Learning Terrain-Aware Bipedal Locomotion via Reduced-Dimensional Perceptual Representations

Guillermo A. Castillo, *Graduate Student Member, IEEE*, Himanshu Lodha, *Graduate Student Member, IEEE*, and Ayonga Hereid, *Member, IEEE*

***Abstract*—This work introduces a hierarchical strategy for terrain-aware bipedal locomotion that integrates reduced-dimensional perceptual representations to enhance the reinforcement learning (RL)-based high-level (HL) policies for real-time gait generation. Unlike end-to-end approaches, our framework leverages latent terrain encodings via a convolutional variational autoencoder (CNN-VAE) alongside reduced-order robot dynamics, optimizing the locomotion decision process with a compact state. We systematically analyze the impact of latent space dimensionality on learning efficiency and policy robustness. In addition, we extend our method to be history-aware, incorporating sequences of recent terrain observations into the latent representation to improve robustness. To address real-world feasibility, we introduce a distillation method to learn the latent representation directly from depth camera images and provide preliminary hardware validation by comparing simulated and real sensor data. We further validate our framework using the high-fidelity agility robotics (ARs) simulator, incorporating realistic sensor noise, state estimation, and actuator dynamics. The results confirm the robustness and adaptability of our method, underscoring its potential for hardware deployment.**

***Index Terms*—Humanoid robots, legged locomotion, reinforcement learning.**

I. Introduction

ONE of the main advantages of legged robots over their wheeled counterparts is their potential to navigate challenging and unstructured environments. The early stage of legged locomotion research focused on ***blind locomotion***, where robots were designed to move without real-time perceptual feedback from their environment. These systems relied heavily on preprogrammed movements and robust locomotion controllers to navigate their surroundings. However, as anyone who has observed the effortless grace of animals and humans to traverse rugged terrains can attest, there is an essential difference between simply moving and moving with awareness of one's environment, known as ***perceptive locomotion***.

Most existing work on reinforcement learning (RL)-based controllers for bipedal locomotion has focused on blind locomotion. The impressive robustness of the learned policies allows the robot to walk in challenging terrains, such as hills [1], slopes [2], and even flights of stairs [3]. The shift from blind to perceptive locomotion has enabled robots to see and respond to their environment in real time. Integrating visual sensors improves the stability and safety by allowing adaptive adjustments to gait to account for different terrains and environments. In particular, there has been a growing interest in integrating terrain information as part of the state feedback for training gait policies for legged locomotion, especially on quadruped robots.

Some of the first attempts to integrate terrain information in an RL policy for bipedal locomotion were implemented in simulations of physics-based animations. In [4], a reduced-character state and reduced-terrain state were used to train a policy to navigate terrains with steps and gaps in a simulation of a 2-D environment. This approach was extended [5] by using the full-height field map and character state in an end-to-end RL framework with a mixture of actor–critic experts updated through temporal difference learning. An extension to the 3-D case in the simulation was proposed by [6], where hierarchical deep RL was used to train a high-level (HL) policy that makes step target decisions based on high-dimensional inputs, including terrain maps or other suitable representations of the surroundings, and a low-level policy that learns to achieve robust walking gaits.

Xie et al. [7] and Singh et al. [8] addressed the challenge of walking on irregular terrains using preplanned footsteps obtained from the environmental height field. The RL policy then uses the foothold sequence and the robot's state to compute the joint target positions. A more effective terrain representation is presented in [9] using a sparse exteroceptive observation from raycasts along the vertical axis. This approach is efficient, but limited in the number of features it can capture.

Van Marum et al. [10] build upon the work in latent terrain representation for quadruped locomotion [11] to train an end-to-end policy to navigate a wide variety of terrains using noisy exteroception. Duan et al. [12] presented a vision-based RL framework for bipedal locomotion, showcasing robust locomotion over challenging terrains with the robot Cassie. A height map expressed in the robot's local frame is used to train an end-to-end RL locomotion policy for stairs and steps of different heights. The height map used to train in simulation is replaced by a height map predictor obtained from depth camera images and the robot state. Gadde et al. [13]

Received 10 September 2025; accepted 19 January 2026. This work was supported in part by the National Science Foundation under Grant FRR-21441568. Recommended by Associate Editor U. Rosolia. *(Guillermo A. Castillo and Himanshu Lodha contributed equally to this work.) (Corresponding author: Guillermo A. Castillo.)*

Guillermo A. Castillo and Himanshu Lodha are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: castillomartinez.2@osu.edu; lodha.11@osu.edu).

Ayonga Hereid is with the Department of Mechanical and Aerospace Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: hereid.1@osu.edu).

Digital Object Identifier 10.1109/TCST.2026.3664022

build upon [12] to replace the height map estimator with a perception encoder trained with a teacher–student approach. Gu et al. [14] and Wang et al. [15] presented an alternative to visual perception by integrating the terrain information only as privileged information and using the history of the observation data and an asymmetric actor–critic architecture [14] or a student–teacher approach [15] to replace the visual perception by encoded latent space that captures the terrain conditions along with the robot dynamics.

While these recent advancements push the boundaries of perceptive locomotion through sophisticated attention mechanisms that dynamically select terrain features [16], novel hybrid training paradigms [17], and advanced sensor fusion with internal models [18], our work provides a distinct and complementary contribution. These state-of-the-art methods focus on building complex integrated systems, often leveraging rich perception sources like LiDAR-based elevation maps [18] or innovating on the training algorithm itself [17].

Our work, in contrast, addresses a more fundamental and generalizable question: What is the principle of the minimal sufficiency for perceptual information in locomotion? Our primary contribution is the development of a new policy architecture and the systematic analysis of the information bottleneck between perception and action. We extend our analysis to history-aware autoencoders, which provide further evidence for our central hypothesis. Moreover, we introduce a distillation process that enables the policy to learn the same compact representation directly from multiple depth camera images—a critical step toward hardware deployment. We leverage a lightweight, hierarchical framework and a simple CNN-VAE not to build the most complex system, but as a precise tool to investigate the tradeoff between the dimensionality of the latent space and the resulting policy's performance.

This article extends our prior work, which first established a sample-efficient hierarchical framework for bipedal locomotion [19] and subsequently validated its real-world viability with successful zero-shot sim-to-real transfer on the digit robot [20]. While this proprioceptive-based controller proved robust to external disturbances, it was fundamentally "blind" and thus incapable of navigating unstructured terrain.

The primary contribution of this work is therefore the integration of a perception module into this proven hierarchical framework. Leveraging insights from our previous research on data-driven latent spaces [21], we introduce a learned, low-dimensional terrain representation that allows the policy to make informed, terrain-aware decisions. This addition bridges the critical gap from the blind disturbance rejection to agile, perceptive locomotion over complex terrains.

Therefore, in this work, we propose a perceptive bipedal locomotion framework that combines the versatility of *RL*-based policies for HL commands with the robustness of a low-level task-space controller and the effectiveness of an efficient latent representation of the terrain height map. One of the key contributions of our work is the systematic analysis of the impact of the dimension of the latent representation of the height map on the efficiency of the learning process, showing that a too small or too large dimension of the latent representation hurts sample efficiency. This analysis is further extended to history-aware perception, and we introduce a distillation method to learn the same compact representation directly from depth camera images, a critical step for real-world deployment. By focusing on the efficiency of the representation itself, our work provides concrete, empirical evidence that an optimal level of compression exists.

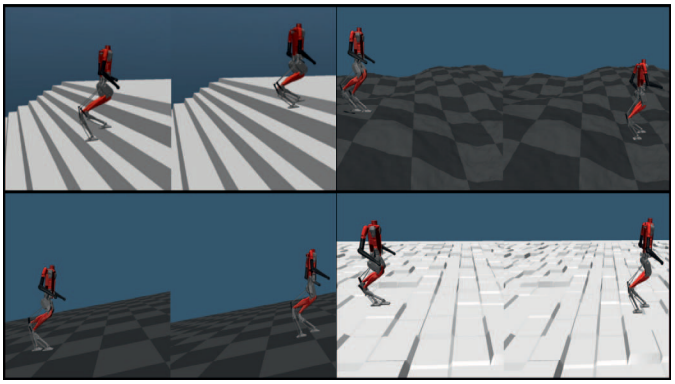

Fig. 1. Digit walking over challenging terrains (stairs, hills, slopes, and squares) using a terrain-aware locomotion policy.

The remainder of this article is organized as follows. Section II explains the supervised learning approach to encode a latent terrain representation with CNN-VAEs and discusses the importance of the latent space dimension. Section III introduces a hierarchical RL framework for terrain-aware locomotion. Section IV shows simulation results of the proposed framework with ablation studies and baseline comparisons, addressing real-world feasibility through a distillation process with depth images, realistic sensor noise, and additional tests with the high-fidelity agility robotics (ARs) simulator. Finally, Section V briefly concludes this article and discusses the future directions of our work.

II. Terrain Representation in Latent Space

In this section, we introduce our proposed method to learn an adequate representation of the terrain information around the robot that successfully captures the critical features of the ground. The goal is to design a robust locomotion policy, introduced in Section III that allows humanoid robots to navigate challenging terrains actively.

A. Terrain Data Collection

We use a local height map corresponding to the area of 2 m^2 in front of the robot to perceive the terrain around the robot. When using a resolution of 5 cm^2, the local terrain height map, $\mathbf{x}$, is represented by a matrix of size 20×40, resulting in a total of 800 elements.

The height map grid resolution choice is based on the width of the digit robot's feet, which is about 5 cm. It is also aligned with relevant works in the literature, where a grid resolution of 5–6.5 cm is used to capture features of different terrains effectively [10], [12], [22]. We show in Fig. 3 a sample of this local height map obtained from a simulation. Using all the elements of the local terrain height map matrix as input

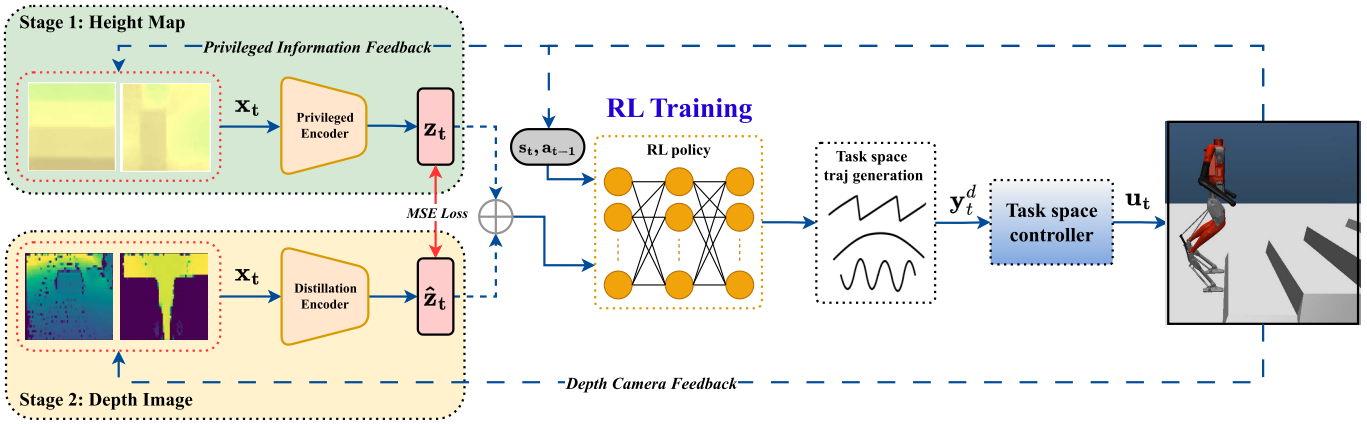

Fig. 2. Hierarchical structure of the proposed framework: an HL RL policy for gait planning trained with a multistage approach, and a low-level controller for trajectory tracking. The privileged encoder uses a CNN-VAE to encode the local height map to a reduced-dimensional latent variable to train terrain-aware perception locomotion policies. In Stage 2, a distillation process replaces the privileged information from the height map with the input from depth cameras by matching the latent representation obtained from these two exteroceptive sources.

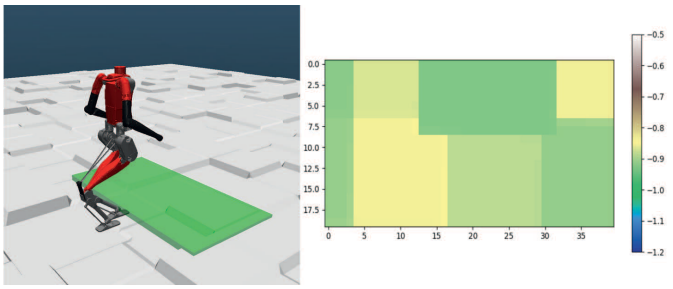

Fig. 3. World view of the local height map used to detect the terrain around the robot (left) with the corresponding local height map matrix (right) relative to the robot's base. The height map covers an area of 2×1 m at a 5-cm resolution.

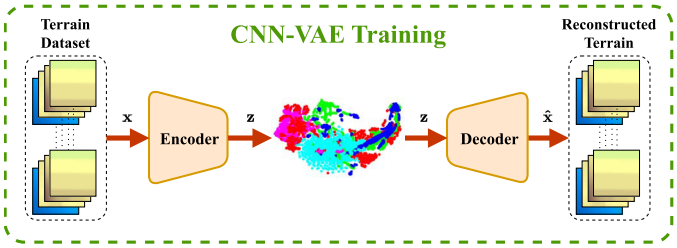

Fig. 4. CNN-VAE encodes the local height map to a reduced-dimensional latent variable, $\mathbf{z}$, used to train terrain-aware perception locomotion policies with the framework presented in Fig. 2.

for the locomotion policy would result in a significantly large neural network and, consequently, more parameters and larger inference time, even though many of the elements of the height map may not have useful information for the policy to produce effective gait actions.

To train a CNN-VAE of the terrain map, we use a customized simulation environment in MuJoCo [23] to create different terrain profiles, including sloped planes, hills, squared steps, and stairs with various configurations (up and down) and dimensions (width, depth, and height). We collect 60 000 samples of local height maps for each terrain type. To collect a diverse dataset of terrain height maps without the existence of a capable locomotion policy, we only simulate the kinematic motion of the robot around the terrain by updating the position of the robot's base in the simulation environment according to randomly sampled velocity while keeping the base height at a height that follows a normal distribution with a mean 0.92 m and standard deviation 0.1 m. Thus, the complete dataset $\mathbf{X}$ consists of 360 000 samples of local terrain height maps $\mathbf{x}$, i.e., $\mathbf{X} = \{\mathbf{x}^{(i)} | i \in [1\,360\,000]\}$.

B. Convolutional Variational Autoencoder

One of CNN-VAEs' primary advantages is their proficiency in handling higher dimensional data, such as large maps, and effectively compressing them into lower dimensional representations that capture the essential features and structures of the original data through a conditioned probability distribution.

In particular, as shown in Fig. 4, we use a CNN-VAE to encode the terrain height map into a reduced-dimensional latent variable $\mathbf{z} \in \mathbb{R}^m$ to reduce the dimension of terrain information used for locomotion. In this work, the *encoder* part of the CNN-VAE comprises three convolutional layers followed by two fully connected layers. The convolutional layers progressively reduce the spatial dimensions of the input while increasing the depth of the feature maps, with 32, 64, and 128 channels, respectively, each using a kernel size of 4, a stride of 2, and a padding of 1. After the convolutional layers, the output is flattened and passed through two fully connected layers. Specifically, these layers output the mean μ and variance σ of the prior distribution, both sized according to the predefined latent variable dimension m. Then, the latent random variable $\mathbf{z}$ can be expressed as a deterministic variable

$$\mathbf{z} = g_\theta(\epsilon, \mathbf{x}) \tag{1}$$

where $\mathbf{x}$ is the sample vector corresponding to the local height map, θ represents the learnable parameters (weights and biases) of the encoder network, $g_\theta(\cdot)$ is the encoder function parameterized by θ, and ϵ is an auxiliary random variable with an independent marginal probability distribution. In this article, the term latent random variable refers to the hidden

variables (the elements of the vector z) that are not directly observed but are instead inferred from the input data (the terrain height map). In a VAE, these variables are treated probabilistically, allowing the model to capture a distribution of the underlying terrain features. Therefore, if we choose ϵ to be the univariate Gaussian distribution $\mathcal{N}(0, 1)$, the latent random variable $\mathbf{z}$ is determined by

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{\epsilon}. \tag{2}$$

This method, known as the reparameterization trick, allows backpropagation through random sampling processes, which is essential to train VAEs through standard stochastic gradient descent methods. The encoder part of the CNN-VAE efficiently reduces the dimensionality of the input data and captures its essential features in a form conducive to generative tasks. By learning this compact latent representation, the CNN-VAE can effectively generate a probability distribution from which one can reconstruct the local height map samples and even generate new, unseen local height maps that share statistical properties with the training data.

C. Reconstruction of the Height Map

The *decoder* is the closed-form parameterized function that maps from the latent space back to the full-order state. This function is defined by

$$\hat{\mathbf{x}} = d_\phi(\mathbf{z}) \tag{3}$$

where ϕ represents the parameters of the decoder network, d_ϕ is the decoder function parameterized by ϕ, $\mathbf{z}$ is the encoded latent variable, and $\hat{\mathbf{x}}$ is the reconstruction of the original input data $\mathbf{x}$.

The CNN-VAE is trained by minimizing the standard β-VAE loss $\mathcal{L}$, which consists of reconstruction loss and the Kullback–Leibler (KL) divergence as the latent loss [24]. Then, the VAE loss is formulated as follows:

$$\mathcal{L} = \mathrm{MSE}\left(\mathbf{x}^{(i)}, \hat{\mathbf{x}}^{(i)}\right) + \beta D_{\mathrm{KL}}\left(q\left(\mathbf{z}^{(i)}|\mathbf{x}^{(i)}\right) \| p\left(\mathbf{z}^{(i)}\right)\right) \tag{4}$$

where $\hat{\mathbf{x}}^{(i)}$ is the reconstructed height map, $p(\mathbf{z}^{(i)})$ is the prior distribution parameterized by the Gaussian distribution ϵ, and $q(\mathbf{z}^{(i)}|\mathbf{x}^{(i)})$ is the posterior distribution of the latent variable $\mathbf{z}^{(\mathbf{i})}$ given $\mathbf{x}^{(i)}$. The autoencoder is trained using Adam optimizer [25] with a learning rate of 0.001 and a batch size B of 256. The autoencoder is trained for 40 epochs on a 12-core CPU machine with an NVIDIA RTX 2080 GPU.

There is no rule of thumb for the proper size of the latent state used to capture the encoder input's features fully. On the one hand, a latent variable of large dimension allows a better reconstruction of the local height map. Conversely, a smaller dimension of the latent variable enables more efficient encoding, resulting in more compact networks for the VAE and the locomotion policy. To analyze the tradeoff between these two properties, we conduct an ablation study with different values of the latent space dimension. In Fig. 5, we show the loss $\mathcal{L}$ during training of the CNN-VAE for different values of m. The latent dimensions 256, 128, and 64 show the most rapid decrease in loss, indicating efficient learning and improved ability to capture data features. The latent dimensions 64 and 32 clearly balance model complexity and learning efficiency. The smaller dimensions 8 and 4 exhibit a higher average loss, which means the model is not complex enough to capture all the necessary data features.

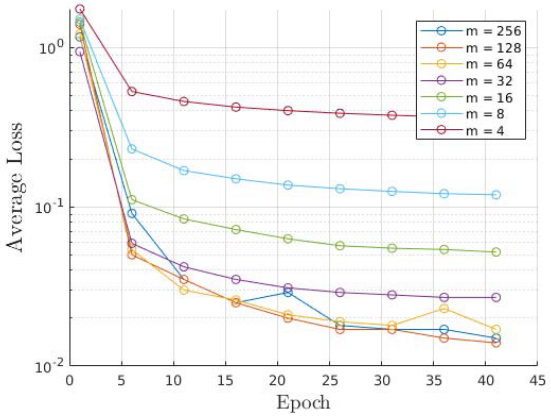

Fig. 5. Latent representation learned by the CNN-VAE for different dimensions of the latent variable m. Larger latent sizes (e.g., 64 and 128) converge faster and to a lower loss, indicating better reconstruction, showing diminishing returns for $m \geq 32$, whereas minimal latent dimensions (e.g., $m = 4$) show significant error. This suggests that overly compressing the terrain representation hurts accuracy, while moderately sized latent effectively balance compactness and fidelity.

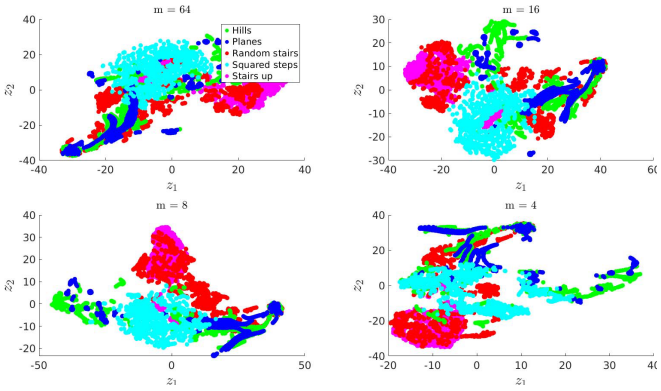

Fig. 6. t-SNE of the latent representation learned by the CNN-VAE for different dimensions of the latent variable $\mathbf{z}$. t-SNE visualization of the learned latent space for different latent dimensions m. For sufficiently large latent dimensions ($m \geq 16$), the latent vectors form well-separated clusters corresponding to distinct terrain types, indicating that the VAE has retained meaningful terrain features. In contrast, with a minimal latent ($m = 4$), the clusters—particularly for complex terrains like "squared steps"—are poorly formed, showing that important details are lost when the latent space is too limited.

In addition, we apply t-distributed stochastic neighbor embedding (t-SNE) to the latent representations of different dimensions to analyze the structure of the encoded data. These results, presented in Fig. 6, demonstrate that for $m \geq 16$, the latent space retains a consistent and meaningful structure, with clusters representing distinct terrain types. However, the structure deteriorates significantly for $m = 4$, particularly for the squared steps terrain. This terrain exhibits the highest degree of irregularity and complexity, is the most challenging to encode, and requires more features to capture its structure accurately. This behavior is expected because

more irregular terrains demand a higher dimensional latent space to retain their essential characteristics. This analysis highlights that a latent dimension as low as $m = 8$ can still effectively capture the key features of the terrain height map, achieving a balance between compactness and representational power. Consequently, we choose $m = 16$ as a good tradeoff between feature capturability and reconstruction error for the CNN-VAE during the training of our locomotion policy, described in Section III.

D. History-Aware Perception

Recent advancements in perceptive history for legged locomotion demonstrate that integrating historical data enhances robots' adaptability to changing environments, optimizing movement, and improving navigation efficiency. However, seamlessly combining historical and real-time data across multiple modalities could significantly increase the input dimension of end-to-end RL approaches. Therefore, analyzing the optimality and efficiency of the latent representation is even more relevant as it could significantly impact the design and efficiency of the framework. Thus, in this section, we explore the use of reduced-order representations developed in Section II to capture the features of the terrain along a history of local height maps.

The CNN-VAE architecture illustrated in Fig. 4 can be easily modified to integrate the history of the local height maps. While its fundamental structure remains unchanged, adjustments are made to accommodate n-inputs and n-outputs, corresponding to the last n history steps, enabling an n-to-n mapping. Since the local height map is a single-channel image, each of the n inputs is stacked in a channel to obtain an n-channel image, with the output following the same structure.

We experiment with two modified structures of the framework, n-to-n and n-to-1. In the n-to-n case, the latent representation captures the terrain features associated with the latest n terrain height maps and reconstructs the same n height maps from the latent variable. In the n-to-1 case, the latent variable is used to reconstruct only the single latest height map of the sequence in the n inputs. Similar to what was observed in Section II-C, there is a tradeoff in the dimension of the latent representation and the reconstruction accuracy, where increasing the size of the latent representation does not improve the reconstruction loss.

Section IV-G presents an ablation study that explores the impact of the dimension of the latent representation of the height map with and without the history of the terrain height map.

III. Hierarchical Terrain-Aware Bipedal Locomotion

Building on the success of reduced-order models for online generation of HL trajectories [19], [26], we employ RL to train an HL planner policy that harnesses an effective representation of the robot's dynamics and the terrain information. As shown in Fig. 2, the proposed HL RL policy takes as input a latent space encoding learned from the local height map of the terrain together with a reduced-order representation of the robot's states inspired by the linear inverted pendulum (LIP) model and the state of the swing foot of the robot. The output of the RL policy is a set of task-space commands used to generate online task-space trajectories for the robot's base and end-effectors. The low-level controller is a model-based whole-body controller used to guarantee the tracking performance of the desired task-space trajectories. The proposed hierarchical framework allows for replacing the latent variable encoded from the terrain with a latent variable encoded from depth images through an additional distillation stage. The details of the distillation process for the latent space reconstruction from depth images are presented in Section IV-C.

A. RL for HL Planning

The problem of determining a motion policy for bipedal robots can be modeled as a Markov decision process (MDP). The stochastic transition of the MDP process captures the random sampling of initial states in the policy training and dynamics uncertainty due to model mismatch and random interactions with the environment (e.g., early ground impacts).

B. Reduced-Order State Space

In this work, we leverage the insights provided by template models to regulate the walking speed of biped robots [26]. Inspired by the success of [2], [19] in using template-based models, we select the state

$$\mathbf{s} = \left(\mathbf{x}_{\mathrm{b}}, h_{\mathrm{b}}, \mathbf{e}_{\bar{v}}, v_x^d, v_y^d, \mathbf{p}_{\mathbf{sw}}, \mathbf{v}_{\mathbf{sw}}, \mathbf{z}, \mathbf{a}_{t-1}\right) \tag{5}$$

where $\mathbf{x}_{\mathrm{b}} = (x, y, \dot{x}, \dot{y})$ is the LIP state composed of the robot's base position relative to the stance foot and the base velocity, h_{b} is the robot's base height relative to the stance foot, $\mathbf{e}_{\bar{v}} = (e_{\bar{v}_x}, e_{\bar{v}_y})$ is the error between the average velocity of the robot's base $(\bar{v}_x, \bar{v}_y)$ and the commanded robot's velocity (v_x^d, v_y^d), $\mathbf{p}_{\mathbf{sw}}$ and $\mathbf{v}_{\mathbf{sw}}$ are the 3-D position and velocity of the robot's swing foot, $\mathbf{z}$ is the latent variable encoded from the local height map centered at the robot's base introduced in Section II, and $\mathbf{a}_{t-1}$ is the last policy action. The positions and velocities of the robot's base and swing foot are expressed in the frame coordinates of the robot's stance foot. All variables correspond to the data at time t unless explicitly denoted with the subscript t as in the last policy action $\mathbf{a}_{t-1}$.

C. Task-Space Actions

The action $\mathbf{a} \in \mathcal{A}$ is chosen to be

$$\mathbf{a} = (\bar{p}_X, \bar{p}_Y, \bar{p}_Z, h_{\mathrm{sw}}, \dot{x}_{\mathrm{off}}, \dot{y}_{\mathrm{off}}) \tag{6}$$

where $\bar{\mathbf{p}} = [\bar{p}_X, \bar{p}_Y, \bar{p}_Z]^T$ correspond to the landing position of the swing foot w.r.t. the robot's base at the end of the swing phase T, h_{sw} is swing foot clearance, and $\dot{x}_{\mathrm{off}}, \dot{y}_{\mathrm{off}}$ are an offset added to the commanded speed of the robot. This selection of the action space encourages the policy's flexibility to exploit the bipedal robot's natural nonlinear dynamics and enhance the policy's robustness under challenging terrains, disturbances, and sudden speed changes caused by irregularities in the terrain, i.e., tripping over. Fig. 8 illustrates the selection of

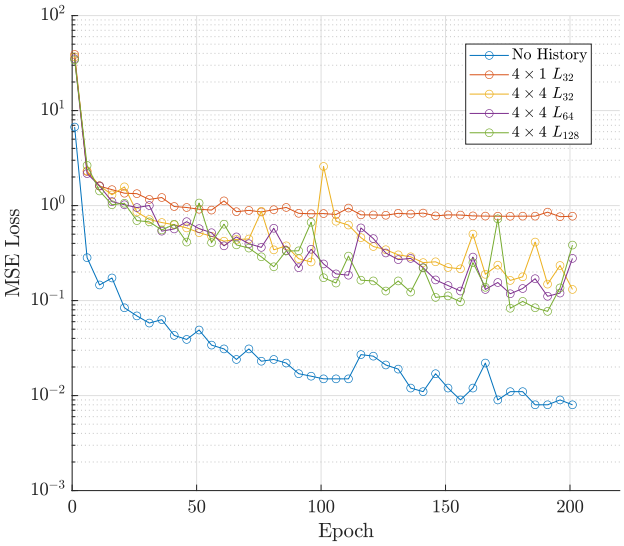

Fig. 7. Training loss curves for CNN-VAE models incorporating terrain history and different latent sizes. The same tradeoff emerges: increasing the latent dimension beyond a certain point yields diminishing improvements.

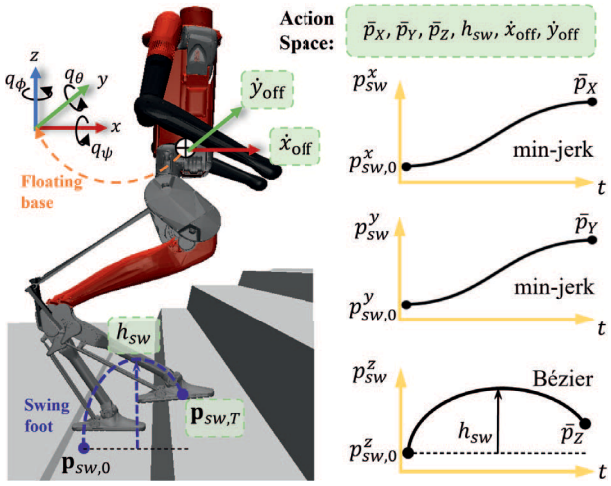

Fig. 8. Action space representation for the RL policy. The policy outputs target trajectories in task space for swing foot position and base velocity offset, which the low-level controller then executes.

the action space on the robot digit. The action space of the policy is updated at 33 Hz.

Given the desired set of policy actions, the trajectory generation module transforms the policy action a into smooth task-space trajectories for the robot's base and end-effectors. Specifically, as shown in Fig. 8, at a time $t \in [0, T]$ the trajectories for the swing foot $p^x_{\text{sw}}(t, \mathbf{a})$ and $p^y_{\text{sw}}(t, \mathbf{a})$ are generated using a minimum jerk trajectory connecting initial foot positions with target foot positions from the policy action. $p^z_{\text{sw}}(t, \mathbf{a})$ is generated using a Bézier polynomial with five control points, with its maximum value corresponding to the height of the swing foot h_{sw}. The initial foot positions are computed at every touchdown event and kept constant throughout the step.

D. Low-Level Task-Space Controller

The desired task-space trajectories derived from the policy outputs are tracked using task-space inverse dynamics (TSIDs) with a quadratic programming formulation. We follow the TSID formulation in [27], which considers the constrained dynamics of closed kinematic chains such as the ones in digit's legs. Here, we only present the problem formulation and refer the interested reader to [27] for more details.

Consider a bipedal robot with configuration space $\mathcal{Q} \subset \mathbb{R}^n$ and generalized coordinates $q \in \mathcal{Q}$. The equations of motion of the constrained dynamics are given by

$$M(q)\ddot{q} + H(q, \dot{q}) = B\tau + J_c^T(q) f_c + N(q)\lambda \tag{7}$$

$$N(q)\ddot{q} + \dot{N}(q, \dot{q})\dot{q} = 0 \tag{8}$$

$$J_c(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} = 0 \tag{9}$$

where $M(q)$ is the inertia matrix, $H(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q) + F$ is the vector sum of the Coriolis, centripetal, gravitational, and additional nonconservative forces, B is the actuation matrix, $\tau \in \mathbb{R}^m$ is the torque inputs at the actuated joints, $J_c(q)$ is the contact Jacobian, $f_c \in \mathbb{R}^{3n_c}$ collects all external contact forces with n_c being the number of contacts. Moreover, $N(q) = J_1(q) - J_2(q)$ is the constraint Jacobian matrix, and λ is the constrained force due to the closed kinematic chain.

Given the current state $(q, \dot{q})$ of the robot and its task-space references $(X_i^*, \mathcal{V}_i^*, \mathcal{A}_i^*, f_c^*)$, TSID for the system with constrained dynamics is formulated as a quadratic programming problem

$$\min_{\ddot{q}, f_c, \lambda} \sum_i \left\| J_i\ddot{q} + \dot{J}_i\dot{q} - \mathcal{A}_i \right\|_{Q_i} + \left\| f_c - f_c^* \right\|_{R_f} + \|\lambda\|_{R_\lambda} \tag{10}$$

$$\begin{aligned} \text{st.} \quad & J_c(q)\ddot{q} + \dot{J}_c(q)\dot{q} = 0 \text{ (contact constraints)} \\ & N(q)\ddot{q} + \dot{N}(q)\dot{q} = 0 \text{ (loop-closure constraints)} \\ & f_c \in \mathcal{F} \text{ (friction cone)} \\ & \tau(\ddot{q}, f_c, \lambda) \in \mathcal{T} \text{ (torque limits)} \end{aligned}$$

where the weighting matrices (Q_i, R_f, R_λ) are the positive definite, $\tau \in \mathbb{R}^m$ is the torque computed by the robot dynamics given $(\ddot{q}, f_c, \lambda)$, J_i is the geometric Jacobian of task-space references, and $\mathcal{A}_i$ represents the desired spatial acceleration with state feedback, and it is defined by

$$\mathcal{A}_i = \mathcal{A}_i^* + K_p \log\left(X_{m,i}^T X_i^*\right) + K_d\left(\mathcal{V}_i^* - \mathcal{V}_{m,i}\right)$$

where $X_{m,i}$ and $\mathcal{V}_{m,i}$ correspond to each task's measured pose and spatial velocity. K_p and K_d could be seen as the stiffness and damping of a system, and both are positive definite matrices.

The task-space references are determined by the desired pose X_i^*, spatial velocity $\mathcal{V}_i^*$, and spatial acceleration $\mathcal{A}_i^*$ of the left foot, right foot, and floating base of the robot. The position and linear velocity for the swing foot are obtained from the trajectories generated from the parameters in the action space of the RL policy as discussed in Section III-C, e.g., $(\mathbf{p}_{\text{sw}}(t, \mathbf{a}), \dot{\mathbf{p}}_{\text{sw}}(t, \mathbf{a}), \ddot{\mathbf{p}}_{\text{sw}}(t, \mathbf{a}))$. The robot's base velocity is also obtained from the policy actions $(\dot{x}_{\text{off}}, \dot{y}_{\text{off}})$. However, the base's position is not being tracked to reduce disturbances caused by position errors due to discontinuities in the terrain. The roll and pitch orientation and angular velocities of the task-space references are 0, while the heading angle determines the yaw orientation. The stance foot target is

equal to its current position. Finally, the force task reference f_c^* is computed using the centroidal dynamics following the approach in [28].

E. Rewards

The reward function adopted in this work is designed to exploit the privileged information from the terrain's height map to shape the motion of the robot's swing foot while keeping track of the desired walking speed and reducing the variation of the policy actions between each iteration. More specifically, we define the reward function

$$\mathbf{r} = \mathbf{w}^T \left[r_{v_x}, r_{v_y}, r_{sw_x}, r_{sw_y}, r_{sw_z}, r_{sw_h}, r_{sw_f}, r_a \right]^T \tag{11}$$

with

$$r_a = \exp\left(-\|\mathbf{a}_k - \mathbf{a}_{k-1}\|^2\right) \tag{12}$$

$$r_\square = \exp\left(-\left\|\square - \square^d\right\|^2\right) \tag{13}$$

where $\square$ represents the measured value and $\square^d$ is the desired value. For the velocity rewards r_{v_x} and r_{v_y}, the target value is the desired walking speed sampled at the beginning of each episode. For the swing-foot rewards $r_{sw_x}, r_{sw_y}, r_{sw_z}$ the target values are the 3-D foothold position, where the (x, y) target coordinates are heuristically estimated from the desired walking speed, e.g., $p_x = v_x^d/T$, $p_y = v_y^d/T + p_{y_{\text{off}}}$ with $p_{y_{\text{off}}} = \pm 0.1$ being an offset to avoid feet collision in the lateral plane, and the z target coordinate is the terrain height corresponding to the (x, y) coordinates. The reward r_{sw_h} encourages the policy to achieve sufficient swing-foot clearance, which is the maximum height the swing foot reaches during the swing phase. The reference value for swing-foot clearance is 5 cm above the terrain height at the swing foot's (x, y) coordinates. This reward also helps to avoid unnecessarily over-lifting the swing foot when finding an obstacle. Finally, the reward r_{sw_f} penalizes any contact force on the swing foot, which is used to encourage the policy to avoid early contact with the edges of the terrain during the swing phase. We denote that all the quantities used in the rewards are expressed in the stance-foot frame, which is a common approach in bipedal locomotion. The weights for the reward terms are chosen as $\mathbf{w}^T = [0.2, 0.1, 0.075, 0.075, 0.15, 0.2, 0.15, 0.1]$.

F. RL Training Setup

We use the proximal policy optimization algorithm [29] with input normalization, fixed covariance, and parallel experience collection to train the RL policy. The neural network selected for the RL policy is an MLP with two hidden layers, each with 128 units and tan$h$ activation function. We use a batch size of 64 for the PPO algorithm, a discount factor of 0.95, and 56 000 samples with six epochs of policy update per algorithm iteration. For each training episode, a terrain type is randomly selected from a set of different terrains: hills, slopes, random stairs, squared steps, and stairs up. These terrains are randomly generated from a diverse set of parameters, such as slope degree, number of stairs, stairs dimensions (width, height, and depth), and size of squares, among others. Moreover, the initial state of the robot is drawn from a normal distribution about an initial pose corresponding to the robot standing in the double support phase. The same terrain parameter and initial state randomization are used during training, evaluation, and testing of the policies.

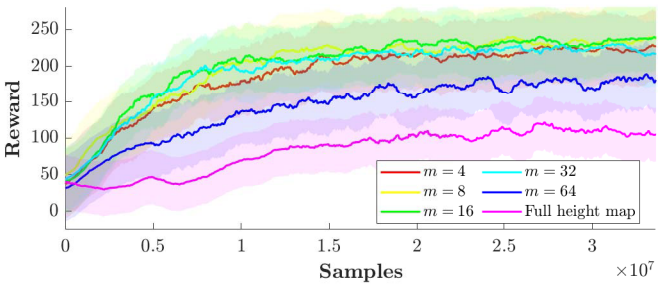

Fig. 9. Reward convergence for different values of the latent variable dimension. Policies using an appropriately-sized latent, e.g., $m = 16$, learn substantially faster and reach higher final rewards than those using an oversized latent, e.g., $m = 64$ or the raw 800-D height map.

One iteration step of the policy corresponds to the interaction of the learning agent with the environment. The RL policy takes the reduced-order state $\mathbf{s}$ and computes an action $\mathbf{a}$ converted into desired task-space trajectories at the time t_k. The reference trajectories are then sent to the task-space controller, which sends torque commands to the robot. This workflow is depicted in Fig. 2. The feedback control loop runs at 1 kHz, while the HL planner policy runs at 33 Hz. The maximum length of each episode is 300 iteration steps, corresponding to 9 s of simulated time. An episode will be terminated early if the torso pitch and roll angles exceed 1 rad or if the height of the robot's base relative to the stance foot is less than 0.4 m.

IV. Simulation Results

A. Learning Convergence

We demonstrate the effectiveness of the latent representation of the terrain by conducting an ablation study of the effect of this latent terrain representation on the efficiency and effectiveness of the learning process. In Fig. 9, we present the evolution of the reward during the RL training for different values of the latent dimension m. Notably, the reward curves with better learning efficiency (fewer epochs to converge) correspond to $m \leq 32$, while the reward curves corresponding to $m \geq 64$ show slower convergence to a lower value. In addition, we replace the latent representation of the terrain with the complete local height map matrix, which results in significantly decreased sample efficiency and policy performance, as shown in Fig. 9.

We do not claim that using the full height map to train RL policies for locomotion successfully is unfeasible, but that an accurate selection of the dimension of the latent representation significantly increases the sample efficiency of the learning process, as demonstrated in Fig. 9.

Comparing our proposed framework with other baselines is not straightforward for two main reasons. First, to the best of our knowledge, our work is the first terrain-aware learning-based locomotion implemented for the robot digit.

Although the work in [10], [12], and [13] shows similar approaches for learning-based perceptive locomotion, with

TABLE I
COMPARISON WITH OTHER RL-BASED APPROACHES FOR PERCEPTIVE LOCOMOTION

Method	# Samples	Architecture	Perception input	Policy output	Pre-trained
[10]	60×10^6	End-to-end	Full height map	Joint positions	No
[12]	60×10^6	End-to-end	Sampled terrain height	Joint positions	Yes
[13]	60×1.4^9	Teacher-Student	Sampled terrain height	Joint positions	Yes
Ours	30×10^6	Hierarchical	Latent representation	Task-space commands	No

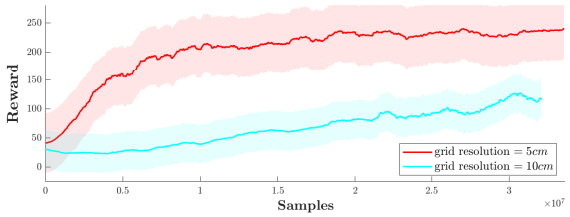

Fig. 10. Comparison of depth learning performance with different resolutions of the height map grid, demonstrating that higher values of the terrain grid result in a degraded performance of the policy.

the robot Cassie, these approaches are based on end-to-end and student–teacher RL frameworks that do not focus on sample efficiency. In these works, either the full local height map [12], samples of the terrain height [10], or a distilled representation from several depth images [13] are used along with the full-order state of the robot as the inputs of the RL policy, and the output is the desired motor position. While this approach is straightforward, it also significantly increases the complexity of the learning problem, requiring a higher number of parameters of the network, and a substantially higher number of samples to train a policy successfully. Table I shows this comparison, exhibiting the advantages of our proposed approach with at least $2\times$ increase in sample efficiency.

Second, the approach in [12] and [13] requires that the RL policy is already pretrained for blind locomotion, and their perceptive modules require a complex network architecture, including additional LSTM, CNN, ResNet, and U-NET networks. However, we acknowledge that these works have demonstrated successful sim-to-real transfer on the Cassie robot.

In contrast, we propose an efficient yet effective framework that mainly focuses on our policies' sample efficiency and lightweight nature. This allows training policies from scratch with fewer samples (without pretrained policies or precomputed reference trajectories), while providing insightful observations about the importance of the dimension of the latent variable used to represent the terrain features, which is not addressed in other perceptive locomotion work. We have not included other perceptive locomotion works, e.g., [16], [17], [18], in this comparison as they do not specify information about the number of samples required for the RL training to converge.

B. Grid Map Resolution

As mentioned in Section II-A, our intuition for the choice of grid size in the height map was to use a resolution as fine as the robot's foot width, which is about 5 cm. A grid of 5 cm is fine enough to capture the features of different types of terrain, while values higher than this, e.g., 10 cm, could be too coarse to capture important features like edges or borders of stairs and irregular terrains. This intuition also aligns with relevant work in the literature. For example, Hoeller et al. [22] used a variable resolution point cloud, where coarse resolution voxels (12.5 cm) are used to map the further scene of the robot. In comparison, high-resolution voxels (6.25 cm) capture the environment close to the robot. In [10], the terrain height is sampled using a pattern of 318 points adaptively spaced circularly around the foot's position, where the samples close to the robot have a resolution of about 5 cm. Finally, Duan et al. [12] also selected a resolution of 5 cm for the height map grid, which is used as an input to an RL policy already pretrained for blind locomotion for the Cassie robot. Fig. 10 shows that using a resolution of 5-cm results in significantly better rewards with fewer samples than 10-cm resolution.

C. Latent Space Reconstruction From Depth Images

The latent variable introduced in (1) enables the policy to capture an efficient and effective representation of the terrain. However, deploying this approach on a physical robot poses a significant challenge: obtaining an accurate local height map can be computationally expensive, highly noise sensitive, and often requires a costly sensor suite.

To address the challenges in sim-to-real transfer for bipedal locomotion, we implement a latent space distillation framework that directly constructs the latent representation from raw depth sensor inputs. This approach circumvents the conventional dependency on local height maps, streamlining perception while enhancing robustness to real-world sensor noise. Although several frameworks use the history of the depth images combined with the robot state in the distillation stage to reconstruct the entire terrain height map [12], [22], these approach results in complex network architectures and additional computational burden in training and inference of the perception module. In our work, we leverage synchronized feeds from two Intel D435i depth cameras mounted on the base of the robot's torso and pelvis to accurately recover the corresponding local height map from only one frame of the two combined depth images, ensuring sufficient coverage of the robot's surrounding terrain with minimal computational burden. A CNN-VAE is trained to align its latent space with that derived from previously trained local height map latent space. The training objective combines the VAE loss (4) for distributional regularization, an mse loss between latent vectors from height-map processing (teacher) and from raw depth images (student). This additional distillation process is shown at the bottom of Fig. 2.

Fig. 11 analyzes the accuracy of the reconstructions of the local height map using the latent variable obtained from: 1) the original local height map and 2) the depth camera images. The first column in Fig. 11 shows the ground truth of height map samples of different terrains (stairs, random stairs, and squared steps) obtained from simulation. The second and third columns show the reconstruction of the local height maps from the latent variable encoded directly from the local height matrix and the latent variable encoded from the depth images,

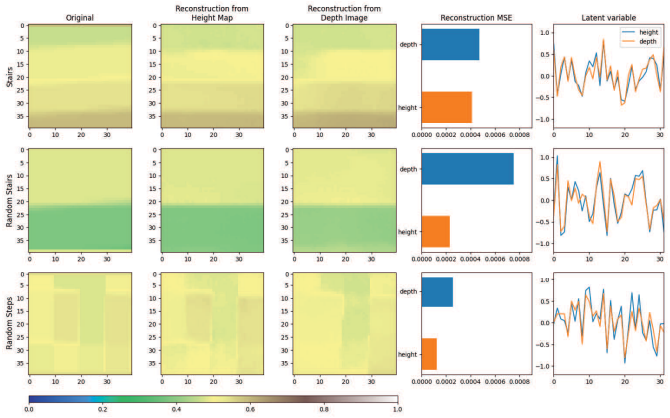

Fig. 11. Reconstruction of local height maps from latent encodings learned from two different exteroceptive sources. Each column shows: 1) the ground-truth height map for a sample terrain; 2) the reconstruction from the latent vector produced by the height map encoder; 3) the reconstruction from the latent vector produced by the depth-image encoder; 4) the error between the two reconstructions; and 5) the latent variable comparison. The reconstruction error is very low, demonstrating that the depth-image encoder successfully captures a similar latent representation to the height-map-based encoder.

respectively. The latent variable from the local height map and the depth images is depicted in the last column, where we show the effectiveness of the distillation process in learning the same latent representation from two different sources. Finally, while the visual comparison of the images in the second and third columns clearly shows a good reconstruction of the terrain's height map, we quantitatively capture the accuracy of the reconstruction from the latent variable by showing the MSE error between the two height-map matrices in the fourth column of Fig. 11.

Moreover, to reduce the sim-to-real gap caused by the difference between the depth images in simulation and hardware, we process the simulated depth images using postprocessing techniques inspired by [13] and [30] that have demonstrated successful sim-to-real transfer of perceptive-locomotion policies. In particular, we: 1) crop the image to remove the blind spots cause by stereo-matching at small distances; 2) add Gaussian blurring and characteristic depth shadowing around edges using Canny edge detectors; 3) clipped max depth to 2 m; 4) inpainting for hole filling with edge coherence; 5) edge pixel dropout simulating occlusion artifacts; and 6) masking of big occlusions caused by the legs in the downward camera. To demonstrate the effectiveness of these techniques, Fig. 12 shows a comparison between simulated and real camera feedback from the D435i cameras on the digit robot, highlighting the resemblance of the simulated environment.

D. Policy Performance

We denote that the RL policy learns to walk from scratch and that a single-trained policy can navigate successfully on various terrains, as shown in Fig. 1. When walking over irregular terrain, the policy adapts the foot landing location by taking shorter or longer steps to avoid collisions with the edges of the terrain. Similarly, the policy adapts the foot location to compensate for the heavy robot's inertia to prevent falling, i.e., when stepping up or down the stairs. These adaptive strategies naturally emerged during training from effectively integrating terrain features into the RL policy and the combination of rewards. We denote all results presented using a fixed step duration of 0.4 s. The policy does not change the step timing in response to terrain but adjusts foot placement. While adding a variable stepping frequency could be an interesting extension, the primary focus of this work is on the impact of the latent representation of the terrain. More details about the policy performance can be seen in the accompanying video: https://youtu.be/tJVfQK2XcQs

Table II shows the range of terrain parameters used during data collection, policy training, and evaluation. These parameters were consistent across all the experiments to ensure a fair comparison and to avoid domain gaps between training and testing. Although the policy navigates successfully on different terrains, there is a tradeoff between robustness and tracking the commanded walking speed as the complexity of the environment causes a higher tracking error in some of the terrains. We denote that the velocity tracking reward is a soft constraint within the RL formulation; therefore, perfect tracking is not expected, especially when it could come at a higher cost for other important rewards, e.g., avoiding the robot from falling, which would result in an early episode termination. In other words, the policy sacrifices velocity tracking performance to guarantee the robustness of the walking gait. This is particularly evident in Fig. 13, where we show the tracking error between the average walking speed $\hat{v}_x$ and the desired walking speed $v_x^d = 0.5$ m/s over 20 runs of the same policy for four different terrains. Despite irregularities in the terrain, the policy adapts its behavior to keep close track of the target speed, except in cases where the terrain conditions are too challenging, e.g., steep stairs, forcing the policy to deviate from the desired walking speed to avoid falling. Even with terrain awareness, the robot must exert greater corrective control on more difficult terrains, which leads to higher tracking error. The latent representation guides foot placement to appropriate locations. However, once a foot makes contact with an irregular surface, disturbances (like small slips or tilts) can still occur and must be corrected by the controller, resulting in deviations. Moreover, on more challenging terrains, the robot's dynamics are more perturbed–for example, when a foot lands on a high step, the robot's body might experience a jolt or require more corrective effort, leading to larger tracking errors in velocities.

E. Robustness and Comparison

To quantitatively assess the policy's robustness, we conduct a Monte Carlo evaluation. We test the policy's performance across diverse terrains for at least 200 experiments per terrain type, where each terrain instance is randomly generated by sampling its parameters from the distributions shown in Table II. The success rate for each terrain type is shown in Fig. 14 with a confidence interval $\geq 95\%$. A successful trial consists of the robot walking without falling for 9 s. This consistency of success across a spectrum of terrains highlights the capability of the policy to navigate effectively and adapt without terrain-specific tuning. These results also

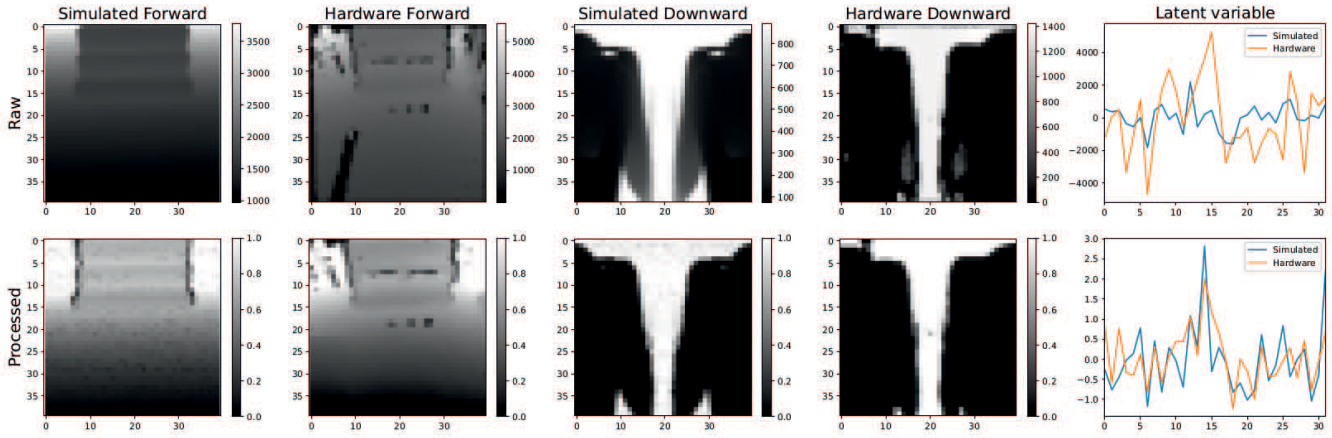

Fig. 12. Comparison of depth images from the hardware and the simulation before and after processing, e.g., clipping, cropping, and filtering. The top row presents the raw data obtained from each source, whereas the bottom row depicts simulated and hardware post-processed depth images. The processed simulated depth image closely resembles the real sensor image, and their resulting latent vectors are also very similar. This shows that with noise and occlusion handling, our simulation model produces depth data that the encoder perceives as real data, supporting the potential for sim-to-real transfer.

TABLE II
PARAMETERS SAMPLED FROM DISTRIBUTIONS FOR DIFFERENT TERRAINS

Terrain Type	Parameter	Distribution	Params (Mean, Std)	Limits (Min, Max)	Ext Limits (Min, Max)
Stairs	Step length [m]	Normal	(0.4, 0.1)	(0.3, 0.5)	(0.25, 0.55)
	Step height [m]	Normal	(0.15, 0.1)	(0.05, 0.25)	(0.025, 0.275)
Hills	Max amplitude	Uniform	N/A	(0.3, 0.4)	(0.26, 0.44)
	Octaves	Normal	(10.0, 2.0)	(5, 15)	(3.5, 16.5)
Slopes	Slope plane about axis x	Normal	(0.0, 0.1)	(-0.2, 0.2)	(-0.22, 0.22)
	Slope plane about axis y	Normal	(0.0, 0.1)	(-0.2, 0.2)	(-0.22, 0.22)
Square Steps	Max step size [m]	Uniform	N/A	(0.3, 0.5)	(0.25, 0.55)
	Max step height [m]	Uniform	N/A	(0.15, 0.25)	(0.125, 0.275)

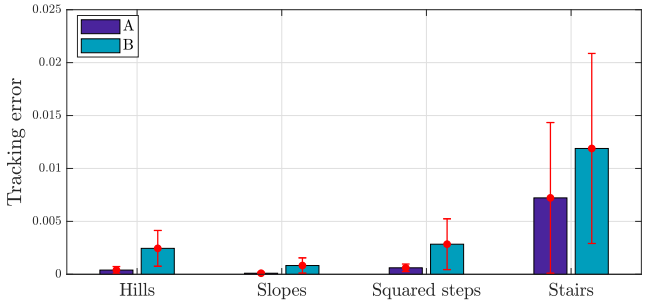

Fig. 13. Average mse for velocity tracking over 20 runs for height map policy (A) and depth-image policy (B).

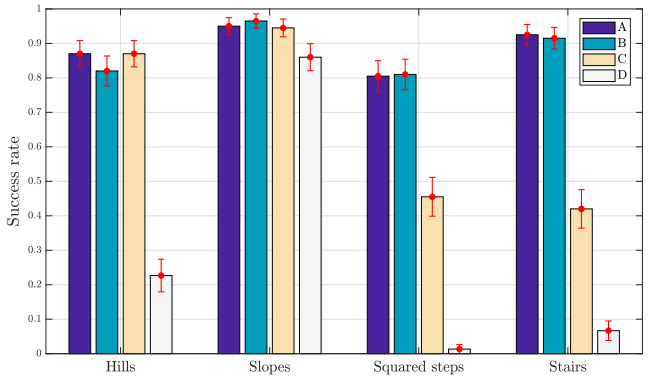

Fig. 14. Robustness of the policy to different terrains collected over more than 200 experiments per terrain with a confidence interval ≥ 95%.

demonstrate the policy's reliability in challenging terrains, an essential quality for humanoid robots to promote real-world deployment.

Furthermore, to demonstrate the actual contribution of the latent representation of the height map, we compare our proposed approaches (policies A and B) with two baselines that share the same RL policy structure but use different inputs for the terrain representation. We denote these two baselines as policy C and policy D.

Policy C corresponds to the case of blind locomotion, where the policy does not have a meaningful representation of the upcoming terrain. By keeping the input of the local height map to a fixed value corresponding to flat terrain, the policy "thinks" that it is walking on flat ground. The policy is robust enough to handle terrains with hills and slopes, which is consistent with several works on blind bipedal locomotion, where it has been shown that the potential of RL policies to navigate on these types of terrains without using exteroceptive feedback. However, its success rate drops significantly for terrains with random square steps and stairs, resulting in the robot immediately falling after tripping over the edges of the steps in the terrain.

Policy D corresponds to the case where the full terrain height map $\mathbf{x} \in \mathbb{R}^{20\times 40}$ is included in the policy state. Although

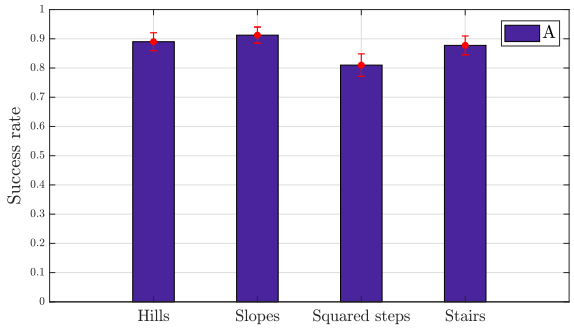

Fig. 15. Success rates with out-of-distribution terrains with an extended range (20%) of key terrain parameters concerning the original training range. The extended limits are presented in Table II. The policy maintains high success on most terrains (e.g., hills, and squared steps) despite the increased difficulty, highlighting the robustness of the learned perceptual latent space and the policy's adaptability.

this approach has been successfully applied in other end-to-end RL frameworks for bipedal locomotion [10], [12], it is incompatible with our proposed compact and sample-efficient framework. We hypothesize that the lack of structure in the raw terrain height map data results in a bottleneck for learning effective actions. This effect is observed from Fig. 9, where the reward curve for the policy with the full height map converges significantly slower to a smaller value than the policies that use the latent representation of the terrain height map. To alleviate this effect, Duan et al. [12] build upon a pretrained RL policy according to [31] and uses the complete height map along with the full-order robot's state to learn compensations added to the base RL policy. As shown in Fig. 14, policy D is the worst performer, even under-performing blind locomotion (policy C).

On the other hand, policy A, which corresponds to our approach of training the policy using the reduced-latent representation of the local height map, and policy B, which corresponds to the approach of generating the latent representation of the local height map using depth cameras, perform consistently across all terrain types with a success rate greater than 80%. This underlines the robustness of reduced-order latent representations and their capabilities to generalize well across different terrains without fine-tuning and to generalize to different perceptive sources, i.e., depth cameras, through a simple distillation process.

F. Generalization to Out-of-Distribution Parameters and Sim-to-Sim Transfer

To evaluate the generalization capability of the learned policy beyond the training distribution, we conducted additional tests using an extended range of terrain parameters not seen during training. Specifically, we increased the upper bounds of the terrain generation parameters by 20% across all terrain types, as detailed in the last column of Table II. This controlled extrapolation aims to assess the robustness of the policy under terrain conditions that exceed the complexity of the training set.

As shown in Fig. 15, the policy maintains high success rates on terrains such as *hills* and *squared steps*, despite the increased difficulty. We denote that the CNN-VAE module used for terrain encoding was not retrained on these extended terrains. The strong performance, therefore, highlights both: 1) the robustness of the learned perceptual latent space in encoding previously unseen terrains and 2) the adaptability of the policy to respond effectively to out-of-distribution scenarios. These results demonstrate that the policy has acquired transferable perceptual-motor representations capable of generalizing beyond the training distribution for various terrain types.

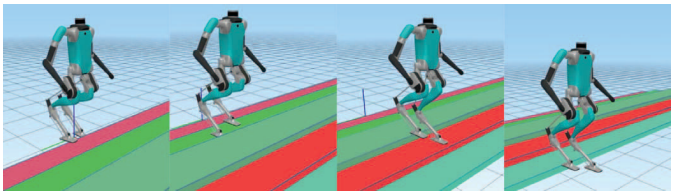

Fig. 16. Sequence of images of digit walking up and down a staircase in the AR's high-fidelity simulator. Using our terrain-aware policy, the robot successfully climbs and descends the stairs without falling.

In contrast, performance on the *slopes* and *stairs* terrains slightly degrades under the extended parameter settings. We attribute this to the structured and discontinuous nature of stair environments, which become significantly more challenging with increased step height and gap width, and the complex interaction between the flat landing foot and the steep slopes. These changes represent a challenge for the kinematic and dynamic limitations of the Digit robot, reducing the available foothold margin and increasing the likelihood of unstable contact or foot scuffing during swing and landing. While out-of-distribution generalization is not the primary focus of this work, we recognize the value of incorporating more complex mechanisms—such as the attention-based models applied in [16]—to better handle such terrain complexities or online adaptive strategies for foot orientation. We consider this an important direction for future work.

Finally, we successfully tested policy A in the ARs Simulator, a highly realistic environment for the Digit robot, including features such as real-time simulation, communication delays, actuator delays, and the exact state estimation used in the hardware. It also shares the same API as the hardware, meaning that the same code used in the simulation can be deployed on the hardware with a high probability of success. The effectiveness of the AR simulator as a good sim-to-real evaluation tool has been demonstrated in several works using Digit, where policies tested in the AR simulator have been successfully transferred to hardware [32], [33], [34].

In Fig. 16, we show a tile plot of the robot walking up and down stairs in the AR simulator. Since the AR simulation does not provide the depth camera feedback, the policy shown in Fig. 16 corresponds to policy A in Section IV-E. The latent representation is encoded from the local terrain height map. The policy successfully leverages the efficient latent representation of the terrain to command the task-space actions that allow the robot to lift its feet at the right time and place to successfully traverse the stairs without falling. A detailed sequence of the motion is also shown in the accompanying

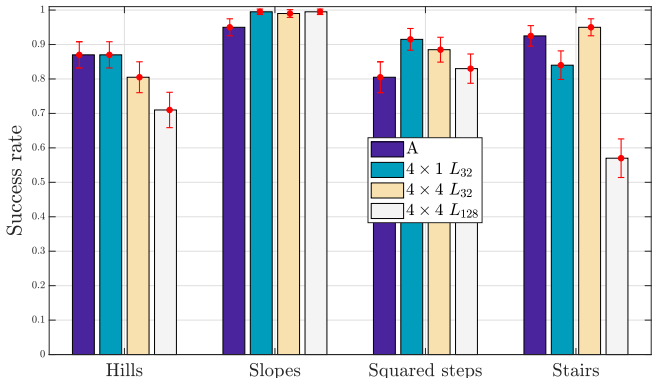

Fig. 17. Success rates on various terrains for policies trained with history-augmented terrain encoders, comparing different input–output structures and latent sizes. Including a short history of height maps can improve robustness on certain terrains (all history-based models exceed 95% success on slopes). However, using a larger latent space, e.g., L_{64} versus L_{32}, does not guarantee better performance. In fact, the $4\times1L_{32}$ model (four-frame history compressed into one latent of size 32) emerged as the best performer on most terrains.

video. In addition, to verify the importance of the latent representation of the terrain, we also test policy C in the AR simulator. As expected and consistent with the results in Section IV-E and Fig. 14, the blind policy falls when the foot hits the edge of the stairs. These results are also presented in the accompanying video.

G. Comparison With History-Aware Perception

Finally, we show the performance of the policies trained using the latent representation based on the reconstruction results of the new CNN-VAEs training loss curves in Section II-D and Fig. 7, which indicates successful convergence for most history-aware CNN-VAEs.

Success rates for history-aware policies across terrains (see Fig. 17) show variations in effectiveness depending on the terrain type. While all models excelled on slopes (success rates >95%), performance differences were more pronounced on complex features like hills and stairs.

Despite the superior reconstruction accuracy, larger latent spaces (L_{64}, L_{128}) did not always result in better locomotion policies. Despite its compressed representation, the 4×4 L_{32} model consistently performed well, indicating that dimensionality reduction preserves essential terrain features while filtering noise. Interestingly, the 4×1 L_{32} model, which averages temporal inputs, emerged as the best performer across terrains, except for stairs, where its success rate dropped to 57%. This counterintuitive result suggests that for most terrains, a temporally compressed representation capturing the average terrain ahead provides robustness for locomotion planning.

We hypothesize that the averaging effect in the $4 \times 1L_{32}$ architecture acts as an implicit regularization mechanism, enhancing policy robustness by emphasizing persistent terrain features over transient details that can act as a predictor of the incoming terrain. This benefits navigation on slopes, hills, and squared steps, where gradual transitions outweigh precise height details. For example, when walking on a slope with a fixed inclination, the history of the height maps could help the policy infer the slope of the terrain, so it could adjust its stepping with the assumption that the incoming terrain in one or two steps could share the same terrain features as the past terrain. However, for stair traversal, where step height and edge detection are critical, temporal compressing blurs essential features, significantly hindering the performance.

V. Conclusion

We propose a framework for learning terrain-aware perceptive locomotion that integrates a latent representation of the local height map with a reduced-order representation of the robot's states to form an efficient state representation. By combining a learning-based HL terrain-aware planner that formulates effective task-space actions with a low-level feedback tracking controller, we obtain a robust controller capable of traversing challenging terrains while preserving excellent speed-tracking performance.

A central contribution of this work is the detailed analysis of the latent space dimension, with ablation studies providing empirical demonstrations that a larger dimension is not necessarily better at capturing meaningful terrain features. Our investigation into this principle of minimal sufficiency for perceptual information revealed that an optimally compressed latent representation is critical for sample efficiency and policy robustness. This principle was further validated through an analysis of history-aware perception.

We have established a clear and promising path toward real-world application by successfully distilling this compact representation from depth camera images with realistic sensor noise and validating our policy in the high-fidelity ARs simulator. Future work will focus on the direct hardware implementation of this framework on the Digit robot, building on the strong sim-to-real evidence presented. Further investigation will also explore adaptive mechanisms to handle more extreme out-of-distribution terrains, pushing the boundaries of agile and perceptive locomotion.

References

[1] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. van de Panne, "Feedback control for Cassie with deep reinforcement learning," in *Proc. EEE Int. Conf. Intell. Robots Syst.*, Oct. 2018, pp. 1241–1246.

[2] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Reinforcement learning-based cascade motion policy design for robust 3D bipedal locomotion," *IEEE Access*, vol. 10, pp. 20135–20148, 2022.

[3] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," in *Proc. Robotics, Sci. Syst.*, Jul. 2021. [Online]. Available: https://roboticsproceedings.org/rss17/p061.html

[4] X. B. Peng, G. Berseth, and M. van de Panne, "Dynamic terrain traversal skills using reinforcement learning," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–11, Jul. 2015.

[5] X. B. Peng, G. Berseth, and M. van de Panne, "Terrain-adaptive locomotion skills using deep reinforcement learning," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–12, Jul. 2016.

[6] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne, "DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, Aug. 2017.

[7] Z. Xie, H. Y. Ling, N. H. Kim, and M. van de Panne, "ALLSTEPS: Curriculum-driven learning of stepping stone skills," *Comput. Graph. Forum*, vol. 39, no. 8, pp. 213–224, Dec. 2020.

[8] R. P. Singh, M. Benallegue, M. Morisawa, R. Cisneros, and F. Kanehiro, "Learning bipedal walking on planned footsteps for humanoid robots," in *Proc. IEEE-RAS 21st Int. Conf. Hum. Robots (Humanoids)*, Nov. 2022, pp. 686–693.

[9] F. Acero, K. Yuan, and Z. Li, "Learning perceptual locomotion on uneven terrains using sparse visual observations," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 8611–8618, Oct. 2022.
[10] B. van Marum, M. Sabatelli, and H. Kasaei, "Learning perceptive bipedal locomotion over irregular terrain," 2023, *arXiv:2304.07236*.
[11] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Sci. Robot.*, vol. 7, no. 62, Jan. 2022. [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.abk2822#tab-citations
[12] H. Duan et al., "Learning vision-based bipedal locomotion for challenging terrain," 2023, *arXiv:2309.14594*.
[13] M. S. Gadde, P. Dugar, A. Malik, and A. Fern, "No more blind spots: Learning vision-based omnidirectional bipedal locomotion for challenging terrain," 2025, *arXiv:2508.11929*.
[14] X. Gu et al., "Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning," in *Proc. Robot., Sci. Syst.*, 2024. [Online]. Available: https://roboticsproceedings.org/rss20/p058.html
[15] H. Wang, H. Luo, W. Zhang, and H. Chen, "CTS: Concurrent teacher–student reinforcement learning for legged locomotion," 2024, *arXiv:2405.10830*.
[16] J. He, C. Zhang, F. Jenelten, R. Grandia, M. Bächer, and M. Hutter, "Attention-based map encoding for learning generalized legged locomotion," *Sci. Robot.*, vol. 10, no. 105, p. 3604, Aug. 2025. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.adv3604
[17] Q. Zhang et al., "Distillation-PPO: A novel two-stage reinforcement learning framework for humanoid robot perceptive locomotion," 2025, *arXiv:2503.08299*.
[18] J. Long et al., "Learning humanoid locomotion with perceptive internal model," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2025, pp. 9997–10003.
[19] G. A. Castillo, B. Weng, S. Yang, W. Zhang, and A. Hereid, "Template model inspired task space learning for robust bipedal locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 8582–8589.
[20] B. Weng, G. A. Castillo, Y.-S. Kang, and A. Hereid, "Towards standardized disturbance rejection testing of legged robot locomotion with linear impactor: A preliminary study, observations, and implications," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 9946–9952.
[21] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Data-driven latent space representation for robust bipedal locomotion learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 1172–1178.
[22] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "ANYmal parkour: Learning agile navigation for quadrupedal robots," 2023, *arXiv:2306.14874*.
[23] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 5026–5033.
[24] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent., Conf. Track*, Jan. 2014, pp. 1–14.
[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent., Conf. Track*, Jan. 2015, pp. 1–15.
[26] Y. Gong and J. W. Grizzle, "Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion," *J. Dyn. Syst., Meas., Control*, vol. 144, no. 12, Dec. 2022. [Online]. Available: https://asmedigitalcollection.asme.org/dynamicsystems/article/144/12/121006/1146629/Zero-Dynamics-Pendulum-Models-and-Angular-Momentum
[27] S. Yang et al., "Improved task space locomotion controller for a quadruped robot with parallel mechanisms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 13578–13585.
[28] P. M. Wensing and D. E. Orin, "Improved computation of the humanoid centroidal dynamics and application for whole-body control," *Int. J. Humanoid Robot.*, vol. 13, no. 1, Mar. 2016, Art. no. 1550039.
[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[30] N. Rudin, J. He, J. Aurand, and M. Hutter, "Parkour in the wild: Learning a general and extensible agile locomotion policy using multi-expert distillation and RL fine-tuning," 2025, *arXiv:2505.11164*.
[31] 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. Autom. (ICRA)*, May 2021, pp. 7309–7315.
[32] G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Robust feedback motion policy design using reinforcement learning on a 3D digit bipedal robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 5136–5143.
[33] V. C. Paredes and A. Hereid, "Safe whole-body task space control for humanoid robots," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2024, pp. 949–956.
[34] A. Shamsah, Z. Gu, J. Warnke, S. Hutchinson, and Y. Zhao, "Integrated task and motion planning for safe legged navigation in partially observable environments," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4913–4934, Dec. 2023.

Guillermo A. Castillo (Graduate Student Member, IEEE) received the B.E. degree in automation and control from the Escuela Politécnica Nacional (EPN), Quito, Ecuador, in 2015, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from The Ohio State University, Columbus, OH, USA, in 2019 and 2024, respectively.

He is currently a Robotics Software Engineer with Apptronik, Austin, TX, USA, where he works at the Motion, Control, and Planning Team, applying reinforcement learning methods for humanoid robots. His research interests include legged locomotion, reinforcement learning, and the integration of learning-based and model-based control for robust dynamic behaviors.

Dr. Castillo was a recipient of the Fulbright Scholarship in 2017, the Best Paper Award at the 2023 ICRA Workshop on Effective Representations, Abstractions, and Priors for Robot Learning, and the Presidential Fellowship at The Ohio State University in 2023.

Himanshu Lodha (Graduate Student Member, IEEE) received the B.Tech. degree in electronics and telecommunication engineering from the University of Pune, Pune, India, and the M.S. degree in electrical and computer engineering from The Ohio State University, Columbus, OH, USA, in 2017 and 2025, respectively, where he is currently pursuing the Ph.D. degree with the Department of Mechanical and Aerospace Engineering.

Before joining The Ohio State University, he was a Research Engineer at Indian Institute of Science, where he worked on the development of quadrupedal robots. His research interests include legged locomotion, learning-based control, computer vision, task generalization, and robotics to understand biological systems.

Ayonga Hereid (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Zhejiang University, Hangzhou, China, in 2007 and 2010, respectively, and the Ph.D. degree in mechanical engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2016.

He is currently an Assistant Professor with the Department of Mechanical and Aerospace Engineering, The Ohio State University, Columbus, OH, USA. Prior to joining The Ohio State University, he was a Post-Doctoral Research Fellow with the University of Michigan, Ann Arbor, MI, USA.

Dr. Hereid was a recipient of the NSF CAREER Award in 2022. His work was recognized for the Best Student Paper Award in 2014 from the ACM HSCC and was nominated as the Best Conference Paper Award Finalist at IEEE ICRA in 2016.