Now You See That: Learning End-to-End Humanoid Locomotion from Raw Pixels

Wandong Sun[1,2] Yongbo Su[1,2] HONOR Robot Team[2] Zongwu Xie[1]
[1]Harbin Institute of Technology [2]HONOR

https://github.com/Hellod035/Now_You_See_That

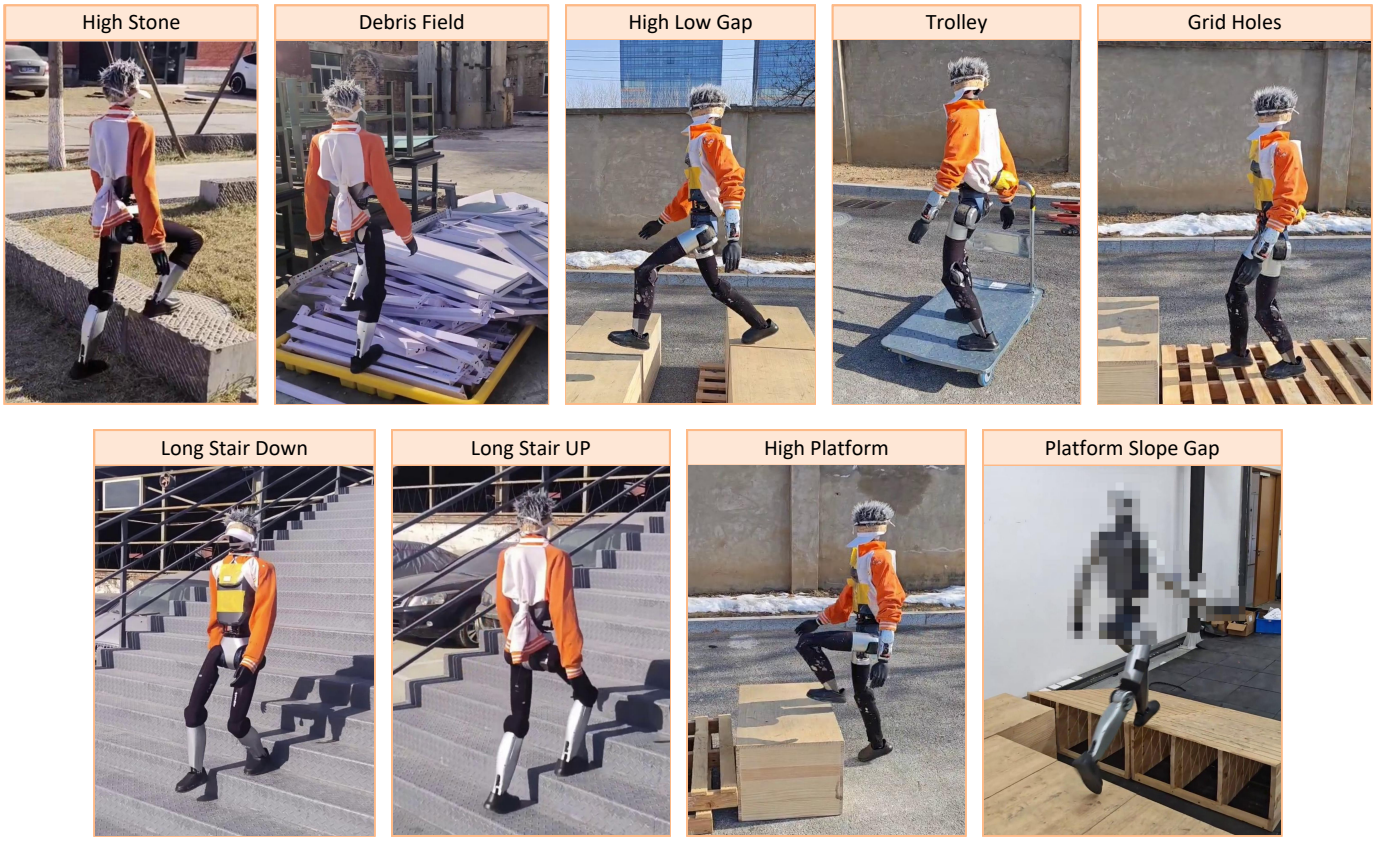

Fig. 1: Overview. Our end-to-end vision-based humanoid locomotion policy enables robust traversal across diverse challenging terrains, including high stones, long staircases (both ascending and descending), debris fields, gaps with varying heights, trolleys, high platforms, grid holes, and platform-slope-gap combinations. All behaviors emerge from a single unified policy trained with raw depth images.

***Abstract*—Achieving robust vision-based humanoid locomotion remains challenging due to two fundamental issues: the sim-to-real gap introduces significant perception noise that degrades performance on fine-grained tasks, and training a unified policy across diverse terrains is hindered by conflicting learning objectives. To address these challenges, we present an end-to-end framework for vision-driven humanoid locomotion. For robust sim-to-real transfer, we develop a high-fidelity depth sensor simulation that captures stereo matching artifacts and calibration uncertainties inherent in real-world sensing. We further propose a vision-aware behavior distillation approach that combines latent space alignment with noise-invariant auxiliary tasks, enabling effective knowledge transfer from privileged height maps to noisy depth observations. For versatile terrain adaptation, we introduce terrain-specific reward shaping integrated with multi-critic and multi-discriminator learning, where dedicated networks capture the distinct dynamics and motion priors of each terrain type. We validate our approach on two humanoid platforms equipped with different stereo depth cameras. The resulting policy demonstrates robust performance across diverse environments, seamlessly handling extreme challenges such as high platforms and wide gaps, as well as fine-grained tasks including bidirectional long-term staircase traversal.**

I. Introduction

Vision-based humanoid locomotion provides a compelling benchmark for embodied intelligence, requiring robots to traverse diverse terrains, from extreme obstacles like high platforms and wide gaps to fine-grained challenges such as continuous staircases, using onboard visual and proprioceptive sensing. Unlike quadrupedal locomotion where static stability tolerates moderate control errors [19], humanoid robots operate in inherently unstable regimes [8, 30] that demand precise coordination between perception and action. Vision-based

Method	Representation	End-to-End	Noise Modeling	Long-Term Deploy	Fine Locomotion	Extreme Parkour
Long et al. [26]	Elevation Map	✗	Moderate	✗	✓	✗
Sun et al. [47]	Elevation Map	✗	Moderate	✗	✓	✗
He et al. [12]	Elevation Map	✗	Moderate	✗	✓	✗
Ben et al. [2]	Voxel	✗	Moderate	✓	✓	✗
Zhuang et al. [60]	End-to-End Vision	✓	Moderate	✓	✗	✓
Song et al. [45]	Vision-to-Elevation	✗	Moderate	✓	✓	✗
Ours	End-to-End Vision	✓	Comprehensive	✓	✓	✓

TABLE I: Comparison of perceptive humanoid locomotion methods. Representation indicates the terrain perception approach. Noise Modeling indicates the comprehensiveness of depth sensor simulation. Long-Term Deploy indicates drift-free operation capability. Fine Locomotion indicates support for precise movements like stair climbing. Extreme Parkour indicates support for dynamic maneuvers across challenging obstacles.

humanoid locomotion faces two fundamental challenges: **(1)** perception noise from the sim-to-real gap severely degrades performance on fine-grained tasks requiring centimeter-level accuracy, and **(2)** training a single policy to handle diverse terrain scenarios remains difficult due to conflicting learning objectives across heterogeneous environments.

Current perception approaches for legged robots can be broadly categorized into LiDAR-based and vision-based methods. Elevation map-based methods fuse LiDAR height measurements with odometry to construct terrain representations [6, 7, 32, 20], achieving success on both quadrupedal [31, 15, 23] and humanoid platforms [27, 47, 24]. Voxel-based 3D occupancy grids [2, 34, 16] explicitly represent volumetric geometry. However, both LiDAR-based approaches suffer from limited sensing frequency and inherent latency, constraining their applicability for highly dynamic maneuvers. Vision-based methods using depth cameras [1, 5, 58, 51, 28] offer higher bandwidth ego-centric perception, yet their application to humanoid platforms remains limited due to sim-to-real gaps from imperfect depth sensor modeling.

Beyond perception, learning unified control across heterogeneous terrains poses additional challenges. Standard reinforcement learning with a single value function struggles to capture these diverse reward landscapes, leading prior work to train separate specialist policies before distillation [15, 59], incurring substantial training overhead.

In this work, we present an end-to-end vision-driven locomotion framework addressing both challenges. For robust perception transfer, we introduce comprehensive depth augmentation operators that reproduce realistic stereo camera imperfections, combined with vision-aware behavior distillation that aligns latent spaces through noise-invariant auxiliary tasks. For unified terrain mastery, we employ terrain-specific reward shaping with multi-critic reinforcement learning [33] and multi-discriminator[11] adversarial motion priors, where dedicated networks capture distinct dynamics and terrain-appropriate behaviors. We validate our approach on two humanoid platforms with different depth cameras, demonstrating robust performance across both extreme challenges (high platforms, wide gaps) and fine-grained locomotion tasks (long term staircases in both ascending and descending directions). Our contributions are:

- **Realistic depth sensor simulation:** A comprehensive augmentation pipeline simulating stereo matching artifacts, depth-dependent noise, optical distortions, and calibration uncertainties for seamless sim-to-real transfer.
- **Vision-aware behavior distillation:** A distillation framework combining latent space shaping with noise-invariant auxiliary tasks, transferring locomotion knowledge from privileged observations to noisy depth inputs.
- **Multi-critic and Multi-discriminator terrain learning:** Terrain-specific reward shaping with dedicated value networks and discriminators capturing distinct dynamics and motion priors across diverse terrains within a single unified policy.
- **Cross-platform validation:** Extensive real-world experiments on two humanoid robots with different stereo depth cameras, demonstrating successful traversal across both extreme and fine-grained terrains in diverse indoor and outdoor environments.

II. Related Work

A. Perceptive Legged Locomotion

Early work on legged locomotion primarily focused on height-based terrain perception using LiDAR sensors combined with odometry systems [6, 7]. These methods construct elevation maps that enable quadrupedal robots to traverse challenging terrains by providing dense geometric information for foothold planning [23, 15, 32]. Recent humanoid locomotion systems have adopted similar approaches [40, 24, 27, 13]. However, the dynamic stability of bipedal locomotion significantly amplifies odometry drift issues, cumulative errors in map registration directly cause foothold misalignment and falls, particularly in high-impact scenarios such as running, jumping, or stair navigation.

To eliminate odometry dependency, depth camera-based approaches have emerged as an alternative. Direct depth-to-action learning has demonstrated robust terrain traversal in quadrupedal systems [1, 5, 58, 59, 51, 28], where static stability provides inherent tolerance to perception noise. However, extending these methods to humanoid platforms remains challenging [46, 47]. An alternative approach reconstructs elevation maps directly from egocentric depth observations without external odometry [45], avoiding cumulative drift but introducing additional complexity compared to end-to-end approaches.

B. Data Augmentation in Visual Reinforcement Learning

Data augmentation (DA) has become a fundamental technique in visual reinforcement learning for improving both

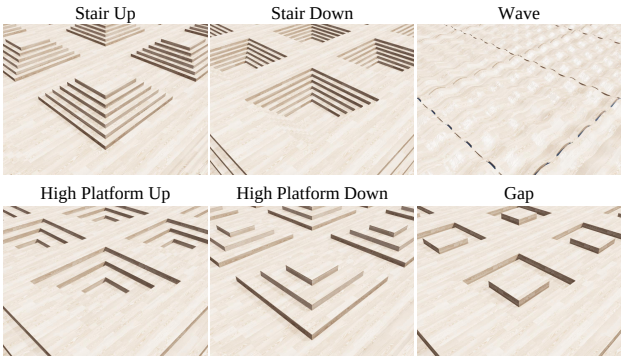

Fig. 2: Diverse terrain types used during training. Each terrain type contains 20 difficulty levels for curriculum learning.

sample efficiency and generalization ability [22, 53, 54, 52, 50]. Early work explored various image transformations, with random cropping emerging as particularly effective for sample-efficient training [53, 54]. Beyond simple transformations, spectrum-based augmentation [18] and saliency-guided approaches [3] have been proposed to improve generalization. However, existing domain randomization techniques primarily focus on simple perturbations [48, 38, 4], lacking systematic modeling of structured sensor artifacts present in real-world depth cameras. For depth-based perception, comprehensive augmentation strategies must simulate realistic sensor imperfections, including stereo fusion artifacts, depth-dependent noise patterns, optical distortions, and calibration variations [9], combined with appropriate preprocessing techniques to handle sensor-specific characteristics.

Recent advances have shown that auxiliary training objectives are essential for improving generalization beyond visual augmentation alone [41]. Auxiliary task approaches have proven essential [41]: contrastive learning methods like CURL [21] maximize mutual information between augmented views to learn invariant representations [29, 17], while consistency regularization techniques such as DrAC [42] explicitly penalize policy differences across augmentations to enforce noise-invariant features. Pre-trained visual representations have also shown promise for improving sample efficiency and generalization [43, 37, 35, 49, 56, 10]. Masked autoencoder-based approaches [14, 55] and predictive representation learning [36, 25] offer alternative strategies for learning robust features.

III. Method

We present an end-to-end framework for vision-based humanoid locomotion that addresses both perception transfer and unified terrain control. As illustrated in Figure 4, our approach follows a two-stage training pipeline. In the first stage, we train a privileged policy using height scan observations with multi-critic and multi-discriminator reinforcement learning, where terrain-specific reward shaping and dedicated value networks enable efficient learning across diverse scenarios. In the second stage, we distill the privileged policy into a deployment policy that operates directly on depth images, using vision-aware behavior distillation with comprehensive depth augmentation to ensure robust sim-to-real transfer.

A. Realistic Depth Sensor Simulation

Real depth cameras exhibit structured imperfections that standard simulation pipelines fail to capture. We introduce a comprehensive augmentation pipeline that sequentially applies eight operators to simulate realistic sensor artifacts, bridging the sim-to-real gap for robust policy transfer.

1) Stereo Depth Fusion: Stereo cameras reconstruct depth through binocular correspondence matching, producing characteristic hole patterns in occluded and textureless regions. We simulate this process by rendering images from left and right viewpoints with baseline distance b. For each pixel (u, v) in the left image, we compute disparity-based correspondence:

$$u_r = u - \frac{f_x b}{d_{\text{left}}(u, v) + \epsilon} \tag{1}$$

where f_x is the focal length and $\epsilon = 10^{-6}$ prevents division by zero. The fused depth applies a consistency check:

$$d_{\text{fused}}(u, v) = \begin{cases} d_{\text{left}}(u, v) & \text{if } |d_{\text{left}} - d_{\text{right}}(u_r, v)| < \tau \cdot d_{\text{left}} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\tau \in [0.05, 0.20]$ controls matching sensitivity. Pixels failing the consistency check are marked invalid, reproducing the hole patterns caused by occlusions and texture-poor surfaces.

2) Depth-Dependent Noise: Real depth measurements exhibit noise profiles that grow nonlinearly with distance due to the inverse relationship between disparity and depth. We model this effect with quadratic scaling:

$$\tilde{d}(u, v) = d(u, v) + \mathcal{N}(0, \sigma^2(d)), \quad \sigma(d) = |c_0 + c_1 d + c_2 d^2| \tag{3}$$

where $c_0, c_1, c_2 \sim \mathcal{U}(-0.03, 0.03)$ are randomly sampled per environment to capture inter-device variations. The absolute value ensures non-negative noise amplitude while preserving the randomization of noise characteristics across different simulated sensors.

3) Structured Noise Patterns: Environmental interference and sensor artifacts manifest as spatially correlated noise. We generate multi-octave Perlin noise to simulate these structured patterns:

$$n_{\text{perlin}}(u, v) = \sum_{o=0}^{4} 0.5^o \cdot \mathcal{P}(2^o u, 2^o v) \tag{4}$$

where $\mathcal{P}(\cdot, \cdot)$ denotes the standard Perlin noise function that generates smooth, continuous pseudo-random values through gradient interpolation. We use 5 octaves with persistence 0.5 to balance between large-scale intensity variations and fine-grained texture, matching the multi-scale nature of real sensor interference patterns. The noise is applied with depth-dependent amplitude: $\tilde{d} = d + \sigma_{\text{p}}(d) \cdot n_{\text{perlin}}$, where $\sigma_{\text{p}}(d) = |c_0^{\text{p}} + c_1^{\text{p}} d + c_2^{\text{p}} d^2|$ follows the same quadratic scaling with absolute value to ensure non-negative amplitude.

Order	Operation	Schedule	Parameters	Description
Preparation Stage				
1	Camera intrinsics	Once at startup	$s_{\text{h}}, s_{\text{v}} \sim \mathcal{U}(0.90, 1.10)$	Focal length variations
2	Camera extrinsics	Once at startup	$\Delta\mathbf{p} \sim \mathcal{U}(-0.05, 0.05)^3$ m	Mounting position offset
			$\Delta\boldsymbol{\theta} \sim \mathcal{U}(-0.10, 0.10)^3$ rad	Mounting orientation offset
3	Observation delay	Once at startup	$d_{\text{frame}} \sim \mathcal{U}[2, 4]$ frames	Processing pipeline latency
Processing Pipeline				
1	Stereo fusion	Per frame	$\tau \sim \mathcal{U}[0.05, 0.20]$	Disparity consistency check
2	Random convolution	Per frame	$w_{i,j,k,l} \sim \mathcal{U}(-0.05, 0.05)$	Optical aberrations
3	Gaussian noise	Per frame	$c_0, c_1, c_2 \sim \mathcal{U}(-0.03, 0.03)$	Distance-dependent noise
4	Perlin noise	Per frame	$c_0^{\text{p}}, c_1^{\text{p}}, c_2^{\text{p}} \sim \mathcal{U}(-0.02, 0.02)$	Time-dependent noise
5	Scale randomization	Per frame	$s_i \sim \mathcal{U}(0.90, 1.10)$	Calibration errors
6	Pixel failures	Per frame	$p_{\text{zero}} = p_{\text{max}} = 0.001$	Dead/saturated pixels
7	Depth clipping	Per frame	$[d_{\min}, d_{\max}] = [0.3, 2.0]$ m	Valid sensing range
8	Spatial cropping	Per frame	$(t, b, l, r) = (3, 3, 4, 4)$ pixels	Edge distortion removal

TABLE II: Vision Augmentation Pipeline: Operations, Schedule, Parameters and Descriptions

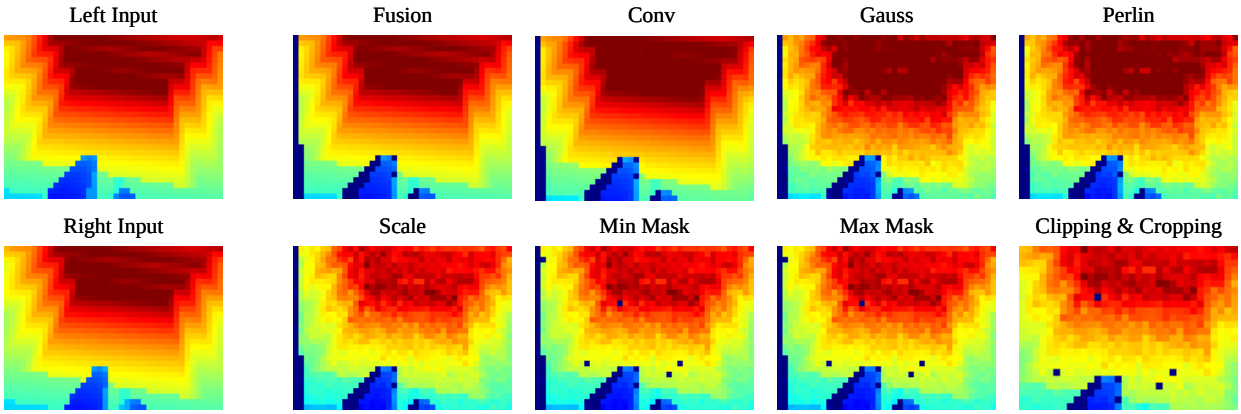

Fig. 3: Visualization of the depth augmentation pipeline. Starting from clean left and right depth images, the pipeline sequentially applies: (1) stereo fusion, (2) random convolution, (3) Gaussian noise, (4) Perlin noise, (5) scale randomization, (6) zero pixel failures, (7) max pixel failures, (8) depth clipping and spatial cropping to produce realistic depth observations for sim-to-real transfer.

4) Optical Distortions: Lens aberrations introduce local geometric distortions that vary across the image plane. We model these effects through randomized convolution:

$$\tilde{d} = d * (W + I) \tag{5}$$

where $W \in \mathbb{R}^{3\times3}$ with $w_{ij} \sim \mathcal{U}(-0.05, 0.05)$ and I is the identity kernel, ensuring the distortion remains centered around the original measurement.

5) Calibration Uncertainties: Manufacturing variations and imperfect calibration cause systematic measurement errors. We randomize depth scaling $s \sim \mathcal{U}(0.90, 1.10)$ to simulate factory calibration drift, perturb camera intrinsics through aperture scaling, and add extrinsic offsets to camera position and orientation. Additionally, we model pixel failures through random masking with probability $p_{\text{zero}} = p_{\text{max}} = 0.001$.

6) Preprocessing: We clip depth values to the valid sensing range $[0.3, 2.0]$ m, normalize to $[0, 1]$, and crop from 30×40 to 24×32 pixels to reduce peripheral distortion artifacts. Observation delay sampled from $\mathcal{U}[2, 4]$ frames simulates processing pipeline latency. Table II summarizes the complete augmentation pipeline with parameter ranges.

B. Privileged Reinforcement Learning

Parkour terrains exhibit fundamentally different dynamics, and a single value function struggles to capture these diverse reward landscapes. We address this challenge through terrain-specific reward shaping with dedicated critic and discriminator networks.

1) Height Scan Observations: The privileged policy receives dense geometric information via ray casting over a 1.6 m $\times$ 1.0 m ego-centric window with 0.05 m resolution, yielding a grid of $21 \times 33 = 693$ height samples. This terrain representation is subsequently transferred to depth-based perception through distillation. The architecture of the depth encoder is detailed in Appendix A.

2) Terrain-Specific Reward Shaping: We partition terrains into $K = 3$ categories with specialized reward components: (1) *stairs and platforms*, encompassing stair ascending, stair descending, platform ascending, and platform descending; (2) *gap crossing*; and (3) *rough terrain* locomotion. Detailed reward formulations are provided in Appendix D.

3) Multi-Critic and Multi-Discriminator Architecture: We maintain $K = 3$ critic networks $\{V_k\}_{k=1}^K$ and $K = 3$ discriminator networks $\{D_k\}_{k=1}^K$, each specialized for a terrain category: (1) stairs and platforms, (2) gap crossing, and (3)

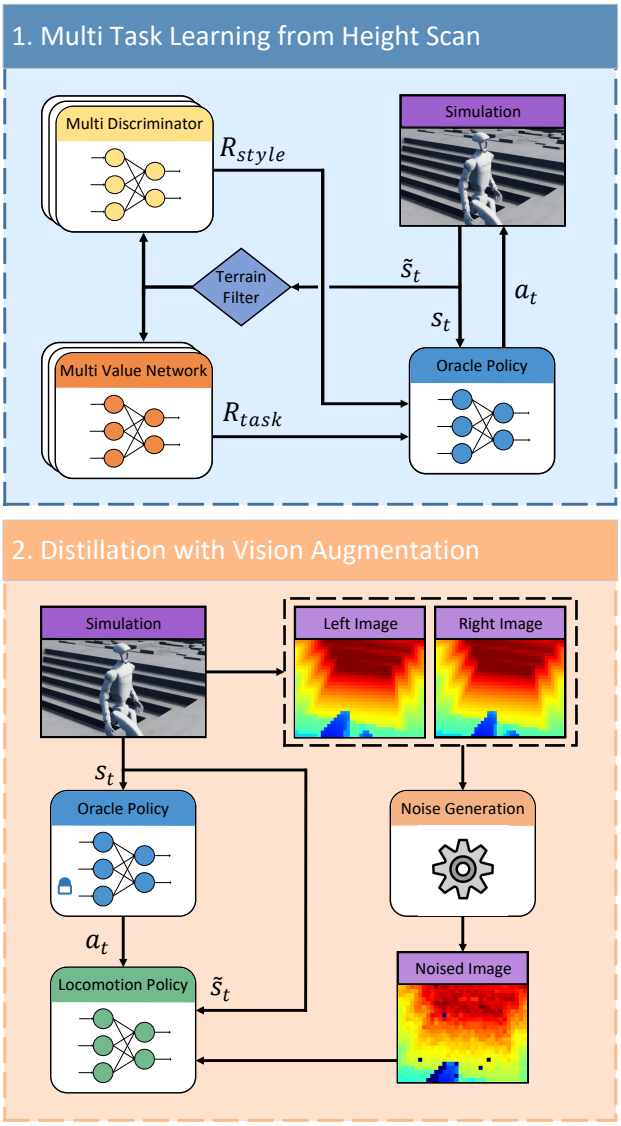

Fig. 4: Method Overview. Our framework consists of two stages: (1) *Privileged RL Training*: A teacher policy is trained with height scan observations using multi-critic and multi-discriminator learning, where terrain-specific reward shaping and dedicated value networks handle diverse terrain categories (stairs/platforms, gaps, rough terrain). (2) *Vision-Aware Distillation*: The privileged policy is distilled into a deployment policy operating on augmented depth images, combining behavior cloning with denoising objectives for robust sim-to-real transfer.

rough terrain. The critics share a common backbone while maintaining separate output heads, enabling efficient parameter sharing while preserving terrain-specific value estimation. During training, the appropriate critic and discriminator are selected based on terrain labels:

$$V(s_t) = V_{k(s_t)}(s_t), \quad D(s_t) = D_{k(s_t)}(s_t) \tag{6}$$

where $k(s_t) \in \{1, 2, 3\}$ identifies the current terrain category. Each critic learns the value landscape for its specific dynamics, while the actor receives policy gradients from the terrain-appropriate critic. Similarly, each discriminator provides style rewards tailored to the motion characteristics required for its terrain type.

Parameter	Range
Contact Properties	
Static/Dynamic friction	$(0.4, 1.2)$
Restitution	$(0.0, 0.4)$
Body Properties	
Mass scaling	$(0.8, 1.2)$
COM offset	± 0.03 m
Default joint offset	± 0.03 rad
Actuator Dynamics	
Armature scaling	$(0.5, 1.5)$
Stiffness/Damping scaling	$(0.9, 1.1)$
External Disturbances	
Push velocity	± 1.2 m/s
IMU bias	± 0.04 rad/s

TABLE III: Robot Dynamics Domain Randomization

4) Motion Priors: We incorporate Adversarial Motion Priors [39] to guide natural locomotion. The discriminator operates on torso-centric observations:

$$\Phi_{\text{amp}} = [\mathbf{q}_{\text{rel}}, \dot{\mathbf{q}}_{\text{rel}}, \mathbf{v}^b_{\text{torso}}, \omega^b_{\text{torso}}, \mathbf{g}^b, \mathbf{p}^b_{\text{body}}, \mathbf{q}^b_{\text{body}}] \tag{7}$$

We collect separate motion datasets for each terrain category to provide appropriate style guidance.

5) Advanced Domain Randomization: We apply comprehensive randomization to robot dynamics to ensure robust sim-to-real transfer. Table III details the complete randomization schedule covering contact properties, body parameters, actuator dynamics, and external disturbances.

C. Vision-Aware Behavior Distillation

We transfer the privileged policy to a deployment policy operating on depth images through DAgger-style distillation. The student policy interacts with the environment while the teacher provides supervision, enabling learning from its own state distribution.

1) Behavior Cloning: The deployment policy minimizes the action discrepancy with the privileged policy:

$$\mathcal{L}_{\text{behavior}} = \mathbb{E}_{s_t}\left[\|\mu_{\text{deploy}}(s_t) - \mu_{\text{priv}}(s_t)\|_2^2\right] \tag{8}$$

where the student receives augmented depth images with proprioceptive observations, while the teacher receives clean height scans.

2) Denoising Objective: To ensure robust feature extraction under sensor noise, we enforce consistency between clean and augmented depth representations:

$$\mathcal{L}_{\text{denoise}} = \mathbb{E}_{d,\tilde{d}}\left[\left\|E(d) - E(\tilde{d})\right\|_2^2\right] \tag{9}$$

where $E(\cdot)$ is the depth encoder, d is clean depth, and $\tilde{d}$ is its augmented counterpart.

3) Feature Regularization: We regularize the depth encoder features to avoid collapse by matching the *batch-wise feature distribution* to a standard normal prior. Let $z = E(d) \in \mathbb{R}^{N \times D}$

Method	Stairs Up			Stairs Down			Gaps			Platform			Average		
	SR	P↓	PDR↓	SR	P↓	PDR↓	SR	P↓	PDR↓	SR	P↓	PDR↓	SR	P↓	PDR↓
No Augmentation	$45.3_{\pm 1.8}$	$52.3_{\pm 1.2}$	$68.4_{\pm 2.1}$	$42.1_{\pm 2.0}$	$48.7_{\pm 1.1}$	$71.2_{\pm 2.3}$	$44.8_{\pm 1.7}$	$51.2_{\pm 1.0}$	$69.5_{\pm 1.9}$	$39.7_{\pm 2.2}$	$54.6_{\pm 1.3}$	$74.3_{\pm 2.5}$	$43.0_{\pm 1.9}$	$51.7_{\pm 1.2}$	$70.9_{\pm 2.2}$
Partial Aug.	$61.2_{\pm 1.5}$	$43.8_{\pm 0.9}$	$42.6_{\pm 1.4}$	$57.4_{\pm 1.6}$	$41.2_{\pm 0.8}$	$45.3_{\pm 1.5}$	$59.8_{\pm 1.4}$	$42.5_{\pm 0.7}$	$43.8_{\pm 1.3}$	$53.6_{\pm 1.8}$	$46.1_{\pm 1.0}$	$48.7_{\pm 1.6}$	$58.0_{\pm 1.6}$	$43.4_{\pm 0.9}$	$45.1_{\pm 1.5}$
Standard DR	$65.4_{\pm 1.3}$	$41.5_{\pm 0.8}$	$38.2_{\pm 1.2}$	$62.1_{\pm 1.4}$	$39.4_{\pm 0.7}$	$40.6_{\pm 1.3}$	$63.7_{\pm 1.2}$	$40.2_{\pm 0.6}$	$39.1_{\pm 1.1}$	$56.8_{\pm 1.6}$	$44.3_{\pm 0.9}$	$45.2_{\pm 1.4}$	$62.0_{\pm 1.4}$	$41.4_{\pm 0.8}$	$40.8_{\pm 1.3}$
Humanoid Parkour	$74.2_{\pm 1.1}$	$38.2_{\pm 0.7}$	$28.5_{\pm 1.0}$	$71.5_{\pm 1.2}$	$36.5_{\pm 0.6}$	$30.2_{\pm 1.1}$	$72.8_{\pm 1.0}$	$37.1_{\pm 0.5}$	$29.4_{\pm 0.9}$	$65.5_{\pm 1.4}$	$40.8_{\pm 0.8}$	$35.6_{\pm 1.2}$	$71.0_{\pm 1.2}$	$38.2_{\pm 0.7}$	$30.9_{\pm 1.1}$
Direct RL	$57.3_{\pm 1.9}$	$48.6_{\pm 1.1}$	$55.2_{\pm 1.8}$	$53.8_{\pm 2.1}$	$45.8_{\pm 1.0}$	$58.4_{\pm 2.0}$	$55.6_{\pm 1.8}$	$47.2_{\pm 0.9}$	$56.8_{\pm 1.7}$	$49.3_{\pm 2.3}$	$51.4_{\pm 1.2}$	$62.1_{\pm 2.2}$	$54.0_{\pm 2.0}$	$48.3_{\pm 1.1}$	$58.1_{\pm 1.9}$
Single Critic/Disc.	$85.6_{\pm 0.8}$	$34.6_{\pm 0.5}$	$18.3_{\pm 0.6}$	$82.3_{\pm 0.9}$	$32.8_{\pm 0.4}$	$20.1_{\pm 0.7}$	$83.8_{\pm 0.7}$	$33.5_{\pm 0.4}$	$19.2_{\pm 0.5}$	$76.3_{\pm 1.1}$	$37.8_{\pm 0.6}$	$25.4_{\pm 0.8}$	$82.0_{\pm 0.9}$	$34.7_{\pm 0.5}$	$20.8_{\pm 0.7}$
BC Only	$\underline{89.2}_{\pm 0.7}$	$\underline{32.4}_{\pm 0.4}$	$\underline{15.6}_{\pm 0.5}$	$\underline{86.5}_{\pm 0.8}$	$\underline{30.6}_{\pm 0.4}$	$\underline{17.2}_{\pm 0.6}$	$\underline{87.8}_{\pm 0.6}$	$\underline{31.2}_{\pm 0.3}$	$\underline{16.3}_{\pm 0.4}$	$\underline{80.5}_{\pm 0.9}$	$\underline{35.2}_{\pm 0.5}$	$\underline{21.8}_{\pm 0.7}$	$\underline{86.0}_{\pm 0.8}$	$\underline{32.4}_{\pm 0.4}$	$\underline{17.7}_{\pm 0.6}$
Ours	$\mathbf{99.2}_{\pm 0.3}$	$\mathbf{28.5}_{\pm 0.3}$	$\mathbf{5.2}_{\pm 0.2}$	$\mathbf{98.6}_{\pm 0.4}$	$\mathbf{27.2}_{\pm 0.3}$	$\mathbf{6.1}_{\pm 0.3}$	$\mathbf{99.3}_{\pm 0.2}$	$\mathbf{25.8}_{\pm 0.2}$	$\mathbf{4.5}_{\pm 0.2}$	$\mathbf{98.4}_{\pm 0.5}$	$\mathbf{29.4}_{\pm 0.4}$	$\mathbf{7.2}_{\pm 0.3}$	$\mathbf{98.9}_{\pm 0.4}$	$\mathbf{27.7}_{\pm 0.3}$	$\mathbf{5.8}_{\pm 0.3}$

TABLE IV: Performance on RDT-Bench across four terrain configurations. SR: Success Rate (%), P: Average Power ($\times 10^1$ W), PDR: Power Degradation Ratio (%). All methods are evaluated under CycleGAN-augmented realistic depth noise. Results report mean$_{\pm \text{std}}$ over 5 random seeds. Best results are in **bold**, second best are underlined. ↓ indicates lower is better.

denote the encoder outputs. We estimate a diagonal Gaussian $q(z) = \mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$ using empirical statistics:

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} z_{i,j}, \quad \sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(z_{i,j} - \mu_j)^2 + \epsilon \tag{10}$$

where ϵ is a small constant for numerical stability. We then minimize the KL divergence between this estimated distribution and the standard normal prior:

$$\mathcal{L}_{\text{kl}} = \mathrm{KL}\big(\mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2)) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\big) \tag{11}$$

4) Combined Objective: The total distillation loss balances all components:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{behavior}} + \lambda_{\text{denoise}}\mathcal{L}_{\text{denoise}} + \lambda_{\text{kl}}\mathcal{L}_{\text{kl}} \tag{12}$$

where $\lambda_{\text{denoise}} = \lambda_{\text{kl}} = 0.1$. The depth encoder output replaces the height scan embedding, enabling vision-based deployment.

IV. Experiments

We evaluate our approach through comprehensive simulation benchmarks and real-world deployment on a custom humanoid robot equipped with an Orbbec Gemini 336L depth camera. Our experiments address the following questions: (1) Does realistic depth augmentation improve sim-to-real transfer compared to standard domain randomization? (2) How does multi-critic learning contribute to performance across diverse terrains? (3) Do the distillation objectives improve policy robustness?

A. Experimental Setup

1) Real-World Depth Transfer Benchmark: Standard simulation benchmarks fail to capture the perception challenges of real-world deployment. We construct the Real-World Depth Transfer Benchmark (RDT-Bench) by training a CycleGAN [57] model on unpaired depth observations from simulation and real-world deployment. During evaluation, policies receive CycleGAN-augmented depth images while operating in the physics simulator, enabling large-scale quantitative comparison under realistic perception conditions. Importantly, CycleGAN is used *exclusively for evaluation*, each method employs its own augmentation strategy during training, ensuring fair comparison that reveals true sim-to-real transfer capability.

2) Evaluation Terrains: We evaluate on four representative terrain configurations spanning three categories:

- **Stairs:** 15 cm step height and 30 cm step depth, both ascending and descending
- **Gap:** 0.45 m width requiring accurate depth perception for crossing
- **Platform:** 0.40 m height requiring precise stepping up and down

Each configuration is tested across 1024 parallel environments for 100 episodes, providing statistically robust performance estimates.

3) Baselines: We compare against the following methods:

- **Humanoid Parkour Learning** [60]: State-of-the-art vision-based humanoid locomotion using standard domain randomization with Gaussian noise and random dropout.
- **No Augmentation**: Depth images with only simple Gaussian noise ($\sigma = 0.01$) and random pixel dropout ($p = 0.001$).
- **Partial Augmentation**: Depth-dependent noise and preprocessing only, without stereo fusion, structured noise, or optical distortion.
- **Standard DR**: Domain randomization following [60] with hole filling and random masking.
- **Single Critic/Disc.**: Our method with a single shared critic and discriminator for all terrains, instead of terrain-specific ones.
- **Direct RL**: End-to-end reinforcement learning directly from depth images without privileged distillation.
- **BC Only**: Behavior cloning with only $\mathcal{L}_{\text{behavior}}$, removing denoising and KL regularization losses.

4) Metrics: We evaluate methods using three complementary metrics:

- **Success Rate (SR):** Percentage of episodes where the robot traverses the terrain without falling.
- **Average Power (P):** Average mechanical power consumption computed as $P = \frac{1}{T}\sum_t \|\tau_t \odot \dot{q}_t\|_2$, where τ_t denotes joint torques and $\dot{q}_t$ denotes joint velocities. Lower values indicate more efficient locomotion.
- **Power Degradation Ratio (PDR):** Relative power increase under realistic noise, defined as $\text{PDR} = (P_{\text{RDT}} - P_{\text{clean}})/P_{\text{clean}} \times 100\%$. Lower PDR indicates greater robustness to sensor artifacts.

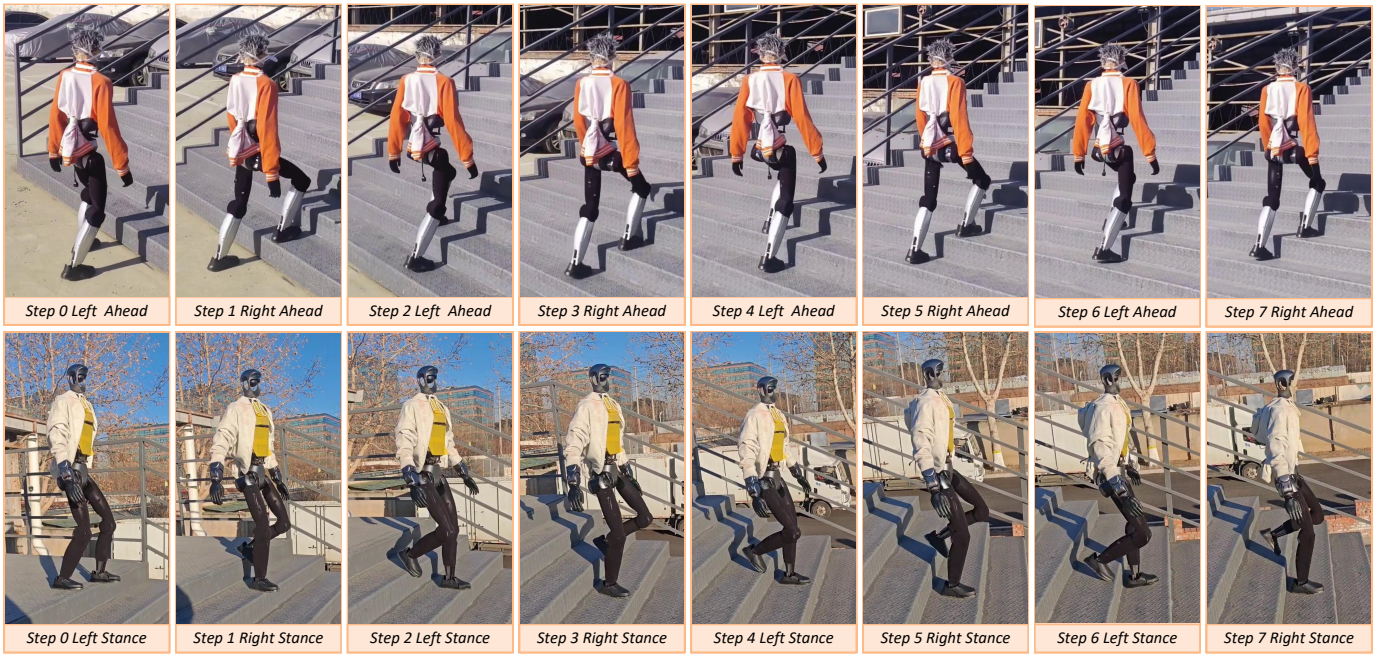

Fig. 5: Real-world deployment sequences demonstrating stair traversal. Top row: ascending stairs with anticipatory leg lifting. Bottom row: descending stairs with controlled foot placement. The policy executes smooth gait patterns without any real-world fine-tuning.

B. Main Results

Table IV presents comprehensive results on RDT-Bench. Our method achieves 98.9% average success rate with the lowest power consumption (27.7 $\times 10^1$ W) and minimal power degradation (5.8% PDR), substantially outperforming all baselines across every terrain and metric.

Perception Transfer. The comparison between augmentation strategies reveals the critical importance of realistic depth simulation. No Augmentation achieves only 43.0% success with severe power degradation (70.9% PDR), demonstrating that policies trained with simple Gaussian noise fail catastrophically under realistic sensor artifacts. Our full augmentation pipeline achieves 98.9% success with only 5.8% PDR, indicating consistent, efficient control despite perception noise.

Comparison with Prior Work. Humanoid Parkour Learning [60] achieves 71.0% success with 30.9% PDR. Our method outperforms this baseline by 27.9% in success rate while reducing PDR by 25.1%, demonstrating that our approach not only improves task completion but also maintains power-efficient locomotion under realistic perception conditions.

Architecture Contributions. Single Critic/Disc. achieves 82.0% success with 20.8% PDR, while our multi-critic and multi-discriminator approach achieves 98.9% success with 5.8% PDR. The terrain-specific value functions and discriminators enable more stable policy optimization by learning distinct motion priors for each terrain type. Direct RL without privileged distillation shows the highest PDR (58.1%), confirming that the two-stage training pipeline is essential for efficient vision-based control.

C. Ablation Studies

1) Augmentation Pipeline Analysis: Table V analyzes the contribution of each augmentation component. Stereo Fusion provides the largest improvement by simulating characteristic hole patterns that cause significant perception failures on real hardware. Depth-Dependent Noise and Calibration Uncertainties offer moderate gains by modeling distance-varying uncertainty and device variations. Even components with smaller SR impact contribute to control smoothness, validating the importance of comprehensive sensor modeling.

Configuration	SR (%)	P ($\times 10^1$ W)↓	PDR (%)↓
Full Pipeline	**98.9**$_{\pm 0.4}$	**27.7**$_{\pm 0.3}$	**5.8**$_{\pm 0.2}$
w/o Stereo Fusion	90.4$_{\pm 0.6}$	34.1$_{\pm 0.5}$	17.5$_{\pm 0.4}$
w/o Depth-Dependent Noise	92.1$_{\pm 0.5}$	32.5$_{\pm 0.4}$	14.9$_{\pm 0.3}$
w/o Structured Noise (Perlin)	96.2$_{\pm 0.3}$	29.4$_{\pm 0.3}$	8.3$_{\pm 0.2}$
w/o Optical Distortions	96.9$_{\pm 0.3}$	28.7$_{\pm 0.3}$	7.2$_{\pm 0.2}$
w/o Calibration Uncertainties	94.5$_{\pm 0.4}$	31.0$_{\pm 0.4}$	11.7$_{\pm 0.3}$
w/o Preprocessing	95.6$_{\pm 0.4}$	30.1$_{\pm 0.3}$	9.9$_{\pm 0.3}$

TABLE V: Ablation study on depth augmentation components. Each row removes one component from the full pipeline. Results report mean$_{\pm \text{std}}$ over 5 random seeds.

2) Multi-Critic Architecture: Table VI compares single-critic/discriminator and multi-critic/discriminator architectures. The multi-critic/discriminator approach provides consistent improvements across all terrain categories in both success rate (+15.1% to +22.5%) and power degradation (-13.5% to -18.2% PDR), indicating that terrain-specific value functions and discriminators enable more stable gradient estimation during training.

Method	Stairs		Gaps		Platforms	
	SR	PDR↓	SR	PDR↓	SR	PDR↓
Single Critic/Disc.	84.0$_{\pm 0.5}$	19.2$_{\pm 0.3}$	83.8$_{\pm 0.4}$	19.2$_{\pm 0.3}$	76.3$_{\pm 0.6}$	25.4$_{\pm 0.4}$
Multi-Critic/Disc. (Ours)	**98.9**$_{\pm 0.4}$	**5.7**$_{\pm 0.2}$	**99.3**$_{\pm 0.2}$	**4.5**$_{\pm 0.2}$	**98.4**$_{\pm 0.5}$	**7.2**$_{\pm 0.3}$
Δ	+14.9	-13.5	+15.5	-14.7	+22.1	-18.2

TABLE VI: Ablation study on multi-critic and multi-discriminator architecture. Stairs column reports the average of ascending (99.2%) and descending (98.6%) configurations. Results report mean$_{\pm \text{std}}$ over 5 random seeds.

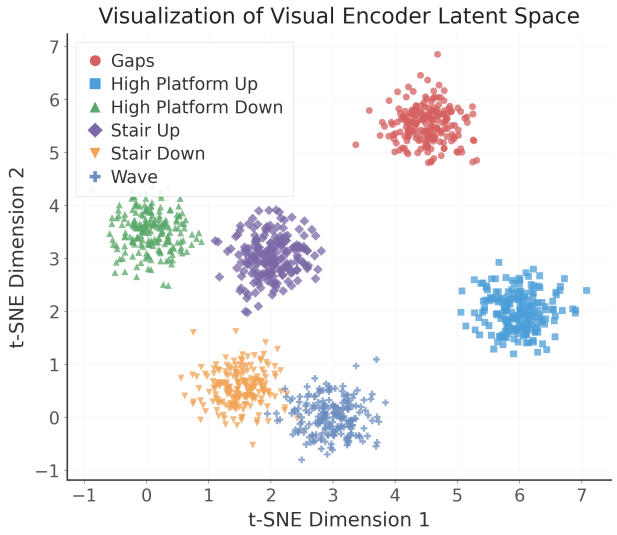

Fig. 6: t-SNE visualization of the depth encoder's latent space across six terrain types. Each terrain forms a distinct cluster, demonstrating effective terrain-specific representation learning despite realistic sensor noise.

3) Distillation Objectives: Table VII evaluates the distillation loss components. The denoising objective $\mathcal{L}_{\text{denoise}}$ contributes 5.5% SR improvement by enforcing consistent latent representations between clean and augmented depth inputs. KL regularization $\mathcal{L}_{\text{kl}}$ provides 2.8% SR improvement by preventing representation collapse. Using behavior cloning alone achieves only 86.0% SR with 17.7% PDR, demonstrating that auxiliary objectives are essential for robust vision-based deployment.

Configuration	SR (%)	P ($\times 10^1$ W)↓	PDR (%)↓
Full	**98.9**$_{\pm 0.4}$	**27.7**$_{\pm 0.3}$	**5.8**$_{\pm 0.2}$
w/o $\mathcal{L}_{\text{denoise}}$	93.4$_{\pm 0.5}$	32.3$_{\pm 0.4}$	13.3$_{\pm 0.3}$
w/o $\mathcal{L}_{\text{kl}}$	<u>96.1</u>$_{\pm 0.4}$	<u>29.5</u>$_{\pm 0.3}$	<u>8.9</u>$_{\pm 0.3}$
BC Only	86.0$_{\pm 0.6}$	32.4$_{\pm 0.4}$	17.7$_{\pm 0.4}$

TABLE VII: Ablation study on distillation loss components. Results report mean$_{\pm \text{std}}$ over 5 random seeds.

D. Latent Space Analysis

To understand how our depth encoder represents different terrains, we visualize the learned latent space using t-SNE dimensionality reduction. Figure 6 shows the latent embeddings of augmented depth observations across six terrain types.

The visualization reveals that each terrain type, stair ascending, stair descending, gap, platform ascending, platform descending, and rough terrain, forms a distinct, well-separated cluster. This clear separation indicates that our encoder learns to extract terrain-specific geometric features despite realistic sensor noise, providing the policy with discriminative information for appropriate locomotion strategies.

E. Real-World Deployment

We validate our approach on a full-sized humanoid robot equipped with a stereo depth camera. The policy runs entirely onboard at 50 Hz without any fine-tuning from simulation training.

1) Test Scenarios: We evaluate four real-world scenarios: (1) outdoor stairs with 15 cm step height (ascending and descending), (2) outdoor platforms with approximately 40 cm height, (3) extended staircases with 30+ consecutive steps, and (4) wide gaps exceeding 45 cm width.

2) Quantitative Results: Table VIII reports results from real-world deployment. The policy achieves 97.8% overall success rate (88/90 trials), with perfect performance on five out of six test conditions. The extended staircase test validates long-horizon stability over 30+ consecutive steps without accumulated drift.

3) Failure Analysis: The performance gap between stair ascending (100%) and descending (86.7%) reveals an inherent asymmetry in bipedal locomotion: during descent, gravitational acceleration amplifies minor errors into irreversible forward momentum, whereas similar misjudgments during ascent remain recoverable. Additionally, the target foot placement is partially occluded by the current step edge, precisely where stereo matching artifacts are most pronounced. These observations suggest that robust stair descent demands higher-resolution near-field depth sensing or predictive foot placement strategies.

Scenario	Success	Rate (%)
Outdoor Stairs (up)	15/15	100.0
Outdoor Stairs (down)	13/15	86.7
Outdoor Platform (up)	15/15	100.0
Outdoor Platform (down)	15/15	100.0
Extended Staircase	15/15	100.0
Wide Gap Crossing	15/15	100.0
Overall	**88/90**	**97.8**

TABLE VIII: Real-world deployment success rates across 15 trials per scenario.

4) Cross-Platform Generalization: To assess whether our training pipeline generalizes beyond the primary platform, we apply the same methodology to a Unitree G1 humanoid robot equipped with an Intel RealSense D435i depth camera, a different robot morphology and sensor modality. Preliminary results are provided in Appendix E, suggesting that our depth augmentation and multi-critic framework can be effectively transferred to new hardware configurations.

V. Conclusion and Limitations

We presented a framework for vision-based humanoid locomotion that combines realistic depth sensor simulation, multi-critic reinforcement learning, and vision-aware behavior distillation. Our depth augmentation pipeline models stereo fusion artifacts, depth-dependent noise, and calibration uncertainties, while terrain-specific critics and discriminators enable unified control across diverse scenarios. Experiments on RDT-Bench and real-world deployment demonstrate strong performance across diverse terrains without fine-tuning. However, stair descent remains less robust than ascent due to gravitational amplification of minor errors and occlusion of target footholds by step edges, which motivates future work on higher resolution near-field depth sensing and predictive foot placement strategies.

References

[1] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on robot learning*, pages 403–415. PMLR, 2023.

[2] Qingwei Ben, Botian Xu, Kailin Li, Feiyu Jia, Wentao Zhang, Jingping Wang, Jingbo Wang, Dahua Lin, and Jiangmiao Pang. Gallant: Voxel grid-based humanoid locomotion and local-navigation across 3d constrained terrains, 2025. URL https://arxiv.org/abs/2511.14625.

[3] David Bertoin, Adil Zouitine, Mehdi Zouitine, and Emmanuel Rachelson. Look where you look! saliency-guided q-networks for generalization in visual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:30693–30706, 2022.

[4] Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei Wang. Understanding domain randomization for sim-to-real transfer. *arXiv preprint arXiv:2110.03239*, 2021.

[5] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450. IEEE, 2024.

[6] Péter Fankhauser, Michael Bloesch, Christian Gehring, Marco Hutter, and Roland Siegwart. Robot-centric elevation mapping with uncertainty estimates. In *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.

[7] Péter Fankhauser, Michael Bloesch, and Marco Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3019–3026, 2018. doi: 10.1109/LRA.2018.2849506.

[8] Ambarish Goswami and Vinutha Kallem. Rate of change of angular momentum and balance maintenance of biped robots. In *International Conference on Robotics and Automation (ICRA)*, 2004.

[9] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] Nicklas Hansen, Zhecheng Yuan, Yanjie Ze, Tongzhou Mu, Aravind Rajeswaran, Hao Su, Huazhe Xu, and Xiaolong Wang. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. *arXiv preprint arXiv:2212.05749*, 2022.

[11] Corentin Hardy, Erwan Le Merrer, and Bruno Sericola. Md-gan: Multi-discriminator generative adversarial networks for distributed datasets. In *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, page 866–877. IEEE, May 2019. doi: 10.1109/ipdps.2019.00095. URL http://dx.doi.org/10.1109/IPDPS.2019.00095.

[12] Junzhe He, Chong Zhang, Fabian Jenelten, Ruben Grandia, Moritz Bächer, and Marco Hutter. Attention-based map encoding for learning generalized legged locomotion. *Science Robotics*, 10(105):eadv3604, 2025.

[13] Junzhe He, Chong Zhang, Fabian Jenelten, Ruben Grandia, Moritz BÄcher, and Marco Hutter. Attention-based map encoding for learning generalized legged locomotion, 2025. URL https://arxiv.org/abs/2506.09588.

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[15] David Hoeller, Nikita Rudin, Dhionis Sako, and Marco Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024.

[16] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. doi: 10.1007/s10514-012-9321-0. URL https://octomap.github.io. Software available at https://octomap.github.io.

[17] Weiran Huang, Mingyang Yi, and Xuyang Zhao. Towards the generalization of contrastive self-supervised learning. *arXiv preprint arXiv:2111.00743*, 2021.

[18] Yangru Huang, Peixi Peng, Yifan Zhao, Guangyao Chen, and Yonghong Tian. Spectrum random masking for generalization in image-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20393–20406, 2022.

[19] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 2019.

[20] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82 (1):35–45, 1960.

[21] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.

[22] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

[23] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 2020.

[24] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *The International Journal of Robotics Research (IJRR)*, 2024.

[25] Minsong Liu, Yuanheng Zhu, Yaran Chen, and Dongbin Zhao. Enhancing reinforcement learning via transformer-based state predictive representations. *IEEE Transactions on Artificial Intelligence*, 2024.

[26] Junfeng Long, Junli Ren, Moji Shi, Zirui Wang, Tao

Huang, Ping Luo, and Jiangmiao Pang. Learning humanoid locomotion with perceptive internal model. *arXiv preprint arXiv:2411.14386*, 2024.

[27] Junfeng Long, Junli Ren, Moji Shi, Zirui Wang, Tao Huang, Ping Luo, and Jiangmiao Pang. Learning humanoid locomotion with perceptive internal model. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9997–10003. IEEE, 2025.

[28] Antonio Loquercio, Ashish Kumar, and Jitendra Malik. Learning visual locomotion with cross-modal supervision. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7295–7302. IEEE, 2023.

[29] Bogdan Mazoure, Remi Tachet des Combes, Thang Long Doan, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. In *Advances in Neural Information Processing Systems*, 2020.

[30] Tad McGeer. Passive Dynamic Walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990. doi: 10.1177/027836499000900206. URL http://ijr.sagepub.com/content/9/2/62.abstract.

[31] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62), January 2022. ISSN 2470-9476. doi: 10.1126/scirobotics.abk2822. URL http://dx.doi.org/10.1126/scirobotics.abk2822.

[32] Takahiro Miki, Lorenz Wellhausen, Ruben Grandia, Fabian Jenelten, Timon Homberger, and Marco Hutter. Elevation mapping for locomotion and navigation using gpu. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2273–2280. IEEE, 2022.

[33] Siddharth Mysore, George Cheng, Yunqi Zhao, Kate Saenko, and Meng Wu. Multi-critic actor learning: Teaching rl policies to act with style. In *International Conference on Learning Representations (ICLR)*, 2022.

[34] I Made Aswin Nahrendra, Byeongho Yu, Minho Oh, Dongkyu Lee, Seunghyun Lee, Hyeonwoo Lee, Hyungtae Lim, and Hyun Myung. Obstacle-aware quadrupedal locomotion with resilient multi-modal reinforcement learning, 2024. URL https://arxiv.org/abs/2409.19709.

[35] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[36] Tianwei Ni, Benjamin Eysenbach, Erfan Seyedsalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive rl. *arXiv preprint arXiv:2401.08898*, 2024.

[37] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning*, pages 17359–17371. PMLR, 2022.

[38] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[39] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40(4):1–20, July 2021. ISSN 1557-7368. doi: 10.1145/3450626.3459670. URL http://dx.doi.org/10.1145/3450626.3459670.

[40] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 2024.

[41] Banafsheh Rafiee, Jun Jin, Jun Luo, and Adam White. What makes useful auxiliary tasks in reinforcement learning: investigating the effect of the target policy. *arXiv preprint arXiv:2204.00565*, 2022.

[42] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5402–5415, 2021.

[43] Rutav M Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. In *International Conference on Machine Learning*, pages 9465–9476. PMLR, 2021.

[44] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018. URL https://arxiv.org/abs/1708.07120.

[45] Haolin Song, Hongbo Zhu, Tao Yu, Yan Liu, Mingqi Yuan, Wengang Zhou, Hua Chen, and Houqiang Li. Gait-adaptive perceptive humanoid locomotion with real-time under-base terrain reconstruction, 2025. URL https://arxiv.org/abs/2512.07464.

[46] Jingkai Sun, Gang Han, Pihai Sun, Wen Zhao, Jiahang Cao, Jiaxu Wang, Yijie Guo, and Qiang Zhang. Dpl: Depth-only perceptive humanoid locomotion via realistic depth synthesis and cross-attention terrain reconstruction. *arXiv preprint arXiv:2510.07152*, 2025.

[47] Wandong Sun, Baoshi Cao, Long Chen, Yongbo Su, Yang Liu, Zongwu Xie, and Hong Liu. Learning perceptive humanoid locomotion over challenging terrain. *arXiv preprint arXiv:2503.00692*, 2025.

[48] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017.

[49] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[50] Mingle Xu, Sook Yoon, Alvaro Fuentes, and Dong Sun Park. A comprehensive survey of image augmentation techniques for deep learning. *arXiv preprint*

arXiv:2205.01491, 2022.

[51] Ruihan Yang, Minghao Zhang, Nicklas Hansen, Huazhe Xu, and Xiaolong Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. *arXiv preprint arXiv:2107.03996*, 2021.

[52] Suorong Yang, Weikang Xiao, Mengcheng Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey. *arXiv preprint arXiv:2204.08610*, 2022.

[53] Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2020.

[54] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2021.

[55] Tao Yu, Zhizheng Zhang, Cuiling Lan, Zhibo Chen, and Yan Lu. Mask-based latent reconstruction for reinforcement learning. *arXiv preprint arXiv:2201.12096*, 2022.

[56] Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, Yi Wu, Yang Gao, and Huazhe Xu. Pre-trained image encoder for generalizable visual reinforcement learning. In *First Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022*, 2022.

[57] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[58] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Soeren Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023.

[59] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Soeren Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning, 2023. URL https://arxiv.org/abs/2309.05665.

[60] Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning. In *Conference on Robot Learning (CoRL)*, 2024.

APPENDIX

A. Network Architectures

We present the detailed network architectures for the teacher policy, student policy, and multi-critic networks used in our framework. The teacher and student policies share identical architectures except for their exteroceptive encoders: the teacher uses an MLP to process privileged height scan observations, while the student uses a CNN to process depth images. During distillation, the student's CNN encoder learns to produce latent representations that match the teacher's MLP encoder output, enabling seamless transfer of the learned locomotion policy from privileged to vision-based observations.

Component	Configuration
Height Scan Encoder	
Input Layer	Height Scan ($21 \times 33 = 693$)
Hidden Layer 1	Linear($693 \rightarrow 512$) + SiLU
Hidden Layer 2	Linear($512 \rightarrow 256$) + SiLU
Output Layer	Linear($256 \rightarrow 128$)
Recurrent Module	
Input	Concat(Proprio[96], Encoder[128]) = 224
RNN Type	GRU (1 layer)
Hidden Dimension	256
Actor MLP	
Input Layer	GRU Output (256)
Hidden Layer 1	Linear($256 \rightarrow 512$) + SiLU
Hidden Layer 2	Linear($512 \rightarrow 256$) + SiLU
Hidden Layer 3	Linear($256 \rightarrow 128$) + SiLU
Output Layer	Linear($128 \rightarrow 29$)
Policy Distribution	
Distribution Type	Gaussian
Initial Noise Std	1.0

TABLE IX: Teacher Policy Network Architecture

Component	Configuration
Depth Image Encoder (CNN)	
Input Layer	Depth Image ($1 \times 24 \times 32$)
Conv Layer 1	Conv2d($1 \rightarrow 32$, k=3, s=2, p=1) + SiLU
Conv Layer 2	Conv2d($32 \rightarrow 64$, k=3, s=2, p=1) + SiLU
Conv Layer 3	Conv2d($64 \rightarrow 128$, k=3, s=2, p=1) + SiLU
Flatten	$128 \times 3 \times 4 = 1536$
Output Layer	Linear($1536 \rightarrow 128$)
Recurrent Module	
Input	Concat(Proprio[96], Encoder[128]) = 224
RNN Type	GRU (1 layer)
Hidden Dimension	256
Actor MLP	
Input Layer	GRU Output (256)
Hidden Layer 1	Linear($256 \rightarrow 512$) + SiLU
Hidden Layer 2	Linear($512 \rightarrow 256$) + SiLU
Hidden Layer 3	Linear($256 \rightarrow 128$) + SiLU
Output Layer	Linear($128 \rightarrow 29$)
Policy Distribution	
Distribution Type	Gaussian
Constant Noise Std	0.1

TABLE X: Student Policy Network Architecture

Component	Configuration
Height Scan Encoder (Shared)	
Input Layer	Height Scan ($21 \times 33 = 693$)
Hidden Layer 1	Linear($693 \rightarrow 512$) + SiLU
Hidden Layer 2	Linear($512 \rightarrow 256$) + SiLU
Output Layer	Linear($256 \rightarrow 128$)
Recurrent Module (Shared)	
Input	Concat(Proprio[96], Encoder[128]) = 224
RNN Type	GRU (1 layer)
Hidden Dimension	256
Critic MLP (Shared Backbone)	
Input Layer	GRU Output (256)
Hidden Layer 1	Linear($256 \rightarrow 512$) + SiLU
Hidden Layer 2	Linear($512 \rightarrow 256$) + SiLU
Hidden Layer 3	Linear($256 \rightarrow 128$) + SiLU
Terrain-Specific Output Heads	
Stair Head	Linear($128 \rightarrow 1$)
Gap Head	Linear($128 \rightarrow 1$)
General Head	Linear($128 \rightarrow 1$)

TABLE XI: Multi-Critic Network Architecture

B. Implementation Details

We provide the detailed hyperparameters for our policy distillation framework in Table XII. The distillation process transfers the learned locomotion policy from the teacher (which operates on privileged height scan observations) to the student (which operates on depth images). We employ a cosine annealing learning rate schedule with warm-up to ensure stable training dynamics. The loss coefficients are carefully balanced to ensure that behavior cloning provides the primary learning signal while the denoising and KL regularization objectives contribute to robustness against sensor noise and prevent representation collapse, respectively. We use gradient accumulation to stabilize training with large effective batch sizes and apply gradient clipping to prevent training instabilities.

Parameter	**Value**
(a) Training Hyperparameters	
Steps per environment per iteration	800
Total training iterations	4000
Learning rate schedule	One Cycle [44]
Initial learning rate	1×10^{-3}
Div factor (peak/init)	10.0
Final div factor (final/init)	50.0
Gradient accumulation steps	10
Max gradient norm	1.0
(b) Loss Coefficients	
Behavior loss weight	1.0
Denoising loss coeff. λ_{denoise}	0.1
KL loss coeff. λ_{kl}	0.1
EMA decay	0.997

TABLE XII: Policy Distillation Configuration

C. CycleGAN Hyperparameters

We train a CycleGAN model to translate between simulated and real-world depth images for evaluation purposes. The real-world dataset consists of approximately 2 hours of depth recordings from the humanoid robot traversing various terrains, yielding approximately 200,000 depth frames. The CycleGAN is used *exclusively for evaluation* to inject realistic depth artifacts into simulation, enabling fair comparison of sim-to-real transfer capabilities across different training augmentation strategies.

1) Translation Quality Evaluation: To validate that our CycleGAN produces realistic depth translations, we evaluate the model using standard image-to-image translation metrics on a held-out test set comprising 10% of the data (approximately 20,000 frames).

Metric	Sim→Real	Real→Sim
Translation Quality		
FID ↓ (no translation baseline)	67.2	
FID ↓ (after CycleGAN)	23.4	21.7
KID ($\times 10^{-3}$) ↓	8.2	7.6
Cycle Consistency		
SSIM ↑	0.89	0.91
PSNR ↑	28.3 dB	29.1 dB
LPIPS ↓	0.12	0.11

TABLE XIII: CycleGAN Translation Quality Metrics. **Translation Quality**: FID and KID measure distributional distance between translated images $G(x_{\text{source}})$ and real target domain images x_{target}. The baseline FID (67.2) is computed between raw simulation and real depth images without translation. **Cycle Consistency**: SSIM, PSNR, and LPIPS evaluate reconstruction quality of $F(G(x)) \approx x$, measuring how well geometric structure is preserved through the translation cycle. Images are resized to 128×128 before FID computation using bilinear interpolation.

The FID score of 23.4 for Sim→Real translation indicates that the generated depth images closely match the statistical properties of real sensor outputs. The high cycle-consistency metrics (SSIM = 0.89, PSNR = 28.3 dB) confirm that the translation preserves geometric structure while introducing realistic sensor artifacts.

Parameter	Value
(a) Dataset Statistics	
Total real-world recording time	∼2 hours
Total depth frames	∼200,000
Frame resolution	24×32
Train/Val split	90% / 10%
(b) Network Architecture	
Generator architecture	ResNet (9 blocks)
Discriminator architecture	PatchGAN (70×70)
Number of filters (first layer)	64
Normalization	Instance Normalization
(c) Training Hyperparameters	
Batch size	64
Total epochs	200
Optimizer	Adam
Learning rate	2×10^{-4}
β_1, β_2	0.5, 0.999
Learning rate decay	Linear (after epoch 100)
(d) Loss Weights	
Adversarial loss weight	1.0
Cycle consistency loss weight λ_{cyc}	10.0
Identity loss weight λ_{idt}	0.5

TABLE XIV: CycleGAN Training Configuration

D. Terrain Specific Rewards

Our multi-critic framework employs terrain-specific reward functions to encourage appropriate locomotion strategies for different obstacle types. We detail the key terrain-specific rewards below.

1) Velocity Tracking Rewards: We employ two distinct velocity tracking formulations depending on terrain requirements:

Exponential Velocity Tracking ($r_{\text{vel}}^{\text{exp}}$) is used for stairs, platforms, and rough terrain, where precise velocity regulation ensures stable foot placement:

$$r_{\text{vel}}^{\text{exp}} = \exp\left(-\frac{\|\mathbf{v}_{xy}^{\text{cmd}} - \mathbf{v}_{xy}^{\text{robot}}\|^2}{\sigma^2}\right) \tag{13}$$

where $\mathbf{v}_{xy}^{\text{cmd}}$ denotes the commanded velocity, $\mathbf{v}_{xy}^{\text{robot}}$ denotes the robot's base linear velocity in the body frame, and σ is a temperature parameter controlling tracking precision. This exponential kernel provides smooth gradients that encourage accurate velocity following, which is critical for controlled stepping on elevated surfaces.

Directional Velocity Tracking ($r_{\text{vel}}^{\text{dir}}$) is used for gap terrain, where the robot benefits from moving faster than the commanded

velocity to execute dynamic crossing motions:

$$r_{\text{vel}}^{\text{dir}} = \frac{\min\left(\mathbf{v}_{xy}^{\text{robot}} \cdot \hat{\mathbf{d}}_{\text{cmd}},\ \|\mathbf{v}_{xy}^{\text{cmd}}\|\right)}{\|\mathbf{v}_{xy}^{\text{cmd}}\| + \epsilon} \tag{14}$$

where $\hat{\mathbf{d}}_{\text{cmd}} = \mathbf{v}_{xy}^{\text{cmd}} / \|\mathbf{v}_{xy}^{\text{cmd}}\|$ is the unit direction of the commanded velocity, and ϵ is a small constant for numerical stability. This formulation rewards movement in the commanded direction without penalizing speeds exceeding the command magnitude, enabling the robot to build momentum for successful gap traversal.

2) Feet Contact Height Reward: For stairs and platform terrains, we introduce a feet contact height reward (r_{contact}) that encourages the robot to place its feet on flat, stable surfaces:

$$r_{\text{contact}} = \sum_{f \in \{\text{left,right}\}} \mathbb{1}_{\text{contact}}^{f} \cdot \text{std}\left(\text{clip}(h_f^{\text{scan}}, -h_{\max}, h_{\max})\right) \tag{15}$$

where h_f^{scan} denotes the height scan measurements around foot f relative to the foot position, $\mathbb{1}_{\text{contact}}^{f}$ is an indicator function that equals 1 when foot f is in contact with the ground, and $h_{\max}$ is a clipping threshold. The standard deviation of clipped height values around each foot serves as a measure of surface irregularity, lower values indicate flatter contact surfaces. This reward is applied as a penalty (with negative weight) to discourage foot placement on edges or uneven surfaces, which is particularly important for stair and platform traversal where precise foot placement determines stability.

This reward is *not* applied to gap and rough terrains: gap crossing requires dynamic leaping motions where contact surface analysis is less relevant, while rough terrain inherently features irregular surfaces where penalizing height variation would be counterproductive.

Reward	Stairs/Platforms	Gaps	Rough
$r_{\text{vel}}^{\text{exp}}$	✓	–	✓
$r_{\text{vel}}^{\text{dir}}$	–	✓	–
r_{contact}	✓	–	–

TABLE XV: Terrain-Specific Reward Configuration

E. Cross-Platform Validation

We conduct a preliminary cross-platform validation to assess whether the learned policy generalizes beyond the primary training platform. We deploy our policy on a Unitree G1 humanoid robot, which differs from our primary platform in both kinematic structure and sensor configuration (Intel RealSense D435i vs. Orbbec Gemini 336L).

Scenario	Success	Rate (%)
Stair Ascending	15/15	100.0

TABLE XVI: Cross-Platform Validation on Unitree G1

As shown in Table XVI and Figure 7, the policy achieves perfect success rate on the stair ascending task without any platform-specific adaptation. This result provides initial evidence that our depth augmentation strategy learns transferable geometric representations rather than sensor-specific artifacts. However, we note that this validation is limited to a single terrain type; comprehensive cross-platform benchmarking across gaps, platforms, and descending scenarios remains an important direction for future work.

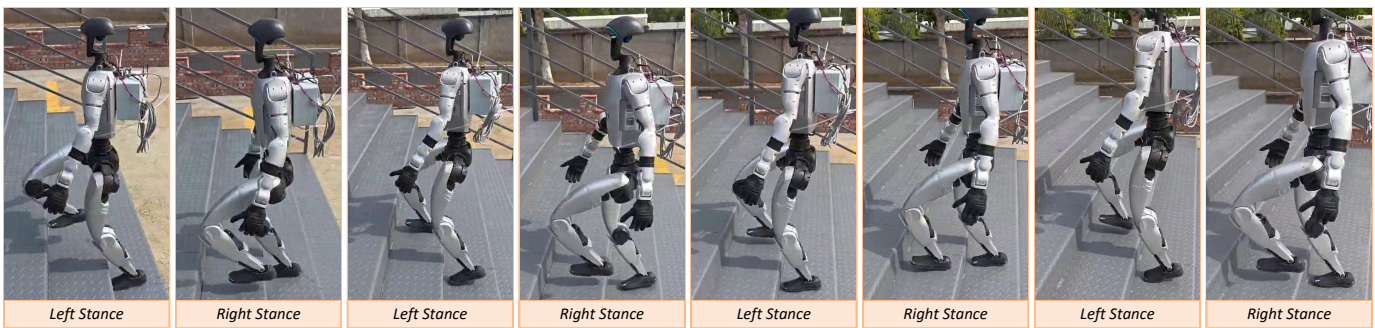

Fig. 7: Cross-platform deployment on Unitree G1 humanoid robot ascending outdoor stairs. The policy transfers zero-shot from training on a different platform with different depth sensor, demonstrating the generality of the learned depth representations.

F. Additional Augmentation Results

We provide additional visualization examples of our depth augmentation pipeline in Figure 8. Each row shows a triplet consisting of: (1) the left camera depth image, (2) the right camera depth image, and (3) the augmented depth output before spatial cropping. All depth values are normalized to the range $[0, 2]$ meters and converted to color maps for intuitive visualization, where cooler colors (blue/purple) indicate closer surfaces and warmer colors (red/yellow) represent farther distances.

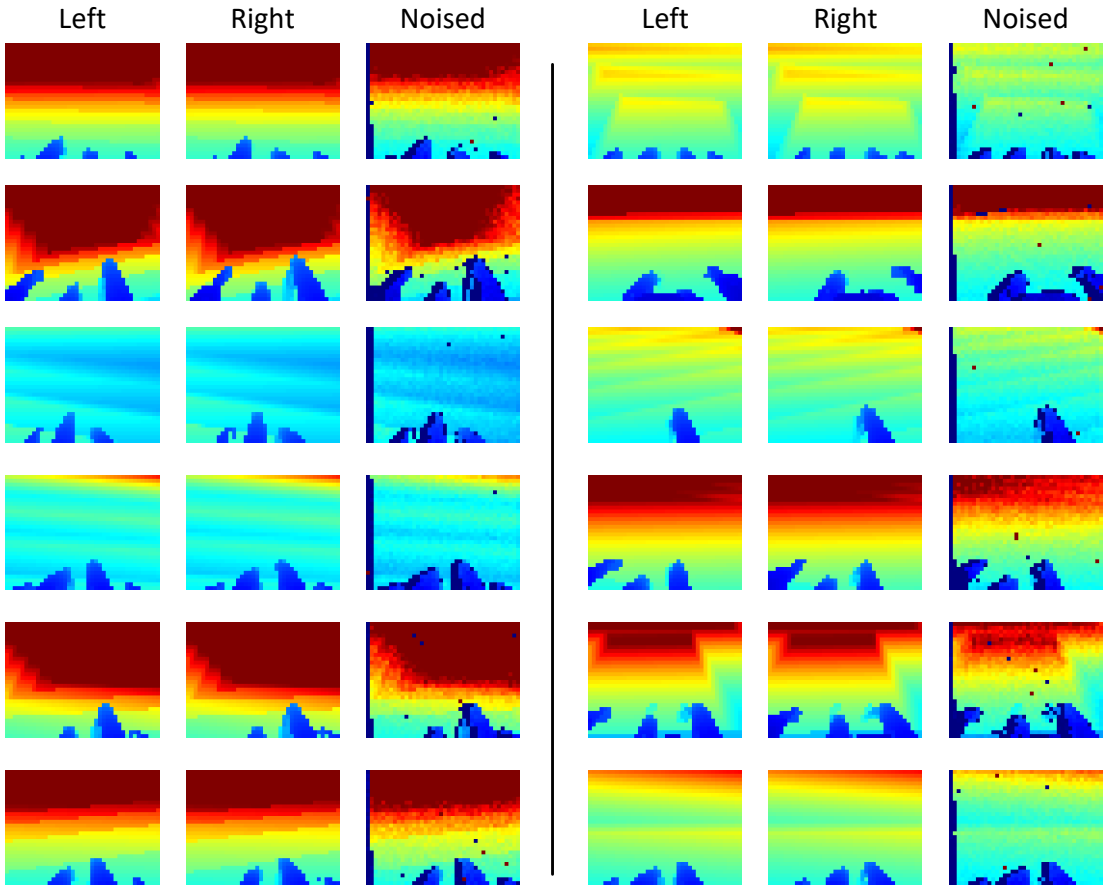

Fig. 8: Additional depth augmentation examples across diverse terrains. Each triplet shows (left to right): left camera depth, right camera depth, and augmented output before spatial cropping. Depth values are normalized to $[0, 2]$ m and rendered as color maps (cool = near, warm = far). The augmented images exhibit realistic stereo fusion holes (black regions), depth-dependent noise, and structured Perlin patterns while preserving terrain geometry essential for locomotion control.