SSR: Scaling Surefooted and Symmetric Humanoid Traversal to the Open World

Ruiqi Yu[1]* **Yiwen Wang**[1]* **Yuan Hao**[1] **Jun Wu**[1] **Qiuguo Zhu**[1]†
[1]Zhejiang University

Figure 1: SSR scales surefooted and symmetric humanoid traversal to diverse open-world human environments. Driven by egocentric vision, the robot reliably traverses grassy slopes, stairs of varying sizes and structures, steps onto high platforms, and crosses wide gaps. Throughout long-horizon traversal, it maintains safe foot placement together with coordinated and natural whole-body motion.

Abstract: Extending humanoid traversal to the open world is key to practical deployment in human environments, but remains challenging. The robot must use vision to ensure safe and reliable foot placement on heterogeneous terrain under highly dynamic motion, while producing coordinated, natural whole-body behaviors. We propose **SSR**, an efficient end-to-end framework for egocentric vision-based humanoid traversal that jointly learns these capabilities. SSR introduces *imagined foothold guidance*, which learns to model forthcoming swing-foot contacts and evaluates their support to guide pre-touchdown swings toward stable regions, reducing edge slips. It further employs *equivariant latent-space symmetry augmentation* to efficiently induce bilateral coordination under high-dimensional visual observations, and uses *terrain-specific multi-discriminator motion priors* to encourage human-like behavior across scenes. Extensive experiments show that SSR achieves safe, stable, and high-quality locomotion on diverse real-world terrains, including stairs with varied structures and extreme challenges such as wide gaps and high platforms, while enabling reliable long-horizon traversal in open outdoor environments. Project website: `https://ssr-humanoid.github.io/`.

Keywords: Open-World Humanoid Traversal, Safe Foot Placement, Equivariant Symmetry Learning

*Equal contribution
†Corresponding author

1 Introduction

Open-world human environments require humanoids to traverse heterogeneous terrains with agile, stable, and natural whole-body motion, from everyday stairs to wide gaps, and high platforms [1]. This demands that the robot perceive changing surroundings, place each step within safe support regions, and maintain coordinated, human-like behavior throughout long-horizon locomotion [2].

Recent learning-based humanoid locomotion methods leverage onboard perception to adapt to terrain changes. Among them, policies conditioned on egocentric depth images are especially promising for dynamic real-world scenes, as they avoid explicit, noise-sensitive mapping [3, 4, 5, 6]. However, turning such perception into robust open-world traversal remains an open challenge. The policy must infer local terrain structure from visual cues and convert this understanding into reliable foothold decisions. This is critical for flat-footed humanoids, since stepping near terrain edges sharply reduces the available support area and increases the risk of slips or falls. Beyond safety, traversal in human environments also requires coordinated, controllable, and natural whole-body motion. Prior work often introduces symmetry priors [7, 8, 9] or human-motion imitation [10, 11, 12], yet integrating them into egocentric-vision policies can be inefficient and brittle. High-dimensional visual observations make symmetry learning computationally expensive, while terrain-dependent dynamics can destabilize a shared human-like style prior. Overall, an efficient framework that jointly enables safe foot-placement behavior, symmetric and natural whole-body motion, and robust traversal across diverse terrains under vision is still lacking.

To address these challenges, we propose **SSR**, a unified single-stage end-to-end reinforcement learning (RL) framework for vision-based humanoid traversal in complex real-world environments. SSR learns directly from egocentric depth and proprioception to jointly acquire reliable foot placement and high-quality whole-body motion, coupling gait symmetry with human-like behavior.

For safe and accurate foot placement, we first introduce imagined foothold guidance during training. It learns to model forthcoming swing-foot contacts and evaluates their local support before touchdown, converting sparse contact-time safety assessment into dense predictive guidance. Unlike prior methods that assess foothold safety only at or after contact, leaving swing-phase actions with delayed feedback [3, 4, 13], this imagined interaction enables the policy to steer the foot early toward flat and well-supported regions, thereby reducing edge contacts, slips, and falls.

To efficiently acquire coordinated whole-body motion, we develop equivariant latent-space symmetry augmentation. It applies mirror transformations to compact latent representations rather than high-dimensional visual observations. Compared with input-level augmentation and symmetry regularization, as well as strict equivariant architectures [8, 9, 14], this strategy reduces the overhead of symmetry learning for vision-based policies while preserving flexibility for exploration. We further incorporate a terrain-specific multi-discriminator adversarial motion prior [15] to capture terrain-conditioned styles and maintain human-like motion across scenes.

We evaluate SSR in diverse environments. As shown in Fig. 1, the robot robustly traverses slopes, rough ground, and stairs with varied structures. It can climb onto platforms up to 45 cm high, about $1.6\times$ its shank length, and cross gaps as wide as 90 cm. More importantly, these capabilities transfer to complex outdoor scenes and support reliable long-horizon locomotion with precise foot placement, strong bilateral coordination, robust controllability, and natural motion.

In summary, our main contributions are as follows:

1. We present a novel single-stage training framework that jointly learns safe, symmetric, and human-like motion for vision-based humanoid traversal in diverse environments.
2. We introduce imagined foothold guidance, which learns foot-contact foresight, providing dense swing-phase signals for timely correction of unsafe landing decisions and robust foot placement.
3. We develop equivariant latent-space augmentation for efficient symmetry learning, and combine it with terrain-specific style priors to jointly improve whole-body motion quality.
4. We validate the learned policy through extensive indoor and outdoor experiments, demonstrating agile and reliable humanoid locomotion over heterogeneous and challenging open-world terrains.

2 Related Work

Learning Perceptive Humanoid Locomotion. Data-driven RL has advanced humanoid locomotion under complex contact dynamics and environmental variation [1, 16, 17], and exteroceptive sensing lets controllers adapt to terrain before touchdown. Earlier methods construct height maps from LiDAR and odometry [18, 19, 20, 21], but depend on accurate localization and are sensitive to latency, drift, limited update rates, and occlusion. Recent approaches infer terrain directly from egocentric depth images [3, 4, 5, 22, 23], reducing reliance on mapping. However, many vision-based humanoid systems still target structured settings or require multi-stage pipelines, expert distillation, or dedicated depth modeling and augmentation [5, 17, 24, 25, 26]. In contrast, we learn a unified traversal policy from egocentric vision in a single-stage framework for diverse unstructured terrains.

Safe Foot Placement. Classical footstep control decomposes locomotion into perception, planning, and tracking [27, 28, 29]. While effective in specific settings, they rely on hand-crafted design and conservative assumptions. RL-based methods learn foot placement end to end by tracking planner-generated footholds [30, 31, 32, 33] or penalizing unsafe landing regions [3, 4, 13, 34]. Yet these signals are often sparse, appearing only at or after contact, weakening swing-phase guidance and training efficiency [16, 35]. Liu et al. [25] densify learning by swing-trajectory tracking, but still depend on rule-based foothold generation and focus on stairs. Zhu et al. [36] reward derived foothold actions to improve landing quality. We learn to imagine future footholds during swing, assessing support before touchdown to guide flatter, safer landings without planners or edge detection.

Symmetry Learning in Legged Locomotion. High-quality humanoid traversal requires coordinated bilateral motion. Prior work exploits morphological symmetry via data augmentation or regularization [8, 9, 14, 37], but under visual inputs with short-term memory, these methods require mirrored forward passes on high-dimensional inputs or mirrored hidden-state rollout, increasing computational and memory cost. Strictly equivariant architectures reduce this overhead [7], but may limit symmetry breaking near neutral states for gait exploration [38]. We use an equivariant encoder to move augmentation into latent space, balancing efficiency and flexibility in symmetry learning.

3 Method

Our goal is safe and high-quality humanoid traversal over diverse unstructured terrains directly from proprioception and raw depth in a unified, efficient framework. SSR achieves this with imagined foothold guidance for dense pre-contact swing correction (Sec. 3.2), equivariant latent-space symmetry augmentation for efficient bilateral gait learning via compact latent mirroring (Sec. 3.3), and cross-terrain motion priors for human-like whole-body behavior (Sec. 3.4). Fig. 2 summarizes SSR.

3.1 Single-Stage Learning of Generalizable Traversal Skills

We formulate humanoid locomotion as a partially observable Markov decision process (POMDP) and optimize the policy with PPO [39] under an asymmetric actor-critic architecture [40].

Observations and Action. At timestep t, the policy receives proprioceptive and visual inputs. The proprioception $\mathbf{o}_t^p \in \mathbb{R}^{72}$ includes base angular velocity $\boldsymbol{\omega}_t$, projected gravity $\mathbf{g}_t$, velocity commands $\mathbf{u}_t$, joint positions $\boldsymbol{\theta}_t$, joint velocities $\dot{\boldsymbol{\theta}}_t$, and previous action $\mathbf{a}_{t-1}$. We stack a short history $\mathbf{o}_{t-h+1:t}^p$ to reduce partial observability. Together with the depth image $\mathbf{I}_t \in \mathbb{R}^{36\times 36}$, this forms the policy observation $\mathbf{o}_t = (\mathbf{o}_{t-h+1:t}^p, \mathbf{I}_t)$, which maps to actuated joint position targets $\mathbf{a}_t \in \mathbb{R}^{21}$. The critic additionally uses privileged information, including ground-truth base velocity $\mathbf{v}_t$, foot velocities $\mathbf{v}_t^f$, contact states $\mathbf{c}_t$, limb positions $\mathbf{p}_t^h$ and $\mathbf{p}_t^f$, and height maps around the feet and body, $\mathbf{H}_t^f$ and $\mathbf{H}_t^b$.

Policy Architecture. Our policy comprises a recurrent cross-modal encoder, a motion-state estimator, and a MoE actor [41, 42]. The encoder extracts compact robot-environment context for environmental awareness and adaptability [43, 44]. It encodes the image with a convolutional neural network (CNN) and temporal proprioception with a multi-layer perceptron (MLP), then fuses them with a gated recurrent unit (GRU) into a latent representation with three heads, $\hat{\mathbf{z}}_t = [\hat{\mathbf{z}}_t^f, \hat{\mathbf{z}}_t^b, \hat{\mathbf{z}}_t^p]^\top$. Here, $\hat{\mathbf{z}}_t^f$ and $\hat{\mathbf{z}}_t^b$ capture foot- and base-centric terrain geometry and decode $\hat{\mathbf{H}}_t^f$ and $\hat{\mathbf{H}}_t^b$ [45], while $\hat{\mathbf{z}}_t^p$, learned with a variational autoencoder (VAE), predicts next proprioception $\hat{\mathbf{o}}_{t+1}^p$ to model system dynamics [40]. A separate MLP estimator predicts base velocity $\hat{\mathbf{v}}_t$ from temporal proprioception. During training, they are optimized with a hybrid prediction loss. We further impose mirror

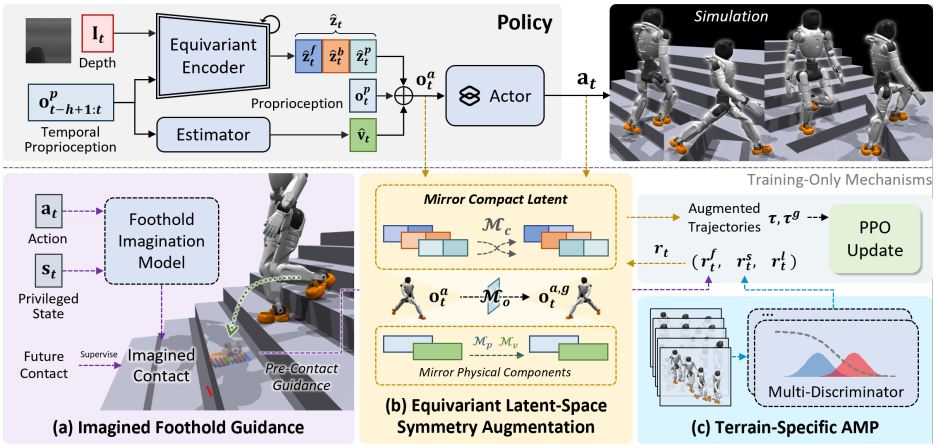

Figure 2: **Overview of the SSR framework**. The policy combines a recurrent equivariant encoder, an estimator, and a MoE actor to learn unified traversal across terrains from egocentric depth images and temporal proprioception. During training, SSR learns surefooted, symmetric, and human-like motion through three mechanisms: (a) imagined foothold guidance for dense pre-contact swing correction, (b) equivariant latent augmentation for efficient symmetry learning, and (c) terrain-specific multi-discriminator AMP.

equivariance on the encoder to support symmetry learning in Sec. 3.3. The actor takes current proprioception, $\hat{\mathbf{v}}_t$, and $\hat{\mathbf{z}}_t$ as input, allowing one policy to represent terrain-dependent motion modes.

Reward Functions. We group the objective into r^l for task-oriented objectives, r^f for foothold guidance, and r^s for human-like motion style. We use three critics to improve estimation of corresponding value functions [46, 47]. See detailed reward definitions in Appendix A.3.

3.2 Imagined Foothold Guidance for Foresighted Foot Placement Correction

Imagining Future Footholds During Swing. During training, a foothold imagination model learns to map privileged state $\mathbf{s}_t$ and action $\mathbf{a}_t$ to a prospective contact distribution $\hat{\mathbf{F}}_t = \{\hat{F}_{i,t}\}_{i=1}^2$. For foot i, $\hat{F}_{i,t} = (\hat{\boldsymbol{\mu}}_{i,t}, \hat{\sigma}_{i,t})$ parameterizes a Gaussian imagined-contact distribution in the current base frame: $q_{i,t}(\mathbf{p}) = \mathcal{N}(\mathbf{p}; \hat{\boldsymbol{\mu}}_{i,t}, (\hat{\sigma}_{i,t})^2\mathbf{I})$, where $\hat{\boldsymbol{\mu}}_{i,t} \in \mathbb{R}^2$ denotes the anticipated contact location and $\hat{\sigma}_{i,t} \in \mathbb{R}$ captures uncertainty. This probabilistic form yields smoother support estimates over the imagined landing region. The supervision target $\mathbf{p}^*_{i,t}$ is foot i's first valid future contact location, expressed in base frame. We train the model with the Gaussian negative log-likelihood:

$$\mathcal{L}_{\text{pred}} = -\sum_{i=1}^{2} \log q_{i,t}(\mathbf{p}^*_{i,t}) \propto \sum_{i=1}^{2} \left(\frac{||\mathbf{p}^*_{i,t} - \hat{\boldsymbol{\mu}}_{i,t}||_2^2}{2(\hat{\sigma}_{i,t})^2} + 2\log\hat{\sigma}_{i,t} \right). \tag{1}$$

Early in training, unstable gaits produce noisy future-contact targets. We therefore use terrain level as a reliability proxy and enable pre-contact guidance beyond a preset curriculum level.

Guiding Safer Swings Before Touchdown. Given the imagined landing distribution, we form the guidance signal by measuring the unsupported fraction within a sole-covered region. Let $\mathbf{H}^f(\mathbf{p}) = \{h_k(\mathbf{p})\}_{k=1}^n$ denote the height map of a 22.5 cm $\times$10 cm sole patch centered at $\mathbf{p} \in \mathbb{R}^2$, sampled every 2.5 cm. Here, $h_k(\mathbf{p})$ is the terrain height at $\mathbf{p} + \delta_k$ and δ_k is the k-th offset. We define *support deficiency* at foothold $\mathbf{p}$ as:

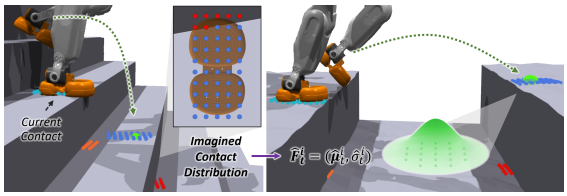

Figure 3: **Imagined foothold guidance**. We measure support deficiency over a sole-sized terrain patch. During swing, a foothold imagination model anticipates future-contact distributions for pre-contact dense guidance.

$$\rho(\mathbf{p}) = 1 - \frac{1}{n}\sum_{k=1}^{n} \mathbb{1}\{[h^f(\mathbf{p}) - h_k(\mathbf{p})] < \epsilon_h\}, \tag{2}$$

where $h^f(\mathbf{p})$ is the sole height. For imagined pre-contact locations, we use $h^f(\mathbf{p}) = \max_k h_k(\mathbf{p})$. Larger $\rho(\mathbf{p})$ indicates less support overlap, often near terrain edges or suspended regions. As each foot alternates between stance and swing, we instantiate this *foothold guidance* in the RL objective:

$$r^f_t = \exp(-(\sum_{i=1}^{2} \tilde{\rho}_{i,t})^2/\sigma_f), \quad \tilde{\rho}_{i,t} = c_{i,t}\rho(\mathbf{p}_{i,t}) + (1 - c_{i,t})\mathbb{E}_{\mathbf{p}\sim q_{i,t}}\rho(\mathbf{p}), \tag{3}$$

where $c_{i,t}$ indicates whether foot i is in stance. Stance feet are evaluated at current contacts, while swing feet use expected deficiency under the imagined-contact distribution $q_{i,t}$, approximated with discrete Gaussian-weighted samples. This formulation converts touchdown-only safety signals into dense pre-contact guidance, encouraging earlier foot-placement correction before touchdown.

3.3 Efficient Symmetry Learning with Equivariant Latent-Space Augmentation

Input-level symmetry augmentation is costly for visual recurrent policies, as each mirrored sample entails re-encoding the depth image and rolling out a mirrored memory state. We instead encode the original observation once and augment the compact actor input. The key requirement is a mirror-equivariant encoder, so that mirroring the latent is equivalent to encoding the mirrored observation.

Mirror Transformations. Let $\mathcal{M}_o$, $\mathcal{M}_s$, and $\mathcal{M}_a$ denote mirror transformations for actor observation, privileged state, and action. Since $\mathbf{o}_t^a = [\mathbf{o}_t^p, \hat{\mathbf{v}}_t, \hat{\mathbf{z}}_t]^\top$, $\mathcal{M}_o$ combines the physical proprioceptive and velocity transforms $\mathcal{M}_p$ and $\mathcal{M}_v$ with the latent transform $\mathcal{M}_c$. For latent variables without predefined mirror signs, we impose a symmetry-structured representation: each head $\hat{\mathbf{z}}_t^{(k)}$, $k \in \{f, b, p\}$, is organized as paired left-right channel groups, and $\mathcal{M}_c$ swaps the two groups. Detailed component-wise rules are listed in Appendix B.1.

Equivariant Latent Encoding. To make this latent transform valid, the encoder must map mirrored observations to mirrored latents. We first organize its inputs into symmetry-structured form: a fixed operator $\mathcal{T}$ reorganizes temporal proprioception as $\bar{\mathbf{o}}_t^p = \mathcal{T}(\mathbf{o}_{t-h+1:t}^p)$, while the first CNN layer lifts the image into paired channels. Inspired by Cesa et al. [48], we then implement the MLP, CNN, and GRU with equivariant linear or convolutional layers. For each head $E_\phi^{(k)}$,

$$E_\phi^{(k)}(\mathcal{M}_c \bar{\mathbf{o}}_t^p, \mathcal{M}_{2\mathrm{D}} I_t) = \mathcal{M}_c E_\phi^{(k)}(\bar{\mathbf{o}}_t^p, I_t), \quad (4)$$

where $\mathcal{M}_{2\mathrm{D}}$ denotes horizontal reflection. Thus, the encoder compresses original inputs into a low-dimensional latent that supports channel-swap mirroring. Appendix B.2 gives the construction and proof.

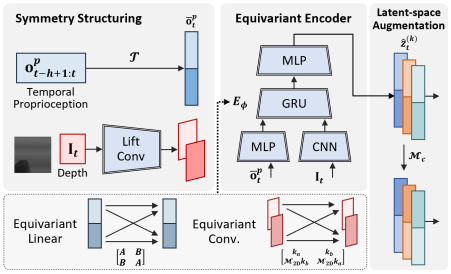

Figure 4: **Equivariant latent-space augmentation**. We form symmetry-structured inputs and build the encoder from equivariant linear and convolutional layers, allowing each latent head to be mirrored by $\mathcal{M}_c$.

Latent-Space Symmetry Augmentation. During training, the encoder runs only on the original observation. We then mirror the compact actor observation, privileged state, and action: $\mathbf{o}_t^{a,g} = \mathcal{M}_o \mathbf{o}_t^a$, $\mathbf{s}_t^g = \mathcal{M}_s \mathbf{s}_t$, and $\mathbf{a}_t^g = \mathcal{M}_a \mathbf{a}_t$. This gives a mirrored trajectory $\tau^g = (\mathbf{o}_0^{a,g}, \mathbf{s}_0^g, \mathbf{a}_0^g, r_0 \ldots)$, which is appended to the original rollout τ for PPO updates [8]. Compared with input-level augmentation, this avoids mirrored image encoding and recurrent hidden-state rollout, reducing memory and time while retaining flexible symmetry learning.

3.4 Cross-Terrain Motion Priors with Multiple Discriminators

To encourage terrain-appropriate motion style, we employ multiple discriminators [15], one for each terrain type i. Each D_i models terrain-specific reference motion style from a five-frame history $\boldsymbol{\psi}_t$, where each frame $\mathbf{s}_t^{\mathrm{amp}} \in \mathbb{R}^{63}$ contains projected gravity, base linear and angular velocities, joint positions and velocities, and limb positions. Following AMP [10], D_i is trained to distinguish reference from policy motions and produce a style reward that encourages terrain-appropriate natural behavior. We collect motion datasets for each terrain type. See details in Appendix A.8.

4 Experiments

4.1 Experimental Setup

We evaluate SSR in extensive simulation and real-world experiments to answer three questions: (1) Can it learn a unified policy for high-performance traversal over diverse challenging terrains? (2) Do the key designs improve motion quality and learning efficiency? (3) Can it transfer zero-shot to unseen open-world scenes with long-horizon robustness and generalization?

Training Environment. We perform single-stage training in Isaac Gym [49] with 4,096 AgiBot X2 humanoids on an NVIDIA RTX 4090 for about 20k iterations. We use NVIDIA Warp [50] to render deployment-consistent depth with terrain and self-occlusion. See Appendices A and C.

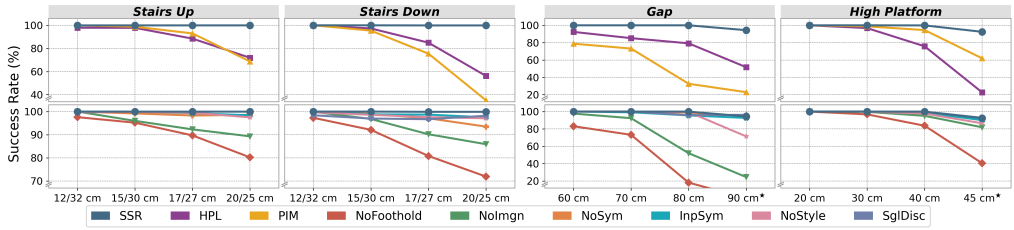

Figure 5: Traversal performance across terrains and difficulty levels in simulation. Top row compares SSR with prior methods, and bottom row reports ablation results. ★ denotes difficulties beyond the training curriculum.

Onboard Implementation. On the real robot, a waist-mounted Intel RealSense D435i captures forward-facing depth at 60 Hz. Images are downsampled and cropped from 640×360 to 36×36. An onboard Jetson AGX Orin runs network inference and outputs actions at 50 Hz.

Compared Baselines. We compare SSR with two perceptive baselines: (1) *HPL* [3], a multi-stage egocentric vision parkour method; and (2) *PIM* [18], a single-stage height-map traversal method with a hybrid internal model. We further consider ablations: (3) *NoFoothold* removes foothold guidance r^f; (4) *NoImgn* removes the imagination branch and keeps only the contact-time support assessment in r^f; (5) *NoSym* disables data augmentation and uses a non-equivariant encoder; (6) *InpSym* applies input-level mirroring with a non-equivariant encoder; as its peak memory exceeds 24 GB, this baseline alone was trained on a 48 GB RTX 4090; (7) *NoStyle* removes the AMP style reward r^s; and (8) *SglDisc* replaces multiple terrain-specific discriminators with a shared one.

Evaluation Metric and Protocol. We use success rate, defined as the percentage of trials that complete a 20 s traversal without termination, as the primary metric of traversability. In simulation, each method is evaluated over 1,000 randomized trials per terrain under a 1.0 m/s forward command.

4.2 Simulation Results

Overall Traversal Performance. Fig. 5 shows that SSR outperforms all baselines. It maintains near-100% success across all training difficulties, even on the hardest stairs, where a 22 cm foot has only a 3 cm support margin. It also traverses a 90 cm gap and a 45 cm platform beyond the curriculum range, demonstrating strong geometric adaptability and extrapolation. By contrast, *HPL* and *PIM* degrade rapidly as difficulty increases, suggesting that sparse or indirect foothold guidance is insufficient for precise foothold learning in these settings. Ablation results show that all SSR components matter, with the largest drops in *NoFoothold* and *NoImgn*, highlighting the importance of reliable foot placement for stable traversal.

Safer Foot Placement via Imagined Foothold Guidance. We evaluate safe foot placement with two metrics: safe foothold rate (SFR), the fraction of footholds with support ratio above 75%, and mean support ratio (MSR), averaged over footholds. The support ratio is one minus support deficiency in Sec. 3.2. As shown in Fig. 6, SSR achieves the highest SFR on all terrains, indicating consistently safer contacts. Even on terrains requiring precise landing such as stairs down and gap, it concentrates footholds within valid support regions. The checkpoint-wise MSR curves further show SSR reaches 75% earliest, reflecting better learning efficiency. In contrast, *NoFoothold* produces more aggressive gaits and the lowest safety. *NoImgn* uses only contact-time assessment, so swing correction is often delayed until the foot nears the ground and contacts remain edge-biased. These

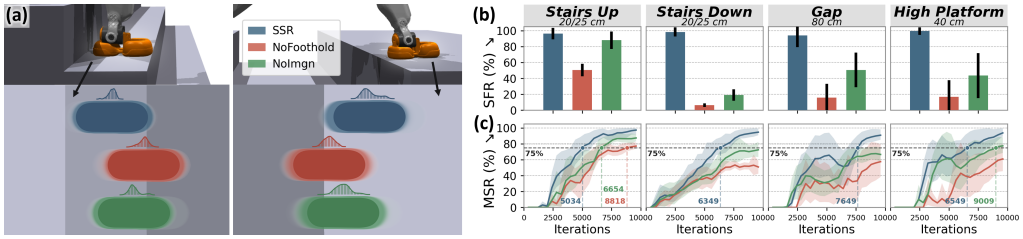

Figure 6: Ablation study of safe foot placement learning. (a) Foothold distributions on stairs down and gap, smoothed with kernel density estimation (KDE). For each terrain, (b) the safe foothold rate (SFR); and (c) the mean support ratio (MSR) versus training iterations, evaluated using checkpoints saved every 100 iterations, with the dashed line marking the first point at which the MSR reaches 75%.

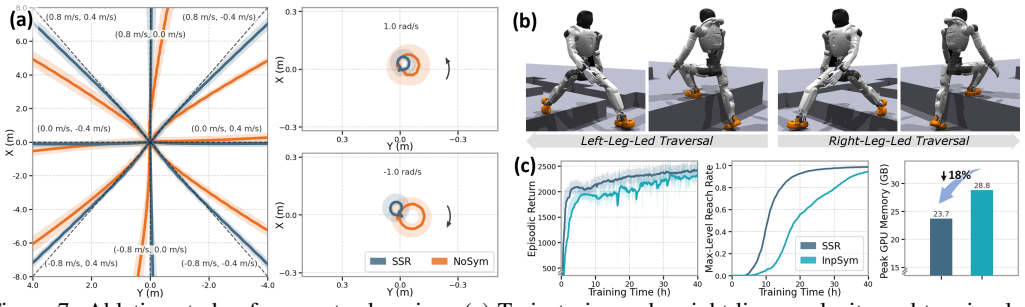

Figure 7: Ablation study of symmetry learning. (a) Trajectories under eight linear-velocity and two in-place yaw-rate commands. (b) SSR shows either-leg-led traversal on gaps and platforms, indicating bilateral balance. (c) Training-time curves of episodic return and maximum-terrain-level reach rate, with peak GPU memory.

results show that guidance from imagined future contacts improves credit assignment for swing-phase decisions, yields consistently supported landings, and accelerates safe traversal learning.

Bilateral Coordination via Efficient Latent-Space Symmetry Augmentation. Fig. 7(a),(b) shows that symmetry augmentation improves controllability and bilateral coordination. With eight-direction velocity commands, SSR tracks more accurately and, under bidirectional yaw-rate commands, stays more localized during in-place turning, with nearly mirror-symmetric trajectories. *NoSym* instead accumulates drift and left-right imbalance, revealing direction-dependent biases and fixed lead-leg dominance. In contrast, SSR supports both left- and right-foot-led traversal, indicating stronger bilateral skill acquisition. Fig. 7(c) further evaluates training efficiency. Compared with *InpSym*, SSR achieves higher returns within the same training time, reaches maximum difficulty earlier, and reduces GPU memory from 28.8 GB to 23.7 GB (18%), making training feasible on one RTX 4090. This gain comes from mirroring a compact latent rather than high-dimensional inputs, avoiding extra encoding passes and hidden-state rollouts. Overall, equivariant latent-space augmentation scales symmetry-driven high-quality locomotion to visual recurrent policy learning.

Human-Like Motion via Terrain-Specific Style Priors. Human-like legged locomotion is associated with low mechanical cost and smooth contacts [51, 52]. We therefore report average power and peak foot contact force as motion-quality proxies. Table 1 shows that SSR achieves the best overall metrics, with more harmonious whole-body motion, including a more natural arm swing and softer impacts, suggesting that the terrain-specific discriminators provide motion priors better aligned with terrain dynamics. Fig. 8 illustrates these natural motions across terrains.

Method	Stairs Up		Stairs Down		Gap		High Platform	
	AP (W)	PFCF (N)	AP (W)	PFCF (N)	AP (W)	PFCF (N)	AP (W)	PFCF (N)
SSR	**381.3**$\pm$5.4	**507.7**$\pm$12.6	**342.4**$\pm$8.6	**519.2**$\pm$19.3	**214.4**$\pm$5.7	**463.4**$\pm$15.3	**253.0**$\pm$6.3	**480.4**$\pm$14.9
NoStyle	451.6$\pm$10.1	538.5$\pm$13.5	371.4$\pm$43.5	555.5$\pm$21.5	317.0$\pm$6.5	518.1$\pm$16.6	292.3$\pm$6.3	508.7$\pm$16.4
SglDisc	391.8$\pm$5.9	544.7$\pm$9.7	345.3$\pm$24.3	549.4$\pm$23.4	241.1$\pm$7.4	481.8$\pm$14.7	258.2$\pm$5.9	505.8$\pm$16.4

Table 1: Ablation study of human-like motion, measured by average power (AP) and peak foot contact force (PFCF). Results are reported as mean$_{\pm\text{std}}$.

4.3 Real-World Results

Lab-Level Evaluation. We evaluate SSR's sim-to-real transfer via zero-shot deployment on the real robot. Table 2 shows high success rates on all terrains, including unseen 90 cm gap and 45 cm platform, close to simulation. To the best of our knowledge, these settings are among the most challenging real-world terrain levels reported for a single traversal policy. Fig. 8 shows consistently

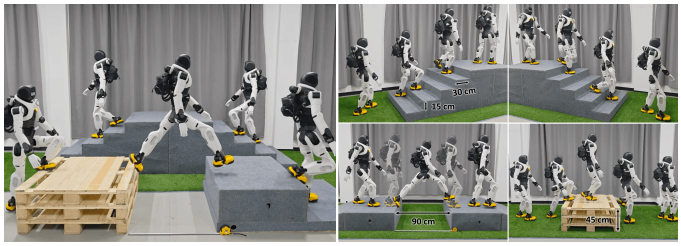

Figure 8: Key frames of zero-shot real-world lab-level deployment. SSR enables stair traversal, 90 cm gap crossing, and 45 cm platform climbing with safe foot placement and natural whole-body motion.

Terrain	Settings	Success Rate
Standard Settings		
Stairs Up	15 / 30 cm	100.0% (20/20)
Stairs Down	15 / 30 cm	100.0% (20/20)
Gap	80 cm	95.0% (19/20)
Platform	40 cm	100.0% (20/20)
Hard Settings		
Gap-H	90 cm*	85.0% (17/20)
Platform-H	45 cm*	95.0% (19/20)

Table 2: Real-world lab-level traversal performance. Success rates are computed over 20 trials per terrain.

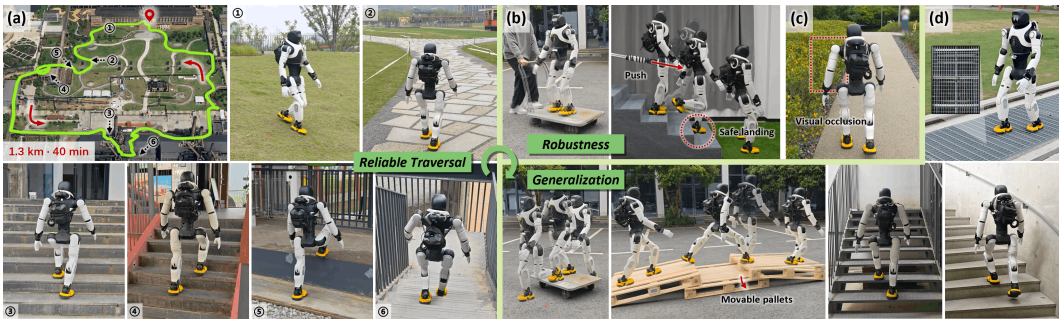

Figure 9: (a) The humanoid completes a 1.3 km, 40 min open-world traversal in an industrial heritage park, reliably crossing stairs of varying sizes, high platforms, rough ground, and grassy slopes. We further test robustness to (b) external disturbances, including wheeled-trolley shaking and pushes on descending stairs, and (c) visual occlusions from tall grass. (d) Zero-shot transfer to OOD terrains, including perforated grid flooring, a slippery trolley, movable forklift pallets, narrow 1.3 m-wide open-riser stairs, and spiral stairs.

precise footholds on flat support regions, coordinated gait, and natural whole-body behavior, jointly improving traversal reliability and motion quality.

Outdoor Deployment. We validate SSR in extensive outdoor environments, as shown in Fig. 1 and Fig. 9. In an outdoor industrial-park route, the robot completes a continuous 1.3 km traversal across diverse terrains without a reset or misstep, maintaining safe and coordinated locomotion over 40 min. Fig. 9(b)–(d) further presents three qualitative stress tests. **Perturbation Recovery**: the robot remains stable under horizontal and torsional shaking; when pushed from behind during stair descent, it still lands on the flat tread, showing timely and accurate foothold decisions under limited response time. **Visual Occlusion Adaptation**: under grass-induced occlusion, it shows no obvious gait degradation and actively steers away from poorly observed regions. **Zero-Shot Out-of-Distribution (OOD) Generalization**: despite no prior exposure to perforated grids, slippery trolleys, or movable pallets during training, the policy traverses them stably. On narrow open-riser stairs and spiral stairs, it handles visual interference from holes and handrails, avoids sidewalls that could trap the feet, and adapts to varying stair widths, showing strong open-world generalization.

Cross-Platform Validation. To evaluate embodiment-level applicability, we instantiate SSR on a full-size DEEP Robotics DR02 humanoid. Despite major differences from AgiBot X2, Appendix E reports high DR02 success rates, suggesting that SSR is not tied to a single humanoid platform.

5 Conclusion

In this paper, we present SSR, a unified humanoid traversal framework that maps egocentric vision to stable, high-quality whole-body locomotion over diverse open-world terrains. Through imagined foothold guidance, SSR models forthcoming foot-terrain contacts to steer foresighted swing-phase foot-placement correction, enabling precise stepping in dynamic, complex settings. Our framework also leverages equivariant latent-space symmetry augmentation to efficiently learn bilateral coordination, while terrain-specific multi-discriminator AMP further encourages consistently natural motion across terrains. Extensive experiments demonstrate that SSR generalizes surefooted, symmetric, and human-like traversal to a broad spectrum of challenging scenarios, supporting reliable long-horizon real-world deployment.

6 Limitations and Future Work

While SSR traverses diverse terrains effectively, several limitations remain. First, its fixed forward-facing depth camera limits perceptual coverage, making omnidirectional traversal challenging. Future work may incorporate multi-view sensing or active viewpoint adjustment. Second, although SSR is robust to visual occlusions and several outdoor OOD terrains, severe depth-sensing failures, such as specular reflections under strong sunlight, can still distort local geometry. Integrating more robust perceptual embeddings may mitigate this issue. Third, SSR mainly focuses on foot-ground interaction, leaving upper-body contacts underexplored. Extending the framework to multi-contact skills and richer human demonstrations could further expand humanoid mobility.

Acknowledgments

We thank Qianshi Wang for assistance with the real-world experiments, and gratefully acknowledge AgiBot and DEEP Robotics for providing hardware support. This work was supported in part by the "Leading Goose" R&D Program of Zhejiang under Grant 2023C01177, the National Key R&D Program of China under Grant 2022YFB4701502, and the 2035 Key Technological Innovation Program of Ningbo City under Grant 2024Z300.

References

[1] I. Radosavovic, S. Kamat, T. Darrell, and J. Malik. Learning humanoid locomotion over challenging terrain. *arXiv preprint arXiv:2410.03654*, 2024.

[2] K. Bonnen, J. S. Matthis, A. Gibaldi, M. S. Banks, D. M. Levi, and M. Hayhoe. Binocular vision and the control of foot placement during walking in natural terrain. *Scientific reports*, 11(1):20881, 2021.

[3] Z. Zhuang, S. Yao, and H. Zhao. Humanoid parkour learning. In *Conference on Robot Learning (CoRL)*, 2024.

[4] S. Zhu, Z. Zhuang, M. Zhao, K.-Y. Lee, and H. Zhao. Hiking in the wild: A scalable perceptive parkour framework for humanoids. *arXiv preprint arXiv:2601.07718*, 2026.

[5] W. Sun, Y. Su, L. Huang, A. Zhang, D. Wei, M. San, D. Tian, E. Cao, F. Yan, E. Xie, et al. Now you see that: Learning end-to-end humanoid locomotion from raw pixels. *arXiv preprint arXiv:2602.06382*, 2026.

[6] J. Sun, G. Han, P. Sun, W. Zhao, J. Cao, J. Wang, Y. Guo, and Q. Zhang. Dpl: Depth-only perceptive humanoid locomotion via realistic depth synthesis and cross-attention terrain reconstruction. *arXiv preprint arXiv:2510.07152*, 2025.

[7] B. Nie, Y. Zhang, R. Jin, Z. Cao, H. Lin, X. Yang, and Y. Gao. Coordinated humanoid robot locomotion with symmetry equivariant reinforcement learning policy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18523–18531, 2026.

[8] M. Mittal, N. Rudin, V. Klemm, A. Allshire, and M. Hutter. Symmetry considerations for learning task symmetric robot policies. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7433–7439. IEEE, 2024.

[9] Z. Su, X. Huang, D. Ordoñez-Apraez, Y. Li, Z. Li, Q. Liao, G. Turrisi, M. Pontil, C. Semini, Y. Wu, et al. Leveraging symmetry in rl-based legged locomotion control. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6899–6906. IEEE, 2024.

[10] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4): 1–20, 2021.

[11] A. Tang, T. Hiraoka, N. Hiraoka, F. Shi, K. Kawaharazuka, K. Kojima, K. Okada, and M. Inaba. Humanmimic: Learning natural locomotion and transitions for humanoid robot via wasserstein adversarial imitation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13107–13114. IEEE, 2024.

[12] Q. Zhang, P. Cui, D. Yan, J. Sun, Y. Duan, G. Han, W. Zhao, W. Zhang, Y. Guo, A. Zhang, et al. Whole-body humanoid robot locomotion with human reference. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11225–11231. IEEE, 2024.

[13] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450. IEEE, 2024.

[14] C. Zhang, W. Xiao, T. He, and G. Shi. Wococo: Learning whole-body humanoid control with sequential contacts. In *Conference on Robot Learning (CoRL)*, 2024.

[15] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter. Advanced skills through multiple adversarial motion priors in reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5120–5126. IEEE, 2023.

[16] H. Wang, Z. Wang, J. Ren, Q. Ben, T. Huang, W. Zhang, and J. Pang. Beamdojo: Learning agile humanoid locomotion on sparse footholds. In *Robotics: Science and Systems (RSS)*, 2025.

[17] Z. Wu, X. Huang, L. Yang, Y. Zhang, K. Sreenath, X. Chen, P. Abbeel, R. Duan, A. Kanazawa, C. Sferrazza, et al. Perceptive humanoid parkour: Chaining dynamic human skills via motion matching. *arXiv preprint arXiv:2602.15827*, 2026.

[18] J. Long, J. Ren, M. Shi, Z. Wang, T. Huang, P. Luo, and J. Pang. Learning humanoid locomotion with perceptive internal model. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9997–10003. IEEE, 2025.

[19] X. Cui, L. Feng, Y. Zhou, H. Han, Z. Liu, and H. Wang. Pilot: A perceptive integrated low-level controller for loco-manipulation over unstructured scenes. *arXiv preprint arXiv:2601.17440*, 2026.

[20] S. Ma, H. Chen, Z. Xu, Y. Zhao, K. Wu, R. Yang, L. Zou, Z. Gan, and W. Ding. Cmoe: Contrastive mixture of experts for motion control and terrain adaptation of humanoid robots. *arXiv preprint arXiv:2603.03067*, 2026.

[21] J. He, C. Zhang, F. Jenelten, R. Grandia, M. Bächer, and M. Hutter. Attention-based map encoding for learning generalized legged locomotion. *Science Robotics*, 10(105):eadv3604, 2025.

[22] D. Wang, X. Wang, X. Liu, J. Shi, Y. Zhao, C. Bai, and X. Li. More: Mixture of residual experts for humanoid lifelike gaits learning on complex terrains. *arXiv preprint arXiv:2506.08840*, 2025.

[23] Z. Zhuang, S. Zhu, M. Zhao, and H. Zhao. Deep whole-body parkour. *arXiv preprint arXiv:2601.07701*, 2026.

[24] Q. Zhang, G. Han, J. Sun, W. Zhao, C. Sun, J. Cao, J. Wang, Y. Guo, and R. Xu. Distillation-ppo: A novel two-stage reinforcement learning framework for humanoid robot perceptive locomotion. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2916–2922. IEEE, 2025.

[25] Y. Liu, T. Yu, H. Song, H. Zhu, N. Hu, Y. Hao, X. Yao, X. Zang, H. Chen, and J. Zhao. Faststair: Learning to run up stairs with humanoid robots. *arXiv preprint arXiv:2601.10365*, 2026.

[26] 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. *arXiv preprint arXiv:2505.11164*, 2025.

[27] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini. Motion planning for quadrupedal locomotion: Coupled planning, terrain mapping, and whole-body control. *IEEE Transactions on Robotics*, 36(6):1635–1648, 2020.

[28] A. Agrawal, S. Chen, A. Rai, and K. Sreenath. Vision-aided dynamic quadrupedal locomotion on discrete terrain using motion libraries. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4708–4714. IEEE, 2022.

[29] S. Fahmi, V. Barasuol, D. Esteban, O. Villarreal, and C. Semini. Vital: Vision-based terrain-aware locomotion for legged robots. *IEEE Transactions on Robotics*, 39(2):885–904, 2022.

[30] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter. Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2):3699–3706, 2020.

[31] F. Jenelten, J. He, F. Farshidian, and M. Hutter. Dtc: Deep tracking control. *Science Robotics*, 9(86):eadh5401, 2024.

[32] W. Yu, D. Jain, A. Escontrela, A. Iscen, P. Xu, E. Coumans, S. Ha, J. Tan, and T. Zhang. Visual-locomotion: Learning to walk on complex terrains with vision. In *Conference on Robot Learning (CoRL)*, 2021.

[33] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis. Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control. *IEEE Transactions on Robotics*, 38(5):2908–2927, 2022.

[34] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. In *Conference on Robot Learning (CoRL)*, 2023.

[35] R. S. Sutton. *Temporal credit assignment in reinforcement learning*. University of Massachusetts Amherst, 1984.

[36] D. Zhu, C. Zhu, Z. Zhang, S. Xin, and Y. Liu. Learning safe locomotion for quadrupedal robots by derived-action optimization. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6870–6876. IEEE, 2024.

[37] C. Zhang, N. Rudin, D. Hoeller, and M. Hutter. Learning agile locomotion on risky terrains. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11864–11871. IEEE, 2024.

[38] F. Abdolhosseini, H. Y. Ling, Z. Xie, X. B. Peng, and M. Van de Panne. On learning symmetric locomotion. In *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pages 1–10, 2019.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[40] I. M. A. Nahrendra, B. Yu, and H. Myung. Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5078–5084. IEEE, 2023.

[41] R. Huang, S. Zhu, Y. Du, and H. Zhao. Moe-loco: Mixture of experts for multitask locomotion. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 14218–14225. IEEE, 2025.

[42] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[43] S. Luo, S. Li, R. Yu, Z. Wang, J. Wu, and Q. Zhu. Pie: Parkour with implicit-explicit learning framework for legged robots. *IEEE Robotics and Automation Letters*, 9(11):9986–9993, 2024.

[44] P. Li, H. Li, Y. Ma, L. Chang, X. Yang, R. Yu, Y. Zhang, Y. Cao, Q. Zhu, and G. Sartoretti. Kivi: Kinesthetic-visuospatial integration for dynamic and safe egocentric legged locomotion. *arXiv preprint arXiv:2509.23650*, 2025.

[45] R. Yu, Q. Wang, H. Li, Z. Jun, Z. Wang, J. Wu, and Q. Zhu. Start: Traversing sparse footholds with terrain reconstruction. *IEEE Robotics and Automation Letters*, 11(2):2194–2201, 2025.

[46] F. Zargarbashi, J. Cheng, D. Kang, R. Sumner, and S. Coros. Robotkeyframing: Learning locomotion with high-level objectives via mixture of dense and sparse rewards. In *Conference on Robot Learning (CoRL)*, 2024.

[47] T. Huang, J. Ren, H. Wang, Z. Wang, Q. Ben, M. Wen, X. Chen, J. Li, and J. Pang. Learning humanoid standing-up control across diverse postures. In *Robotics: Science and Systems (RSS)*, 2025.

[48] G. Cesa, L. Lang, and M. Weiler. A program to build e(n)-equivariant steerable cnns. In *International Conference on Learning Representations (ICLR)*, 2022.

[49] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[50] M. Macklin. Warp: A high-performance python framework for gpu simulation and graphics. In *NVIDIA GPU Technology Conference (GTC)*, volume 3, 2022.

[51] Z. Fu, A. Kumar, J. Malik, and D. Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. In *Conference on Robot Learning (CoRL)*, 2021.

[52] B. R. Whittington and D. G. Thelen. A simple mass-spring model with roller feet can induce the ground reactions observed in human walking. *Journal of biomechanical engineering*, 131 (1):011013, 2009.

[53] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2021.

[54] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[55] W. Xie, J. Han, J. Zheng, H. Li, X. Liu, J. Shi, W. Zhang, C. Bai, and X. Li. Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 62406–62433. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5a0e51901cff2b42d379ec7869603e91-Paper-Conference.pdf.

A Policy Training Details

A.1 Terrain Curriculum

As shown in Fig. 10, we generate five types of terrain for policy learning. Each terrain type contains 20 difficulty levels, with the difficulty progressively increased according to the curriculum strategy in [53]. Each terrain instance is constructed over a rectangular region measuring $8\,\mathrm{m} \times 4\,\mathrm{m}$.

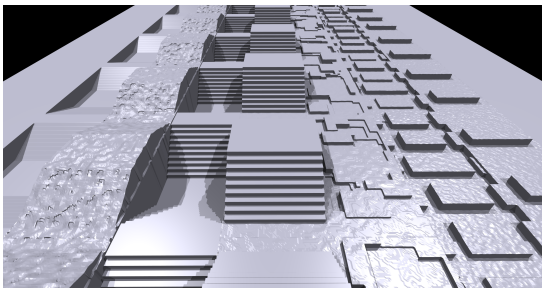

Figure 10: Overview of terrain types used for training.

Terrain Type	Parameter	Range
Slope	Incline	$[0,\ 20]^\circ$
Stairs	Step length	$[0.25,\ 0.4]\,\mathrm{m}$
	Step height	$[0.05,\ 0.2]\,\mathrm{m}$
Discrete	Obstacle height	$[0.05,\ 0.2]\,\mathrm{m}$
Gap	Width	$[0.1,\ 0.8]\,\mathrm{m}$
	Depth	$[0.3,\ 0.6]\,\mathrm{m}$
Platform	Height	$[0.05,\ 0.4]\,\mathrm{m}$

Table 3: Parameter ranges in the terrain curriculum.

A.2 Commands

At timestep t, the locomotion command is defined as $\mathbf{u}_t = [v_x^{\mathrm{cmd}}, v_y^{\mathrm{cmd}}, \omega_z^{\mathrm{cmd}}]^\top$. During training, the robot is commanded to traverse the terrain along a straight path aligned with the x-axis of the world frame while tracking a target heading. The yaw angular velocity command ω_z^{cmd} is computed from the heading error, while the translational commands v_x^{cmd} and v_y^{cmd} are obtained by projecting the prescribed forward velocity from the world frame into the base frame. The terrain-specific ranges of the forward velocity and target heading are summarized in Table 4.

Terrain Type	Forward Velocity (m/s)	Heading (°)
Slope	$\mathcal{U}(-1, 1)$	$\mathcal{U}(-80, 80)$
Stairs, discrete	$\mathcal{U}(0, 1.2)$	$\mathcal{U}(-30, 30)$
Gap, platform	$\mathcal{U}(0, 1.2)$	$\mathcal{U}(-15, 15)$

Table 4: Ranges of the forward velocity and target heading.

To encourage omnidirectional locomotion, we further uniformly sample $\mathbf{u}_t$ in the base frame from a predefined command space after the robot has cleared the highest terrain difficulty and exited the terrain region. This stage exposes the policy to diverse combinations of forward, lateral, and turning motions beyond structured terrain-traversal commands. The corresponding command ranges are listed in Table 5.

Command Term	Range
Forward velocity	$\mathcal{U}(-1, 1)$ m/s
Lateral velocity	$\mathcal{U}(-0.4, 0.4)$ m/s
Yaw angular velocity	$\mathcal{U}(-1.2, 1.2)$ rad/s

Table 5: Ranges of the locomotion commands.

A.3 Rewards

Table 6 summarizes the reward terms used for policy learning, grouped by reward category. For each term, the table provides its mathematical expression and coefficient. It also reports the advantage weight associated with each reward group in the multi-critic formulation. Table 7 defines the main symbols used in these reward expressions.

Term	Expression	Weight
(a) *Locomotion Reward*	r^l	$w_l = 1.0$
Linear Velocity Tracking	$\exp\left(-\lVert\mathbf{v}_{xy} - \mathbf{v}_{xy}^{\text{cmd}}\rVert_2^2/\sigma_v\right)$	1.0
Angular Velocity Tracking	$\exp\left(-(\boldsymbol{\omega}_{\text{yaw}} - \boldsymbol{\omega}_{\text{yaw}}^{\text{cmd}})^2/\sigma_\omega\right)$	0.8
Orientation	$\exp\left(-\lVert\mathbf{g}_{xy}\rVert^2/\sigma_g\right)$	0.5
Angular Velocity (xy)	$\exp\left(-\lVert\boldsymbol{\omega}_{xy}\rVert_2^2/\sigma_\omega\right)$	0.25
Base Height	$\exp\left(-(h_b - h_b^{\text{tar}})^2/\sigma_h\right)$	0.4
Action Rate	$\frac{1}{N}\lVert\mathbf{a}_t - \mathbf{a}_{t-1}\rVert_2^2$	-0.12
Smoothness	$\frac{1}{N}\lVert\mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2}\rVert_2^2$	-0.06
DoF Velocity	$\frac{1}{N}\lVert\dot{\boldsymbol{\theta}}/\dot{\boldsymbol{\theta}}^{\max}\rVert_2^2$	-0.96
DoF Torque	$\frac{1}{N}\lVert\boldsymbol{\tau}/\boldsymbol{\tau}^{\max}\rVert_2^2$	-0.6
DoF Deviation	$\frac{1}{N}\lVert\boldsymbol{\theta} - \boldsymbol{\theta}^{\text{def}}\rVert_1$	-1.8
DoF Position Limits	$\frac{1}{N}\lVert\mathbb{1}(\boldsymbol{\theta} \notin [\boldsymbol{\theta}^{\min}, \boldsymbol{\theta}^{\max}])\rVert_1$	-1.5
DoF Velocity Limits	$\frac{1}{N}\lVert\mathbb{1}(\lvert\dot{\boldsymbol{\theta}}\rvert > \dot{\boldsymbol{\theta}}^{\max})\rVert_1$	-6.0
DoF Torque Limits	$\frac{1}{N}\lVert\mathbb{1}(\lvert\boldsymbol{\tau}\rvert > \boldsymbol{\tau}^{\max})\rVert_1$	-6.0
Stand Still	$\frac{1}{N}\lVert\boldsymbol{\theta} - \boldsymbol{\theta}^{\text{def}}\rVert_1 \cdot \mathbb{1}(\lVert\mathbf{u}_t\rVert_2 \le \epsilon)$	-0.12
Single Support	$\mathbb{1}[\lVert\mathbf{u}_t\rVert_2 \le \epsilon \ \vee \ \exists\, t' \in [t - 0.2\,\text{s},\ t] \text{ s.t. } n_{t'}^{\text{con}} = 1]$	0.2
Impact Velocity	$\sum_{i=1}^{2} {v_{z,i}^f}^2 \cdot c_i$	-1.3
Contact Slippage	$\sum_{i=1}^{2} \lVert\mathbf{v}_{xy,i}^f\rVert_2^2 \cdot c_i$	-0.2
Feet Air Time	$\sum_{i=1}^{2} \max(t_{\text{air},i} - t_{\text{air}}^{\text{tar}}, 0) \cdot c_i^{\text{first}}$	-2.0
Feet Stumble	$\sum_{i=1}^{2} \mathbb{1}(\lVert\mathbf{F}_{xy,i}^f\rVert_2 > 3\lvert F_{z,i}^f\rvert)$	-2.0
Feet Lateral Distance	$\exp\left(\min(d_y - d_y^{\min}, 0)/\sigma_d\right)$	0.08
(b) *Foothold Reward*	r^f	$w_f = 0.25$
Imagined Foothold Guidance[1]	$\exp\left(-(\sum_{i=1}^{2} \tilde{\rho}_{i,t})^2/\sigma_f\right)$	1.0
(c) *Style Reward*	r^s	$w_s = 0.2$
AMP	$\max[0,\ 1 - 0.25(D_i(\boldsymbol{\psi}_t) - 1)^2]$	1.0

[1] On slope terrains, we set $\tilde{\rho}_{i,t} = 0$ for both feet, so that $r_t^f = 1$, avoiding unintended suppression on inclined contacts.

Table 6: Reward terms and weights.

Symbol	Description
$\sigma_v,\ \sigma_\omega,\ \sigma_g,\ \sigma_h,\ \sigma_d,\ \sigma_f$	Gaussian variance scales, set to 0.25, 0.25, 0.01, 0.01, 0.03, and 0.0625.
$\boldsymbol{\theta}^{\min},\ \boldsymbol{\theta}^{\max}$	Lower and upper joint position limits.
$\boldsymbol{\theta}^{\text{def}}$	Default joint positions.
$\dot{\boldsymbol{\theta}}^{\max}$	Maximum allowable joint velocities.
$\boldsymbol{\tau}$	Computed joint torques.
$\boldsymbol{\tau}^{\max}$	Maximum allowable joint torques.
ϵ	Threshold for identifying zero-command, set to 0.15.
$h_b,\ h_b^{\text{tar}}$	Base height and target base height relative to the ground. $h_b^{\text{tar}} = 0.68$.
$t_{\text{air},i},\ t_{\text{air}}^{\text{tar}}$	Air time of foot i and target air time. $t_{\text{air}}^{\text{tar}} = 0.4$.
c_i^{first}	Binary indicator of whether foot i makes its first contact with the ground.
$n_{t'}^{\text{con}}$	Number of feet in contact with the ground at time t'.
$\mathbf{F}_i^f$	Contact force on foot i.
$\mathbf{v}_i^f$	Velocity of foot i.
d_y	Lateral distance between the two feet.
$d_y^{\min}$	Minimum allowable lateral distance between two feet, set to 0.22.

Table 7: Symbols in the reward definitions.

A.4 Network Architectures

Table 8 summarizes the network architectures used in our framework.

Module	Submodule	Component	Structure
Asymmetric Actor-Critic	MoE Actor	Number of experts	5
		Expert MLP hidden size	[1024, 512, 128]
		Gate MLP hidden size	128
		Activation	ELU
	Multi-Critic	Number of critics	3
		MLP size	[512, 256, 128]
		Activation	ELU
Encoder	Proprioception Encoder	MLP hidden size	[512, 256, 128]
		Activation	ELU
		MLP embedding dim	128
	Depth Encoder	CNN Channels	[32, 64, 128]
		CNN kernel sizes	[8, 4, 3]
		CNN strides	[4, 2, 2]
		Activation	ELU
		CNN embedding dim	128
	Temporal Encoder	GRU hidden dim	256
		GRU layers	1
	Fusion Encoder	MLP hidden size	[512, 256, 128]
		Activation	ELU
		MLP embedding dim	48
	Latent Linear Heads	Output dims	$\hat{\mathbf{z}}_t^f : 16,\ \hat{\mathbf{z}}_t^b : 16,\ \hat{\mathbf{z}}_t^p : 16$
Estimator		MLP hidden size	[512, 256, 128]
		Activation	ELU
		Output dims	$\hat{\mathbf{v}}_t : 3$
Foothold Imagination Model		MLP hidden size	[256, 128]
		Activation	ELU
Multi-Discriminator		Number of discriminators	5
		MLP hidden size	[512, 256, 128]
		Activation	ReLU

Table 8: Network architectures.

A.5 Training Hyperparameters

Table 9 summarizes the main hyperparameters used to train the components of our framework. We use four separate Adam optimizers for PPO, the encoder and estimator, the foothold imagination model, and the multi-discriminator, all initialized with a learning rate of 5×10^{-4}.

Component	Parameter	Value
PPO	Value loss coef	1.0
	Entropy coef	0.005
	Desired KL	0.01
	Minimum policy std	0.2
	GAE λ	0.95
	Reward discount γ	0.99
	Number of environments	4096
	Num steps per iteration	24
	Num learning epochs	5
	Num mini-batches	4
Encoder	Base height map MSE coef	2.0
	Foot height maps MSE coef	1.0
	Next proprioception MSE coef	5.0
	VAE KL divergence coef	1.0
Estimator	Velocity MSE coef	2.0
Multi-Discriminator	Weight decay	1×10^{-4}
	Gradient penalty coef w^{gp}	20.0

Table 9: Training hyperparameters.

A.6 Domain Randomization

Table 10 summarizes the domain randomization settings used in our framework, including those applied to the system dynamics and depth sensing.

Category	Parameter	Range / Value
Dynamics	Base added mass	$\mathcal{U}(-2.0, 12.5)\,\mathrm{kg}$
	Base CoM offset	$\mathcal{U}(-0.1, 0.1)\,\mathrm{m}$
	Link mass	$\mathcal{U}(0.9, 1.1) \times$ default
	Link CoM offset	$\mathcal{U}(-0.01, 0.01)\,\mathrm{m}$
	Motor strength	$\mathcal{U}(0.8, 1.2) \times$ default
	PD gains	$\mathcal{U}(0.9, 1.1) \times$ default
	Ground friction	$\mathcal{U}(0.2, 1.25)$
	Action delay	$\mathcal{U}(0, 2)$ control steps
	Push interval	12 s
	Push velocity	$\mathcal{U}(-0.4, 0.4)\,\mathrm{m/s}$
Depth Sensing	Camera position offset	$\mathcal{U}(-0.01, 0.01)\,\mathrm{m}$
	Camera orientation offset	$\mathcal{U}(-1, 1)^\circ$
	Horizontal FoV offset	$\mathcal{U}(-1, 1)^\circ$
	Vertical FoV offset	$\mathcal{U}(-1, 1)^\circ$

Table 10: Domain randomization settings.

A.7 Termination Criteria

An episode is terminated when any of the following conditions is satisfied:

- The episode reaches its maximum duration.
- The robot moves outside the terrain boundary.
- The contact force on any termination-contact link exceeds a predefined threshold.
- The robot exhibits severe orientation instability.
- On gap terrain, any foot drops below a predefined threshold beneath the ground plane.

A.8 Motion Prior Dataset

For each terrain type, the corresponding motion priors are sourced from two datasets: customized motion recordings collected using a motion capture system and a selected subset of the open-source AMASS dataset [54]. All motion sequences are retargeted to our humanoid following Xie et al. [55]. Table 11 summarizes the statistics of the motion datasets used for training.

Terrain Type	Flat-ground motions			Terrain traversals					Total Time (s)
	Forward	Backward	Turn	Slope	Stairs	Step	Gap	Platform	
Slope	✓	✓	✓	✓					261.3
Stairs	✓		✓		✓				295.8
Discrete	✓		✓			✓			136.0
Gap	✓		✓				✓		174.5
Platform	✓		✓			✓		✓	181.6

Table 11: Motion dataset statistics.

B Equivariant Network Details

B.1 Mirror Transformations and Symmetry-Structured Encoder Inputs

This section formalizes the mirror transformations and symmetry operators used in our method. We first define the mirror transformations used for data augmentation in Sec. 3.3. We then describe the structure of the encoder inputs and introduce the associated operators for constructing symmetry-structured representations, which are also used in the proofs in Appendix B.2.

We begin with the mirror transformation $\mathcal{M}_c$ for the latent representation $\hat{\mathbf{z}}_t$, defined as an exchange between the left and right channel groups. More generally, $\mathcal{M}_c$ is the primitive mirror operator for symmetry-structured vectors. For a vector whose channel dimension is partitioned into two paired groups, $\mathcal{M}_c$ swaps the two groups:

$$\mathbf{X} = [\mathbf{X}^{(L)}, \mathbf{X}^{(R)}]^\top \qquad \mathcal{M}_c(\mathbf{X}) = [\mathbf{X}^{(R)}, \mathbf{X}^{(L)}]^\top. \tag{5}$$

We next define the mirror transformation $\mathcal{M}_p$ for proprioception. To specify its action on joint-space variables, we introduce the following block notation. For any joint-space quantity, including joint positions, joint velocities, and actions, we write $\mathbf{x} = \left[\mathbf{x}_R^{\text{leg}},\, \mathbf{x}_L^{\text{leg}},\, x^{\text{waist}},\, \mathbf{x}_R^{\text{arm}},\, \mathbf{x}_L^{\text{arm}}\right]^\top \in \mathbb{R}^{21}$. Each leg block is ordered as *(hip pitch, hip roll, hip yaw, knee, ankle pitch, ankle roll)*, and each arm block is ordered as *(shoulder pitch, shoulder roll, shoulder yaw, elbow)*.

Under reflection about the sagittal plane, the left and right kinematic chains are exchanged, and the sign of each coordinate depends on the corresponding physical axis convention. We therefore define the sign vectors for the joint blocks as $\boldsymbol{s}_{\text{leg}} = [1, -1, -1, 1, 1, -1]^\top$ and $\boldsymbol{s}_{\text{arm}} = [1, -1, -1, 1]^\top$. The resulting joint-block reflection is defined as

$$\mathcal{F}(\mathbf{x}) = \left[\mathbf{x}_L^{\text{leg}} \odot \boldsymbol{s}_{\text{leg}},\, \mathbf{x}_R^{\text{leg}} \odot \boldsymbol{s}_{\text{leg}},\, -x^{\text{waist}},\, \mathbf{x}_L^{\text{arm}} \odot \boldsymbol{s}_{\text{arm}},\, \mathbf{x}_R^{\text{arm}} \odot \boldsymbol{s}_{\text{arm}}\right]^\top, \tag{6}$$

where $\odot$ denotes element-wise multiplication. For the remaining proprioceptive quantities with explicit physical semantics, namely base-frame polar vectors, base angular velocities, and velocity commands, we use the sign vectors $\boldsymbol{s}_y = [1, -1, 1]^\top$, $\boldsymbol{s}_\omega = [-1, 1, -1]^\top$, and $\boldsymbol{s}_u = [1, -1, -1]^\top$.

Table 12 summarizes the mirror transformations used for data augmentation in Sec. 3.3, including those for proprioception, actor observations, privileged states, and actions.

Transformation	Component	Variable	Mirror rule	Dim.
		Proprioception $\mathbf{o}_t^p$		
$\mathcal{M}_p$	Base angular velocity	$\boldsymbol{\omega}_t$	$\boldsymbol{\omega}_t \odot \boldsymbol{s}_\omega$	3
	Projected gravity	$\mathbf{g}_t$	$\mathbf{g}_t \odot \boldsymbol{s}_y$	3
	Velocity command	$\mathbf{u}_t$	$\mathbf{u}_t \odot \boldsymbol{s}_u$	3
	Joint positions	$\boldsymbol{\theta}_t$	$\mathcal{F}(\boldsymbol{\theta}_t)$	21
	Joint velocities	$\dot{\boldsymbol{\theta}}_t$	$\mathcal{F}(\dot{\boldsymbol{\theta}}_t)$	21
	Previous action	$\mathbf{a}_{t-1}$	$\mathcal{F}(\mathbf{a}_{t-1})$	21
	Actor observation $\mathbf{o}_t^a$			
$\mathcal{M}_o$	Proprioception	$\mathbf{o}_t^p$	$\mathcal{M}_p \mathbf{o}_t^p$	72
	Base velocity estimate	$\hat{\mathbf{v}}_t$	$\mathcal{M}_v \hat{\mathbf{v}}_t = \hat{\mathbf{v}}_t \odot \boldsymbol{s}_y$	3
	Encoder latent	$\hat{\mathbf{z}}_t = [\hat{\mathbf{z}}_t^f, \hat{\mathbf{z}}_t^b, \hat{\mathbf{z}}_t^p]^\top$	$[\mathcal{M}_c \hat{\mathbf{z}}_t^f, \mathcal{M}_c \hat{\mathbf{z}}_t^b, \mathcal{M}_c \hat{\mathbf{z}}_t^p]^\top$	48
	Privileged state $\mathbf{s}_t$			
$\mathcal{M}_s$	Proprioception	$\mathbf{o}_t^p$	$\mathcal{M}_p \mathbf{o}_t^p$	72
	Base linear velocity	$\mathbf{v}_t$	$\mathcal{M}_v \mathbf{v}_t = \mathbf{v}_t \odot \boldsymbol{s}_y$	3
	Foot linear velocities	$[\mathbf{v}_{t,R}^f,\, \mathbf{v}_{t,L}^f]^\top$	$[\mathbf{v}_{t,L}^f \odot \boldsymbol{s}_y,\, \mathbf{v}_{t,R}^f \odot \boldsymbol{s}_y]^\top$	6
	Foot contact	$[c_{t,R},\, c_{t,L}]^\top$	$[c_{t,L},\, c_{t,R}]^\top$	2
	Hand positions	$[\mathbf{p}_{t,R}^h,\, \mathbf{p}_{t,L}^h]^\top$	$[\mathbf{p}_{t,L}^h \odot \boldsymbol{s}_y,\, \mathbf{p}_{t,R}^h \odot \boldsymbol{s}_y]^\top$	6
	Foot positions	$[\mathbf{p}_{t,R}^f,\, \mathbf{p}_{t,L}^f]^\top$	$[\mathbf{p}_{t,L}^f \odot \boldsymbol{s}_y,\, \mathbf{p}_{t,R}^f \odot \boldsymbol{s}_y]^\top$	6
	Base height map	$\mathbf{H}_t^b$	$\mathcal{M}_{2\text{D}} \mathbf{H}_t^b$	18×10
	Foot height maps	$(\mathbf{H}_{t,R}^f,\, \mathbf{H}_{t,L}^f)$	$(\mathcal{M}_{2\text{D}} \mathbf{H}_{t,L}^f,\, \mathcal{M}_{2\text{D}} \mathbf{H}_{t,R}^f)$	$2 \times 10 \times 5$
$\mathcal{M}_a$	**Action $\mathbf{a}_t$**			
	Action	$\mathbf{a}_t$	$\mathcal{F}(\mathbf{a}_t)$	21

Table 12: Component-wise mirror rules used in the data augmentation procedure for proprioceptions, latent representations, actor observations, privileged states, and actions.

Next, we analyze the structure of the encoder inputs and define the corresponding mirror operators used in the proofs in Appendix B.2. Because mirror equivariance is implemented through modules with explicitly paired left-right channels, their inputs must share the same symmetry-structured form. In our setting, this form is obtained by organizing the features into paired left and right channels. We describe this construction below for temporal proprioceptive and image inputs.

We first consider the temporal proprioception $\mathbf{o}^p_{t-h+1:t}$. This representation does not explicitly separate the left and right sides. We therefore introduce a fixed reorganization operator $\mathcal{T}$ that maps the length-h proprioceptive history into a symmetry-structured left-right representation:

$$\bar{\mathbf{o}}^p_t = \mathcal{T}\big(\mathbf{o}^p_{t-h+1:t}\big) = [\bar{\mathbf{o}}^p_{t,L}, \bar{\mathbf{o}}^p_{t,R}]^\top, \qquad \bar{\mathbf{o}}^p_{t,L},\, \bar{\mathbf{o}}^p_{t,R} \in \mathbb{R}^{42\times h}. \tag{7}$$

Each branch is constructed by concatenating the corresponding per-frame slices over the past h time steps. The definitions of the per-frame slices $\bar{\mathbf{o}}^p_{t-k,L}$ and $\bar{\mathbf{o}}^p_{t-k,R}$, for $k \in \{0,\dots,h-1\}$, are listed in Table 13. Under this reorganization, the mirror transformation reduces to simple branch exchange:

$$\mathcal{M}_c\big(\bar{\mathbf{o}}^p_t\big) = [\bar{\mathbf{o}}^p_{t,R}, \bar{\mathbf{o}}^p_{t,L}]^\top. \tag{8}$$

Component	Left slice $\bar{\mathbf{o}}^p_{t-k,L}$	Right slice $\bar{\mathbf{o}}^p_{t-k,R}$	Dim.
Base angular velocity	$\boldsymbol{\omega}_{t-k}$	$\boldsymbol{\omega}_{t-k} \odot \boldsymbol{s}_\omega$	3
Gravity vector	$\mathbf{g}_{t-k}$	$\mathbf{g}_{t-k} \odot \boldsymbol{s}_g$	3
Velocity command	$\mathbf{u}_{t-k}$	$\mathbf{u}_{t-k} \odot \boldsymbol{s}_u$	3
Joint angles	$[\boldsymbol{\theta}^{\text{leg}}_{t-k,L},\, \boldsymbol{\theta}^{\text{arm}}_{t-k,L},\, \theta^{\text{waist}}_{t-k}]^\top$	$[\boldsymbol{\theta}^{\text{leg}}_{t-k,R} \odot \boldsymbol{s}_{\text{leg}},\, \boldsymbol{\theta}^{\text{arm}}_{t-k,R} \odot \boldsymbol{s}_{\text{arm}},\, -\theta^{\text{waist}}_{t-k}]^\top$	11
Joint angular velocities	$[\dot{\boldsymbol{\theta}}^{\text{leg}}_{t-k,L},\, \dot{\boldsymbol{\theta}}^{\text{arm}}_{t-k,L},\, \dot{\theta}^{\text{waist}}_{t-k}]^\top$	$[\dot{\boldsymbol{\theta}}^{\text{leg}}_{t-k,R} \odot \boldsymbol{s}_{\text{leg}},\, \dot{\boldsymbol{\theta}}^{\text{arm}}_{t-k,R} \odot \boldsymbol{s}_{\text{arm}},\, -\dot{\theta}^{\text{waist}}_{t-k}]^\top$	11
Previous action	$[\mathbf{a}^{\text{leg}}_{t-k-1,L},\, \mathbf{a}^{\text{arm}}_{t-k-1,L},\, a^{\text{waist}}_{t-k-1}]^\top$	$[\mathbf{a}^{\text{leg}}_{t-k-1,R} \odot \boldsymbol{s}_{\text{leg}},\, \mathbf{a}^{\text{arm}}_{t-k-1,R} \odot \boldsymbol{s}_{\text{arm}},\, -a^{\text{waist}}_{t-k-1}]^\top$	11

Table 13: Per-frame left and right slices used by the reorganization operator $\mathcal{T}$.

We then consider the image input $\mathbf{I}_t$. Images naturally exhibit a left-right spatial structure. We therefore use the horizontal reflection operator $\mathcal{M}_{\text{2D}}$ for two-dimensional vectors. For $\mathbf{Y} \in \mathbb{R}^{H\times W}$, including an image or any image-derived representation, $\mathcal{M}_{\text{2D}}$ acts along the width dimension and performs a horizontal flip:

$$\big[\mathcal{M}_{\text{2D}}(\mathbf{Y})\big]_{i,j} = \mathbf{Y}_{i,\,W+1-j}, \qquad i = 1,\dots,H,\;\; j = 1,\dots,W. \tag{9}$$

Although an image already has a left-right spatial structure, it does not directly provide paired left-right channels along the channel dimension. To address this issue, we introduce a lift convolution layer in the equivariant CNN (Appendix B.2), which lifts the input image to a symmetry-structured feature representation with explicitly paired left and right channels.

B.2 Equivariance Proofs for Network Components

To construct an encoder that satisfies the desired equivariance property, we design each module as an equivariant network. These modules include equivariant MLPs, equivariant GRUs, and an equivariant CNN, which can in turn be decomposed into compositions of equivariant linear and convolutional layers. In this section, we first present the construction and properties of these equivariant network components and provide the corresponding proofs. We then use these results to establish the equivariance of each latent head in the encoder of Sec. 3.3.

Equivariant Linear Layer An equivariant linear layer has the same affine form as a standard linear layer, but its weight matrix and bias are constrained to follow a symmetry-preserving block structure. We define $f : \mathbb{R}^{2C_{\text{in}}} \to \mathbb{R}^{2C_{\text{out}}}$ as

$$f(\mathbf{x}) = W\mathbf{x} + \mathbf{b}, \qquad \mathbf{x} = \begin{bmatrix}\mathbf{x}^{(1)}\\ \mathbf{x}^{(2)}\end{bmatrix}, \qquad W = \begin{bmatrix}A & B\\ B & A\end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix}\mathbf{a}\\ \mathbf{a}\end{bmatrix}. \tag{10}$$

Here, $\mathbf{x} \in \mathbb{R}^{2C_{\text{in}}}$ is the symmetry-structured input feature vector, $A, B \in \mathbb{R}^{C_{\text{out}}\times C_{\text{in}}}$ are learnable weight blocks, and $\mathbf{a} \in \mathbb{R}^{C_{\text{out}}}$ is the shared bias.

Proposition B.1. *The linear layer f is $\mathcal{M}_c$-equivariant:*

$$f(\mathcal{M}_c\mathbf{x}) = \mathcal{M}_c f(\mathbf{x}). \tag{11}$$

Proof. Direct substitution yields

$$f(\mathcal{M}_c\mathbf{x}) = \begin{bmatrix} A\mathbf{x}^{(2)} + B\mathbf{x}^{(1)} + \mathbf{a} \\ B\mathbf{x}^{(2)} + A\mathbf{x}^{(1)} + \mathbf{a} \end{bmatrix} = \mathcal{M}_c \begin{bmatrix} A\mathbf{x}^{(1)} + B\mathbf{x}^{(2)} + \mathbf{a} \\ B\mathbf{x}^{(1)} + A\mathbf{x}^{(2)} + \mathbf{a} \end{bmatrix} = \mathcal{M}_c f(\mathbf{x}).$$

Equivariant MLP The equivariant MLP follows the standard MLP form, except that each linear layer is replaced with an equivariant linear layer f_ℓ:

$$F_{\mathrm{MLP}}(\mathbf{x}) = f_L\Big(\sigma\big(f_{L-1}(\cdots\sigma(f_1(\mathbf{x}))\cdots)\big)\Big), \tag{12}$$

where σ denotes the activation function.

Proposition B.2. *The equivariant MLP F_{MLP} is $\mathcal{M}_c$-equivariant:*

$$F_{\mathrm{MLP}}(\mathcal{M}_c\mathbf{x}) = \mathcal{M}_c F_{\mathrm{MLP}}(\mathbf{x}). \tag{13}$$

Proof. Let

$$G_\ell = \begin{cases} \sigma \circ f_\ell, & \ell = 1, \ldots, L-1, \\ f_L, & \ell = L. \end{cases}$$

By Proposition B.1, each equivariant linear layer f_ℓ commutes with $\mathcal{M}_c$. Since σ is applied element-wise, it also commutes with $\mathcal{M}_c$. Hence every G_ℓ is $\mathcal{M}_c$-equivariant. Because a composition of equivariant maps is equivariant,

$$F_{\mathrm{MLP}} = G_L \circ G_{L-1} \circ \cdots \circ G_1$$

also satisfies

$$F_{\mathrm{MLP}}(\mathcal{M}_c\mathbf{x}) = \mathcal{M}_c F_{\mathrm{MLP}}(\mathbf{x}).$$

Equivariant GRU The equivariant GRU follows the standard update rule, with $\mathbf{h}_t = F_{\mathrm{GRU}}(\mathbf{x}_t, \mathbf{h}_{t-1})$, where all affine transformations use the same block-structured parameterization as the equivariant linear layer:

$$\begin{cases} \mathbf{r}_t = \sigma\left(W_{\mathrm{ir}}\mathbf{x}_t + \mathbf{b}_{\mathrm{ir}} + W_{\mathrm{hr}}\mathbf{h}_{t-1} + \mathbf{b}_{\mathrm{hr}}\right), \\ \mathbf{z}_t = \sigma\left(W_{\mathrm{iz}}\mathbf{x}_t + \mathbf{b}_{\mathrm{iz}} + W_{\mathrm{hz}}\mathbf{h}_{t-1} + \mathbf{b}_{\mathrm{hz}}\right), \\ \mathbf{n}_t = \tanh\left(W_{\mathrm{in}}\mathbf{x}_t + \mathbf{b}_{\mathrm{in}} + \mathbf{r}_t \odot \left(W_{\mathrm{hn}}\mathbf{h}_{t-1} + \mathbf{b}_{\mathrm{hn}}\right)\right), \\ \mathbf{h}_t = (\mathbf{1} - \mathbf{z}_t) \odot \mathbf{n}_t + \mathbf{z}_t \odot \mathbf{h}_{t-1}. \end{cases} \tag{14}$$

Here, $\mathbf{x}_t \in \mathbb{R}^{2C_{\mathrm{in}}}$ denotes the input feature at time t; $\mathbf{h}_{t-1}$ and $\mathbf{h}_t$ are the previous and updated hidden states in $\mathbb{R}^{2C_{\mathrm{h}}}$, and $\mathbf{r}_t$, $\mathbf{z}_t$, and $\mathbf{n}_t$ denote the reset gate, update gate, and candidate hidden state, respectively.

Proposition B.3. *The one-step GRU update is equivariant under channel exchange:*

$$F_{\mathrm{GRU}}\big(\mathcal{M}_c\mathbf{x}_t, \mathcal{M}_c\mathbf{h}_{t-1}\big) = \mathcal{M}_c F_{\mathrm{GRU}}(\mathbf{x}_t, \mathbf{h}_{t-1}). \tag{15}$$

Proof. By Proposition B.1, all affine maps in the GRU update commute with $\mathcal{M}_c$. Since σ, $\tanh$, subtraction from $\mathbf{1}$, and the Hadamard product are all pointwise operations, they also commute with $\mathcal{M}_c$. Therefore

$$\mathbf{r}'_t = \mathcal{M}_c\mathbf{r}_t, \qquad \mathbf{z}'_t = \mathcal{M}_c\mathbf{z}_t.$$

For the candidate hidden state,

$$\begin{aligned} \mathbf{n}'_t &= \tanh\Big(W_{\mathrm{in}}\mathcal{M}_c\mathbf{x}_t + \mathbf{b}_{\mathrm{in}} + \mathbf{r}'_t \odot \big(W_{\mathrm{hn}}\mathcal{M}_c\mathbf{h}_{t-1} + \mathbf{b}_{\mathrm{hn}}\big)\Big) \\ &= \tanh\Big(\mathcal{M}_c\big(W_{\mathrm{in}}\mathbf{x}_t + \mathbf{b}_{\mathrm{in}} + \mathbf{r}_t \odot (W_{\mathrm{hn}}\mathbf{h}_{t-1} + \mathbf{b}_{\mathrm{hn}})\big)\Big) = \mathcal{M}_c\mathbf{n}_t. \end{aligned}$$

Hence

$$\begin{aligned} F_{\mathrm{GRU}}(\mathcal{M}_c\mathbf{x}_t, \mathcal{M}_c\mathbf{h}_{t-1}) &= (\mathbf{1} - \mathbf{z}'_t) \odot \mathbf{n}'_t + \mathbf{z}'_t \odot \mathcal{M}_c\mathbf{h}_{t-1} \\ &= \mathcal{M}_c\big((\mathbf{1} - \mathbf{z}_t) \odot \mathbf{n}_t + \mathbf{z}_t \odot \mathbf{h}_{t-1}\big) = \mathcal{M}_c F_{\mathrm{GRU}}(\mathbf{x}_t, \mathbf{h}_{t-1}). \end{aligned}$$

Equivariant Convolution Layer We define the equivariant convolutional layers using cross-correlation and symmetry-constrained kernels. For an input feature map $\mathbf{x} \in \mathbb{R}^{C_{\text{in}} \times H \times W}$, let $\bar{\mathbf{x}} \in \mathbb{R}^{C_{\text{in}} \times \bar{H} \times \bar{W}}$ denote its padded version, where $\bar{H} = H + 2p_H$ and $\bar{W} = W + 2p_W$. Given a convolution kernel $\mathbf{k} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times K_H \times K_W}$, we define cross-correlation with stride s as

$$(\mathbf{x} \star \mathbf{k})_{o,h,w} = \sum_{i=1}^{C_{\text{in}}} \sum_{u=1}^{K_H} \sum_{v=1}^{K_W} \mathbf{k}_{o,i,u,v}\, \bar{\mathbf{x}}_{i,(h-1)s+u,(w-1)s+v}. \tag{16}$$

Here, $h = 1, \ldots, H_o$ and $w = 1, \ldots, W_o$, where $H_o = \lfloor (\bar{H} - K_H)/s \rfloor + 1$ and $W_o = \lfloor (\bar{W} - K_W)/s \rfloor + 1$. When $s > 1$, we assume $\bar{W} - K_W$ is divisible by s so that horizontal sampling grid remains aligned under flipping. To describe mirror equivariance in the spatial domain, we define the horizontal flip operator on padded feature maps and kernels as

$$[\mathcal{M}_{2\text{D}} \bar{\mathbf{x}}]_{i,h,w} = \bar{\mathbf{x}}_{i,h,\bar{W}+1-w}, \qquad [\mathcal{M}_{2\text{D}} \mathbf{k}]_{o,i,u,v} = \mathbf{k}_{o,i,u,K_W+1-v}. \tag{17}$$

An ordinary image does not explicitly contain paired symmetry channels. We therefore use a lift convolution to map it to a symmetry-structured representation:

$$f_{\text{l}}(\mathbf{x}) = \mathbf{x} \star K_{\text{l}} + \mathbf{b}_{\text{l}}, \qquad K_{\text{l}} = \begin{bmatrix} \mathbf{k}_{\text{l}} \\ \mathcal{M}_{2\text{D}} \mathbf{k}_{\text{l}} \end{bmatrix}, \qquad \mathbf{b}_{\text{l}} = \begin{bmatrix} \mathbf{a}_{\text{l}} \\ \mathbf{a}_{\text{l}} \end{bmatrix}. \tag{18}$$

Here, $\mathbf{k}_{\text{l}} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times K_H \times K_W}$ is a learnable kernel, and $\mathbf{a}_{\text{l}} \in \mathbb{R}^{C_{\text{out}}}$ is a shared bias. After lifting, the subsequent layers preserve the symmetry-structured representation through constrained kernels. Each non-lift convolution is defined as

$$f_{\text{nl}}(\mathbf{x}) = \mathbf{x} \star K_{\text{nl}} + \mathbf{b}_{\text{nl}}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{bmatrix}, \quad K_{\text{nl}} = \begin{bmatrix} \mathbf{k}_{\text{a}} & \mathbf{k}_{\text{b}} \\ \mathcal{M}_{2\text{D}} \mathbf{k}_{\text{b}} & \mathcal{M}_{2\text{D}} \mathbf{k}_{\text{a}} \end{bmatrix}, \quad \mathbf{b}_{\text{nl}} = \begin{bmatrix} \mathbf{a}_{\text{nl}} \\ \mathbf{a}_{\text{nl}} \end{bmatrix}. \tag{19}$$

Here, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \mathbb{R}^{C_{\text{in}} \times H \times W}$ are the two symmetry branches, $\mathbf{k}_{\text{a}}, \mathbf{k}_{\text{b}} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times K_H \times K_W}$ are learnable kernels, and $\mathbf{a}_{\text{nl}} \in \mathbb{R}^{C_{\text{out}}}$ is a shared bias.

Lemma B.4. *Cross-correlation satisfies*

$$(\mathcal{M}_{2\text{D}} \mathbf{x}) \star \mathbf{k} = \mathcal{M}_{2\text{D}} \big(\mathbf{x} \star (\mathcal{M}_{2\text{D}} \mathbf{k}) \big). \tag{20}$$

Proof. For the left-hand side,

$$[(\mathcal{M}_{2\text{D}} \mathbf{x}) \star \mathbf{k}]_{o,h,w} = \sum_{i,u,v} \mathbf{k}_{o,i,u,v}\, \bar{\mathbf{x}}_{i,(h-1)s+u,\bar{W}+1-(w-1)s-v}.$$

For the right-hand side, substituting $v' = K_W + 1 - v$ gives

$$[\mathcal{M}_{2\text{D}}(\mathbf{x} \star (\mathcal{M}_{2\text{D}} \mathbf{k}))]_{o,h,w} = \sum_{i,u,v'} \mathbf{k}_{o,i,u,v'}\, \bar{\mathbf{x}}_{i,(h-1)s+u,(W_o-w)s+K_W+1-v'}.$$

Using $W_o = \frac{\bar{W} - K_W}{s} + 1$ and the alignment assumption, the last index equals

$$\bar{W} + 1 - (w-1)s - v',$$

so the two expressions coincide.

Proposition B.5. *The lift convolution satisfies*

$$f_{\text{l}}(\mathcal{M}_{2\text{D}} \mathbf{x}) = \mathcal{M}_c \mathcal{M}_{2\text{D}} (f_{\text{l}}(\mathbf{x})). \tag{21}$$

Proof. By Lemma B.4,

$$(\mathcal{M}_{2\text{D}} \mathbf{x}) \star \mathbf{k}_{\text{l}} = \mathcal{M}_{2\text{D}} \big(\mathbf{x} \star (\mathcal{M}_{2\text{D}} \mathbf{k}_{\text{l}}) \big), \qquad (\mathcal{M}_{2\text{D}} \mathbf{x}) \star (\mathcal{M}_{2\text{D}} \mathbf{k}_{\text{l}}) = \mathcal{M}_{2\text{D}} (\mathbf{x} \star \mathbf{k}_{\text{l}}).$$

Hence

$$f_{\text{l}}(\mathcal{M}_{2\text{D}} \mathbf{x}) = \begin{bmatrix} \mathcal{M}_{2\text{D}} \big(\mathbf{x} \star (\mathcal{M}_{2\text{D}} \mathbf{k}_{\text{l}}) + \mathbf{a}_{\text{l}} \big) \\ \mathcal{M}_{2\text{D}} \big(\mathbf{x} \star \mathbf{k}_{\text{l}} + \mathbf{a}_{\text{l}} \big) \end{bmatrix} = \mathcal{M}_c \mathcal{M}_{2\text{D}} f_{\text{l}}(\mathbf{x}).$$

Proposition B.6. *The non-lift convolution satisfies*

$$f_{\mathrm{nl}}\big(\mathcal{M}_c\mathcal{M}_{2\mathrm{D}}\mathbf{x}\big) = \mathcal{M}_c\mathcal{M}_{2\mathrm{D}} f_{\mathrm{nl}}(\mathbf{x}). \tag{22}$$

Proof. Let

$$y_1 = \mathbf{x}^{(1)} \star \mathbf{k}_{\mathrm{a}} + \mathbf{x}^{(2)} \star \mathbf{k}_{\mathrm{b}} + \mathbf{a}_{\mathrm{nl}}, \qquad y_2 = \mathbf{x}^{(1)} \star (\mathcal{M}_{2\mathrm{D}}\mathbf{k}_{\mathrm{b}}) + \mathbf{x}^{(2)} \star (\mathcal{M}_{2\mathrm{D}}\mathbf{k}_{\mathrm{a}}) + \mathbf{a}_{\mathrm{nl}}.$$

Then

$$f_{\mathrm{nl}}(\mathbf{x}) = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

Using Lemma B.4,

$$f_{\mathrm{nl}}(\mathcal{M}_c\mathcal{M}_{2\mathrm{D}}\mathbf{x}) = \begin{bmatrix} \mathcal{M}_{2\mathrm{D}} y_2 \\ \mathcal{M}_{2\mathrm{D}} y_1 \end{bmatrix} = \mathcal{M}_c\mathcal{M}_{2\mathrm{D}} f_{\mathrm{nl}}(\mathbf{x}).$$

Equivariant CNN The equivariant CNN is constructed by composing the layer-wise equivariant convolutions defined above. It consists of one lift convolution f_{l} followed by $N-1$ non-lift convolutions f_{nl}^i:

$$F_{\mathrm{CNN}}(\mathbf{x}) = f_{\mathrm{nl}}^{N-1}\Big(\sigma\big(f_{\mathrm{nl}}^{N-2}(\cdots\sigma(f_{\mathrm{nl}}^1(\sigma(f_{\mathrm{l}}(\mathbf{x}))))\cdots)\big)\Big). \tag{23}$$

Let $F_{\mathrm{CNN}}(\mathbf{x}) \in \mathbb{R}^{2C_N \times H_N \times W_N}$ denote the output feature map of the equivariant CNN, where H_N and W_N are its spatial height and width.

Proposition B.7. *The equivariant CNN satisfies*

$$F_{\mathrm{CNN}}(\mathcal{M}_{2\mathrm{D}}\mathbf{x}) = \mathcal{M}_c\mathcal{M}_{2\mathrm{D}} F_{\mathrm{CNN}}(\mathbf{x}). \tag{24}$$

Proof. Let

$$T := \mathcal{M}_c\mathcal{M}_{2\mathrm{D}}.$$

By Proposition B.5, the lift convolution is T-equivariant. By Proposition B.6, each non-lift convolution is also T-equivariant. Since σ is applied element-wise, it commutes with both channel exchange and horizontal flipping, and hence with T. Therefore every layer in the composition defining F_{CNN} is T-equivariant. By closure under composition,

$$F_{\mathrm{CNN}}(\mathcal{M}_{2\mathrm{D}}\mathbf{x}) = T F_{\mathrm{CNN}}(\mathbf{x}) = \mathcal{M}_c\mathcal{M}_{2\mathrm{D}} F_{\mathrm{CNN}}(\mathbf{x}).$$

Corollary B.8. *If the output feature map has spatial resolution 1×1, then*

$$F_{\mathrm{CNN}}(\mathcal{M}_{2\mathrm{D}}\mathbf{x}) = \mathcal{M}_c F_{\mathrm{CNN}}(\mathbf{x}). \tag{25}$$

Theorem B.9. *Let $E_\phi^{(k)}, k \in \{f, b, p\}$ denote the mapping from the symmetry-structured proprioception $\bar{\mathbf{o}}_t^p$ and the image $\mathbf{I}_t$ to the k-th latent head $\hat{\mathbf{z}}_t^k$ in the encoder used in Sec. 3.3. Then each latent head is equivariant under the joint action of channel exchange on proprioception and horizontal reflection on images:*

$$E_\phi^{(k)}(\mathcal{M}_c\bar{\mathbf{o}}_t^p, \mathcal{M}_{2\mathrm{D}}\mathbf{I}_t) = \mathcal{M}_c E_\phi^{(k)}(\bar{\mathbf{o}}_t^p, \mathbf{I}_t), \tag{26}$$

where $\mathcal{M}_c$ denotes the mirror operator acting on the k-th latent head.

Proof. The encoder in Sec. 3.3 consists of an equivariant proprioceptive MLP, an equivariant visual CNN, a GRU applied to their concatenated features, a fusion MLP, and the k-th latent head.

First, the proprioceptive branch is implemented by equivariant MLPs. By Proposition B.2, each such MLP is $\mathcal{M}_c$-equivariant. Hence the proprioceptive feature transforms equivariantly under channel exchange.

Second, the visual branch is implemented by an equivariant CNN. By Proposition B.7, the CNN satisfies

$$F_{\mathrm{CNN}}(\mathcal{M}_{2\mathrm{D}}\mathbf{x}) = \mathcal{M}_c\mathcal{M}_{2\mathrm{D}}F_{\mathrm{CNN}}(\mathbf{x}).$$

For the architecture used in this work, the output spatial resolution is 1×1. Hence, by Corollary B.8, the visual feature produced by the CNN satisfies

$$F_{\mathrm{CNN}}(\mathcal{M}_{2\mathrm{D}}\mathbf{I}_t) = \mathcal{M}_c F_{\mathrm{CNN}}(\mathbf{I}_t).$$

Next, let $\oplus$ denote feature concatenation along the symmetry-structured channel dimension. Since both proprioceptive and visual features transform under the same channel-exchange operator, concatenation preserves equivariance:

$$(\mathcal{M}_c\mathbf{u}) \oplus (\mathcal{M}_c\mathbf{v}) = \mathcal{M}_c(\mathbf{u} \oplus \mathbf{v}).$$

The concatenated feature is then processed by the GRU. By Proposition B.3, the GRU is $\mathcal{M}_c$-equivariant. Its output is further mapped by the fusion MLP, which is also $\mathcal{M}_c$-equivariant by Proposition B.2.

Finally, the fused feature is mapped to the k-th latent head by an equivariant head mapping. By Proposition B.2, this mapping is equivariant with respect to $\mathcal{M}_c$ on both its input and output. Combining the equivariance of the proprioceptive MLP, the visual CNN, the feature concatenation, the GRU, the fusion MLP, and the final head yields

$$E_\phi^{(k)}(\mathcal{M}_c\bar{\mathbf{o}}_t^p, \mathcal{M}_{2\mathrm{D}}\mathbf{I}_t) = \mathcal{M}_c E_\phi^{(k)}(\bar{\mathbf{o}}_t^p, \mathbf{I}_t).$$

This completes the proof.

C Efficient Depth Rendering with Self-Occlusion

To synthesize depth observations consistent with deployment-time sensing, we ray-cast against both the static terrain mesh and the robot's dynamic link meshes. A naive implementation would update dynamic mesh poses and repeatedly refit the corresponding ray-query acceleration structures, such as bounding volume hierarchies (BVHs), at every simulation step, which quickly becomes a bottleneck in massively parallel training. To avoid this overhead, we keep each mesh in its local frame and instead transform each camera ray from the world frame into the queried mesh frame before ray-mesh intersection. Because these transformations are rigid, hit distances remain directly comparable across meshes. This yields an efficient depth renderer with self-occlusion, implemented in NVIDIA Warp [50], for large-scale GPU-parallel simulation. The procedure is summarized in Algorithm 1.

Algorithm 1: Efficient Depth Rendering with Self-Occlusion

Require: terrain mesh $\mathcal{M}^{\mathrm{ter}}$ with pose $(\mathbf{p}^{\mathrm{ter}}, \mathbf{R}^{\mathrm{ter}})$; per-link meshes $\{\mathcal{M}_j^\ell\}_{j=1}^{N^\ell}$ with per-env link poses $\{(\mathbf{p}_{j,e}^\ell, \mathbf{R}_{j,e}^\ell)\}$; per-env camera poses $\{(\mathbf{p}_e^{\mathrm{cam}}, \mathbf{R}_e^{\mathrm{cam}})\}_{e=1}^{N^{\mathrm{env}}}$; half-FoV tangents $(\tan_{x,e}, \tan_{y,e})$; depth bounds $(d_{\min}, d_{\max})$.
Ensure: clipped camera-depth maps $\mathbf{D} \in \mathbb{R}^{N^{\mathrm{env}} \times H \times W}$.

▷ *Parallel mesh-based occluder rendering with one thread per pixel*

for each pixel $(e, h, w) \in [N^{\mathrm{env}}] \times [H] \times [W]$ **in parallel do**
  Image-plane ray coordinates: $t_x \leftarrow \big(2(w - \frac{1}{2})/W - 1\big)\tan_{x,e}$, $t_y \leftarrow \big(2(h - \frac{1}{2})/H - 1\big)\tan_{y,e}$
  World frame ray origin: $\mathbf{o} \leftarrow \mathbf{p}_e^{\mathrm{cam}}$
  World frame ray direction: $\mathbf{d} \leftarrow \mathrm{Normalize}\big(\mathrm{Rot}(\mathbf{R}_e^{\mathrm{cam}}, [t_x, t_y, -1]^\top)\big)$
  Terrain frame ray: $(\mathbf{o}^{\mathrm{ter}}, \mathbf{d}^{\mathrm{ter}}) \leftarrow \mathrm{Transform}^{-1}\big((\mathbf{p}^{\mathrm{ter}}, \mathbf{R}^{\mathrm{ter}}), (\mathbf{o}, \mathbf{d})\big)$
  Terrain hit distance: $t^{\mathrm{ter}} \leftarrow \mathrm{RayQuery}(\mathcal{M}^{\mathrm{ter}}, \mathbf{o}^{\mathrm{ter}}, \mathbf{d}^{\mathrm{ter}})$
  Initialize nearest self-occlusion distance: $t^{\mathrm{occ}} \leftarrow +\infty$
  for $j = 1$ **to** N^ℓ **do**
    Link frame ray: $(\mathbf{o}_j^\ell, \mathbf{d}_j^\ell) \leftarrow \mathrm{Transform}^{-1}\big((\mathbf{p}_{j,e}^\ell, \mathbf{R}_{j,e}^\ell), (\mathbf{o}, \mathbf{d})\big)$
    Link hit distance: $t_j^\ell \leftarrow \mathrm{RayQuery}(\mathcal{M}_j^\ell, \mathbf{o}_j^\ell, \mathbf{d}_j^\ell)$
    $t^{\mathrm{occ}} \leftarrow \min(t^{\mathrm{occ}}, t_j^\ell)$
    if $t^{\mathrm{occ}} \le d_{\min}$ **then break**
  end for
  Convert ray distance to camera z-depth: $c \leftarrow (1 + t_x^2 + t_y^2)^{-1/2}$
  $\mathbf{D}[e, h, w] \leftarrow \mathrm{clamp}\big(c \cdot \min(t^{\mathrm{ter}}, t^{\mathrm{occ}}), d_{\min}, d_{\max}\big)$
end for
return $\mathbf{D}$

D Real-World Deployment Details

We deploy our policy on the AgiBot X2 humanoid robot, which has 29 joints, stands 1.31 m tall, and weighs approximately 39 kg. A forward-facing Intel RealSense D435i depth camera is mounted at the waist and pitched downward by 50° relative to the horizontal plane. The camera provides horizontal and vertical fields of view of 87° and 58°, respectively. Raw 640×360 depth images are filtered, downsampled, and hole-filled using the Intel RealSense SDK, and then cropped to a resolution of 36×36 before being fed into the policy. Depth observations are streamed to the controller at 60 Hz.

Joint Name	K_p	K_d	Action Scale
Hip pitch	120	4	0.25
Hip yaw	100	2	0.25
Hip roll	100	2	0.25
Knee	120	4	0.25
Ankle pitch	40	2.0	0.25
Ankle roll	20	1.0	0.25
Waist yaw	100	3	0.2
Shoulder pitch	30	1	0.2
Shoulder roll	30	1	0.2
Shoulder yaw	30	1	0.2
Elbow	30	1	0.2

Table 14: Joint-wise deployment gains and action scales.

For onboard execution, we establish a 1 kHz communication pipeline between the motion control unit (RK3588) and the inference unit (Jetson AGX Orin) via lightweight ROS 2 topic-based messaging. Policy inference runs with ONNX Runtime at 50 Hz and outputs target joint positions for all 21 policy-controlled joints. During deployment, each actuated joint is assigned an individual proportional gain K_p, derivative gain K_d, and action scale, as summarized in Table 14.

E Cross-Platform Validation

To evaluate the generalizability of SSR across different robotic platforms, we conduct a cross-platform validation on the full-size DEEP Robotics DR02 humanoid. We retain the same training and deployment pipeline used for the primary platform, AgiBot X2, without introducing any platform-specific algorithmic changes. Compared with AgiBot X2, DR02 differs substantially in physical scale and dynamics, standing at roughly 1.8 m tall and weighing around 70 kg. Despite these embodiment differences, SSR can be instantiated effectively on the new platform. As reported in Table 15 and illustrated in Fig. 11, the policy achieves high success rates across representative lab-level terrains, while maintaining precise foothold placement and coordinated, natural whole-body motion. These results indicate that SSR is not tightly coupled to a specific hardware configuration and exhibits promising cross-platform applicability across distinct humanoid embodiments.

Terrain	Settings	Success Rate
Stairs Up	15 / 40 cm	100.0% (20/20)
Stairs Down	15 / 40 cm	100.0% (20/20)
Gap	80 cm	100.0% (20/20)
Platform	45 cm	95.0% (19/20)

Table 15: Cross-platform real-world lab-level traversal performance on DEEP Robotics DR02. Success rates are computed over 20 trials per terrain.

Figure 11: Key frames of cross-platform laboratory deployment on the full-size DEEP Robotics DR02 humanoid. The SSR policy successfully traverses stairs, an 80 cm gap, and a 45 cm platform, demonstrating its cross-platform generalization.