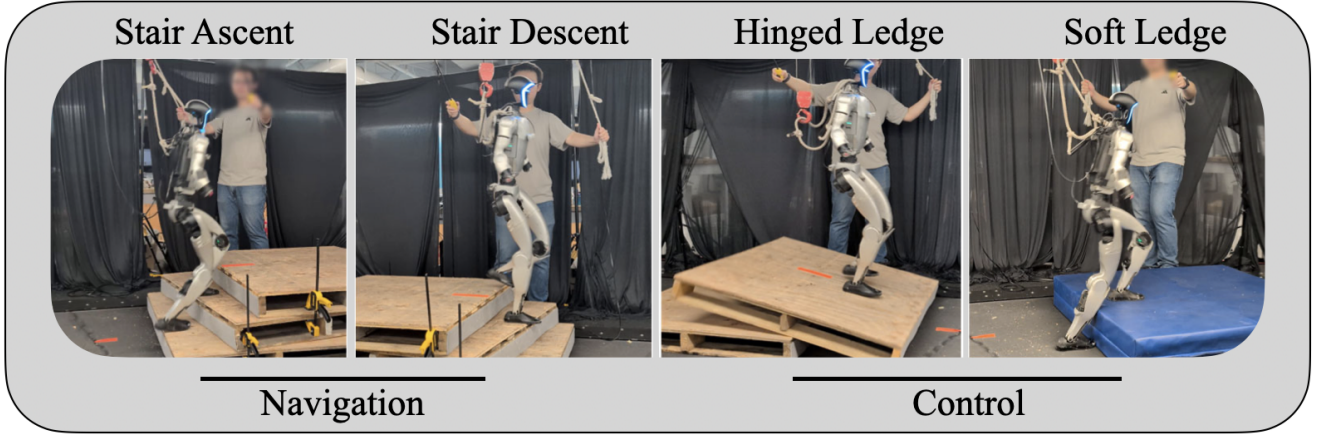


Fig. 5. The four hardware experiment tasks. The first two, stair ascent and stair descent, emphasize navigation, while the second two, hinged and soft ledge, emphasize control.



Fig. 6. Blind, MLP, and CNN depicted results. The monolithic policy is not included, as for some tasks it was never successful.

TABLE V. Four tasks. Success rate ($\uparrow$) and tracking error in m ($\downarrow$).

Policy	Stair Ascent		Stair Descent		Hinged Ledge		Soft Ledge	
	R_{succ} (% , $\uparrow$)	Track (m, $\downarrow$)	R_{succ} (% , $\uparrow$)	Track (m, $\downarrow$)	R_{succ} (% , $\uparrow$)	Track (m, $\downarrow$)	R_{succ} (% , $\uparrow$)	Track (m, $\downarrow$)
Blind	3/5	0.288	2/5	0.137	5/5	0.037	5/5	0.063
One-Stage MLP	1/5	0.000	1/5	0.000	0/5	–	0/5	–
Two-Stage CNN	4/5	0.175	5/5	0.228	5/5	0.059	5/5	0.167
Two-Stage MLP	4/5	0.045	5/5	0.163	5/5	0.199	4/5	0.113

a) Hardware tasks: We evaluate on four hardware tasks designed to probe different components of the stack: stair ascent, stair descent, hinged ledge, and soft ledge. The stair tasks comprise a short flight of three steps (riser ≈ 18 cm) with small landings (~ 20 cm) and a horizontal skew of $\sim 25^\circ$. This geometry forces careful toe/heel placement and weight transfer—missteps induce lateral perturbations—so these tasks primarily stress navigation (longer-horizon foot-

step/velocity planning). The ledge tasks use a 36 cm elevation change with transient compliance: the hinged variant is a plank balanced on a pivot that tips under load, and the soft variant lands onto a compliant gym mat. Perception sees a nominal ledge, but the dominant difficulty is the unmodeled, state-dependent disturbance at contact; these tasks primarily stress control (fast stabilization under transients).

b) Policies and trials: For each task we run five trials for each of four policies: (i) a blind baseline, (ii) a one-stage MLP, (iii) a two-stage MLP, and (iv) a two-stage CNN+MLP.

c) Metrics: We report two task-level metrics. Success rate counts trials that complete the task without a fall or human intervention. Precision measures repeatability: from a standing start we command 0.3 m/s forward, stop after task completion, and record the net lateral drift. We report the mean absolute deviation across the five trials to suppress fixed biases and emphasize stability and consistency.

d) Observations: First, the one-stage MLP underperforms across terrains despite identical training conditions. Empirically, footstep selection often appears reasonable, but stabilization degrades quickly, consistent with over-reliance on noisy heightfields for low-level control. Second, the blind policy transfers reasonably well and is particularly strong on the ledge (control-dominant) tasks, but fails more frequently on stairs where precise edge-aware placement is required (missed steps when descending; edge strikes when ascending). Finally, the two-stage policies (MLP and CNN+MLP) perform similarly and robustly on both navigation- and control-dominant tasks, supporting our central claim: once the layered structure is in place, the specific encoder and backbone choice is of lesser importance. For absolute peak performance, one could further tune architectures or add specialized modules, but our results indicate such complexity is unnecessary to achieve robust behavior on these tasks.

REFERENCES

- [1] H. Tsien, T. Adamson, and E. Knuth, "Automatic navigation of a long range rocket vehicle," *Journal of the American Rocket Society*, vol. 22, no. 4, pp. 192–199, 1952.
- [2] C. S. Draper and W. Wrigley, "Guidance-basic principles," *GUIDANCE AND NAVIGATION*, 1965.
- [3] M. R. Tucker, J. Olivier, A. Pagel, H. Bleuler, M. Bouri, O. Lamercy, J. d. R. Millan, R. Riener, H. Vallery, and R. Gassert, "Control strategies for active lower extremity prosthetics and orthotics: a review," *Journal of neuroengineering and rehabilitation*, vol. 12, no. 1, p. 1, 2015.
- [4] Y. Nakahira, Q. Liu, T. J. Sejnowski, and J. C. Doyle, "Diversity-enabled sweet spots in layered architectures and speed-accuracy trade-offs in sensorimotor control," *Proceedings of the National Academy of Sciences*, vol. 118, no. 22, p. e1916367118, 2021.
- [5] N. Matni, A. D. Ames, and J. C. Doyle, "Towards a theory of control architecture: A quantitative framework for layered multi-rate control," *arXiv preprint arXiv:2401.15185*, 2024.
- [6] "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of experimental biology*, vol. 202, no. 23, pp. 3325–3332, 1999.
- [7] U. Rosolia and A. D. Ames, "Multi-rate control design leveraging control barrier functions and model predictive control policies," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1007–1012, 2020.
- [8] U. Rosolia, A. Singletary, and A. D. Ames, "Unified multirate control: From low-level actuation to high-level planning," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6627–6640, 2022.
- [9] N. Csomay-Shanklin, "Layered control architectures: Constructive theory and application to legged robots," Ph.D. dissertation, California Institute of Technology, 2025.
- [10] Y. Nakahira, Q. Liu, T. J. Sejnowski, and J. C. Doyle, "Diversity-enabled sweet spots in layered architectures and speed-accuracy trade-offs in sensorimotor control," *Proceedings of the National Academy of Sciences*, vol. 118, no. 22, p. e1916367118, 2021.
- [11] H. Wang, Z. Wang, J. Ren, Q. Ben, T. Huang, W. Zhang, and J. Pang, "Beamdojo: Learning agile humanoid locomotion on sparse footholds," *arXiv preprint arXiv:2502.10363*, 2025.
- [12] C. Zhang, W. Xiao, T. He, and G. Shi, "Wococo: Learning whole-body humanoid control with sequential contacts," in *Conference on Robot Learning*. PMLR, 2025, pp. 455–472.
- [13] H. Duan, B. Pandit, M. S. Gadde, B. Van Marum, J. Dao, C. Kim, and A. Fern, "Learning vision-based bipedal locomotion for challenging terrain," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 56–62.
- [14] M. S. Gadde, P. Dugar, A. Malik, and A. Fern, "No more blind spots: Learning vision-based omnidirectional bipedal locomotion for challenging terrain," *arXiv preprint arXiv:2508.11929*, 2025.
- [15] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," in *Robotics: Science and Systems (RSS)*, 2021.
- [16] S. Chamorro, V. Klemm, M. d. L. I. Valls, C. Pal, and R. Siegwart, "Reinforcement learning for blind stair climbing with legged and wheeled-legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8081–8087.
- [17] R. Li, H. Wang, Q. Li, Z. Han, Y. Chu, L. Ye, W. Xie, and W. Liao, "Ctbc: Contact-triggered blind climbing for wheeled bipedal robots with instruction learning and reinforcement learning," *arXiv preprint arXiv:2509.02986*, 2025.
- [18] R. P. Singh, M. Morisawa, M. Benallegue, Z. Xie, and F. Kanehiro, "Robust humanoid walking on compliant and uneven terrain with deep reinforcement learning," in *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2024, pp. 497–504.
- [19] Y. Zhang, Z. Yu, X. Chen, Y. Du, Z. Zhou, and J. Gao, "Bipedal walking outdoors with a point-footed robot via reinforcement learning," in *2024 International Conference on Intelligent Robotics and Automatic Control (IRAC)*. IEEE, 2024, pp. 193–197.
- [20] C. Ji, D. Liu, W. Gao, and S. Zhang, "Robust and efficient walking of a bipedal humanoid robot via reinforcement learning," in *2025 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2025, pp. 381–388.
- [21] M. Su, Y. Jia, and Y. Huang, "Effects of prior knowledge for stair climbing of bipedal robots based on reinforcement learning," in *2024 International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2024, pp. 216–222.
- [22] J. Long, J. Ren, M. Shi, Z. Wang, T. Huang, P. Luo, and J. Pang, "Learning humanoid locomotion with perceptive internal model," *arXiv preprint arXiv:2411.14386*, 2024, submitted to ICRA2025. [Online]. Available: <https://arxiv.org/abs/2411.14386>
- [23] J. He, C. Zhang, F. Jenelten, R. Grandia, M. Bächer, and M. Hutter, "Attention-based map encoding for learning generalized legged locomotion," *Science Robotics*, vol. 10, no. 105, p. eadv3604, 2025.
- [24] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen, "Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning," *arXiv preprint arXiv:2408.14472*, 2024.
- [25] B. Acosta and M. Posa, "Perceptive mixed-integer footstep control for underactuated bipedal walking on rough terrain," *arXiv preprint arXiv:2501.19391*, 2025.
- [26] H. Su, H. Luo, S. Yang, K. Jiang, W. Zhang, and H. Chen, "Lipm-guided reinforcement learning for stable and perceptive locomotion in bipedal robots," *arXiv preprint arXiv:2509.09106*, 2025.
- [27] Q. Liao, T. E. Truong, X. Huang, G. Tevet, K. Sreenath, and C. K. Liu, "Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion," *arXiv e-prints*, pp. arXiv-2508, 2025.
- [28] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.
- [29] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.
- [30] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," *arXiv preprint arXiv:2406.10759*, 2024.
- [31] F. Wu, X. Nal, J. Jang, W. Zhu, Z. Gu, A. Wu, and Y. Zhao, "Learn to teach: Sample-efficient privileged learning for humanoid locomotion over real-world uneven terrain," *IEEE Robotics and Automation Letters*, 2025.
- [32] Y. Fan, T. Gui, K. Ji, S. Ding, C. Zhang, J. Gu, J. Yu, J. Wang, and Y. Shi, "One policy but many worlds: A scalable unified policy for versatile humanoid locomotion," *arXiv preprint arXiv:2505.18780*, 2025.
- [33] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang *et al.*, "Hover: Versatile neural whole-body controller

- for humanoid robots,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 9989–9996.
- [34] D. Picard, “Torch. manual_seed (3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision,” *arXiv preprint arXiv:2109.08203*, 2021.
- [35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [36] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, “Asymmetric actor critic for image-based robot learning,” in *14th Robotics: Science and Systems, RSS 2018*. MIT Press Journals, 2018.