Let Humanoids Hike! Integrative Skill Development on Complex Trails

Kwan-Yee Lin Stella X. Yu
University of Michigan, Ann Arbor
{junyilin, stellayu}@umich.edu

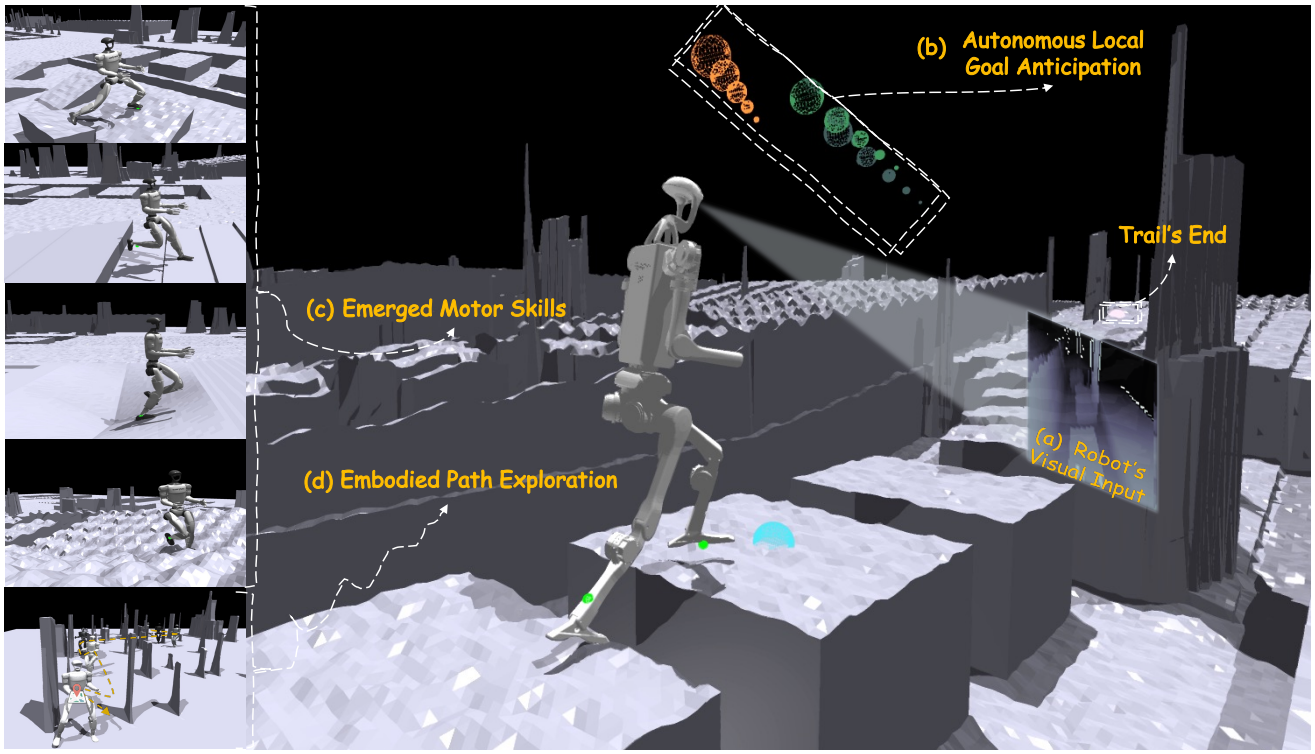

Figure 1. **LEGO-H enables humanoids to hike complex trails independently.** The center H1 robot autonomously adapts to terrain gaps, using near-future navigation goals (b) to guide movement toward the trail end. Larger to smaller bubbles indicate navigation direction, with colors showing future step progression (orange → green → forest). LEGO-H's end-to-end framework integrates visual perception (a) and body dynamics for seamless navigation and locomotion. Left figures show emergent motor skills (c) and path exploration over obstacles (d) in a smaller G1 robot. Project page: LEGO-H-HumanoidRobotHiking.github.io.

Abstract

Hiking on complex trails demands balance, agility, and adaptive decision-making over unpredictable terrain. Current humanoid research remains fragmented and inadequate for hiking: locomotion focuses on motor skills without long-term goals or situational awareness, while semantic navigation overlooks real-world embodiment and local terrain variability. We propose training humanoids to hike on complex trails, fostering integrative skill development across visual perception, decision making, and motor execution.

We develop LEGO-H, a learning framework that enables a humanoid with vision to hike complex trails independently. It has two key innovations. 1) A Temporal Vision Transformer variant anticipates future steps to guide locomotion, unifying local movement and goal-directed navigation. 2) Latent representations of joint movement patterns combined with hierarchical metric learning allow smooth policy transfer from privileged training to onboard execution. These techniques enable LEGO-H to handle diverse physical and environmental challenges without relying on predefined motion patterns. Experiments on diverse simulated hiking trails and humanoids with different morphologies demonstrate LEGO-H's robustness and versatility, establishing a strong foundation for future humanoid development.

1. Introduction

Hiking [27, 29]challenges humans to master diverse motor skills and adapt to complex, unpredictable terrains - such as steep slopes, wide ditches, tangled roots, and abrupt elevation changes *etc.*, – requiring constant balance, agility, and decision-making. Hiking is thus an ideal testbed for de-

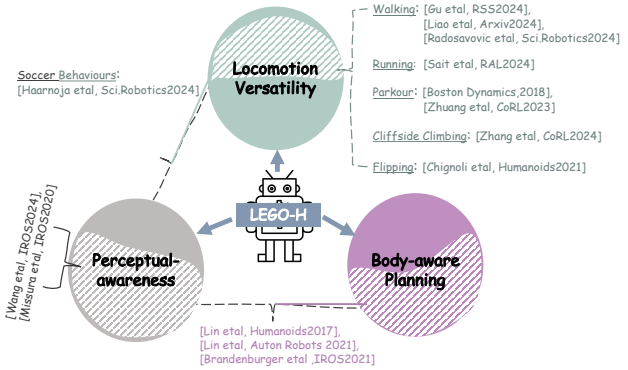

Figure 2. **Hiking requires locomotion versatility, perceptual awareness, and body-aware planning - integrated for the first time in our approach.** Prior work considers only subsets of these capabilities (hatched patterns), whereas LEGO-H unifies all three within a single learning framework to enable embodied autonomy.

veloping humanoid autonomy and coordination between vision, decision-making, and motor execution. Hiking-capable robots could explore remote areas, assist in rescue missions, and guide individuals along rugged paths.

Hiking poses challenges beyond traditional goal navigation, blind locomotion, and single motor pattern learning. To succeed, humanoid robots must excel in three key areas. **1)** Locomotion versatility – The ability to handle mixed terrains like dirt, rocks, stairs, and streams, adapting dynamically with skills like jumping and leaping while maintaining balance. **2)** Perceptual awareness - The ability to sense and respond to complex 3D environments, such as stepping over logs or navigating around trees. **3)** Body awareness – The ability to adjust in real time to local obstacles, terrain changes, and body states by coordinating vision and motor control for adaptive foot placement and movement.

Current humanoids struggle to meet these demands due to the lack of a unified framework that integrates low-level motor skills with high-level navigation (Fig 2). **1)** Locomotion methods simplify terrain interactions to static patterns, focusing narrowly on walking [32, 33] or mimicry [20, 30], limiting real-world adaptability. Advanced frameworks for complex skills like parkour [9] often rely heavily on user commands or engineering. **2)** Navigation methods struggle with real-time adaptability, relying on scene mapping [28] or rigid assumptions [26]. While LLMs/VLMs [44] have advanced, their lack of motor skill integration limits perceptual awareness and last-step feasibility. Bridging motor skills and navigation remains challenging due to asynchronous and divergent responses required for complex environments.

We introduce LEGO-H, a perceptual-aware, end-to-end, embodied learning framework that enables humanoid robots to traverse complex trails using visual inputs (Fig 1). LEGO-H fosters integrative navigation and locomotion skills by refining the Hierarchical Reinforcement Learning (HRL) paradigm and improving the privileged learning scheme.

To achieve perceptual awareness and embodiment at both planning and motor skill levels, a structured framework is essential to manage the interplay between navigation and locomotion. The HRL paradigm suits this purpose, but learning multi-level policies within a single framework often compromises one aspect (*e.g.,*[1] limits motor skills to walking, and [11] oversimplifies the environment). ***Our first contribution is reformulating high-level local navigation as a vision-based sequential anticipation problem to guide locomotion policy learning.*** We introduce TC-ViTs, a Temporal Vision Transformer variant tailored for HRL, combining vision transformer's tokenization with RL's embodiment. Instead of treating the navigation target as a static token, TC-ViTs model **1)** navigation goals and **2)** temporal-spatial relations, considering the robot's past, present, and future states for sequential anticipation. The locomotion policy network then integrates these latent features, and partial anticipation, with proprioceptive input to generate motor actions. This architecture ensures coordination between perception and motor execution, essential for navigating complex dynamic trails.

A key challenge in robot learning is to develop diverse *and* safe motor skills. Privileged learning offers a solution: Assume the intermediate navigation targets are known for the teacher policy to develop versatile motor skills, then jointly train navigation and locomotion for the student policy during distillation. It improves skill acquisition but complicates action learning when integrating visual inputs, increasing the risk of errors and damage from unexpected actions. ***Our second contribution is a hierarchical loss metric set that distills policy based on action rationality — maintaining the structural relationships between joint movements.*** Conventional privileged learning supervises overall data distribution [40] or per-joint errors [18], ignoring inter-joint dependencies. We address this by using structured latent representations and masked reconstruction through Variational Autoencoders (VAEs) [16]. Masking during reconstruction constrains joint dependencies, creating a task-agnostic hierarchical loss set that improves policy learning across motor tasks. Since the latent prior comes from the oracle policy, not human motion data, the robot learns self-reliant motor behaviors suited to its own structure.

We demonstrate LEGO-H's robustness and versatility on diverse simulated hiking trails using a low-cost humanoid, Unitree H1 [43]. Ablation studies confirm the effectiveness of our design, and LEGO-H generalizes well to other types of humanoids such as Unitree G1. Contributions are summarised as follows: **1)** We propose hiking as a testbed for integrative humanoid skill development. **2)** We introduce LEGO-H, a learning framework for independent humanoid hiking. **3)** Experiments on diverse simulated trails and humanoids with different morphologies demonstrate LEGO-H's robustness and versatility, laying a strong foundation for future humanoid development.

2. Related Work

Humanoid locomotion. Existing approaches to low-level motor skill learning typically simplify environmental interactions, abstracting terrains into static patterns at a *momentary* scale, which neglects occlusions caused by obstacles or dynamic environmental disruptions. Research in this domain has primarily focused on learning specific locomotion skills such as walking [5, 13, 21, 32, 33], running [38, 39], and soccer-playing behaviors [14]. These approaches often rely on highly engineered designs optimized for specific lower-body tasks. Other works employ imitation learning [20, 24, 30, 31, 41] to generate human-like behaviors from large-scale motion datasets, but this comes at the cost of reduced embodiment. Some frameworks attempt to push the boundaries of robotic motor skills by exploring tasks like parkour [9, 48], acrobatic flipping [7], or cliffside climbing [47]. While impressive, these methods are often bogged down by complex engineering, reliance on user commands for motion planning, or lack of perceptual awareness.

Humanoid navigation. Research on this direction often struggles to address *real-time* environmental constraints while accounting for the unique mechanisms and actions of humanoid robots. These limitations frequently lead to suboptimal navigation plans in complex terrains. Conventional methods typically rely on scene mapping [8, 28] or structured world assumptions [26], which restrict adaptability in dynamic and unstructured environments. Contact-aware approaches [22, 23] attempt to bridge robot configurations with environmental constraints, but they often depend on pre-generated trajectories, limiting responsiveness. Similarly, mapless methods [4] leverage visual inputs for navigation but are typically constrained to basic locomotion capabilities such as walking. Recent advancements in large language and vision-language models have shown potential for complex high-level planning [44], yet remain uncoupled from motor control systems, failing to achieve autonomous perceptual awareness and last-step feasibility required for navigating diverse, fine-grained environments, like hiking.

Joint learning of navigation and locomotion. Integrating navigation and locomotion into a unified framework remains a significant challenge. In the realm of wheeled-legged and quadruped robots, several studies [15, 19, 35, 45] have explored paradigms that unify local navigation and locomotion. While these approaches provide valuable insights, tailoring them to humanoid robots as a baseline for hiking tasks reveals several critical gaps. First, humanoid robots possess significantly more degrees of freedom (DoF) than quadrupeds or wheeled-legged robots, complicating the development of stable locomotion policies. Achieving balance across diverse lower-body motor skills (*e.g.,* walking, jumping, and leaping *etc.*) within a single framework remains an open problem. Second, the greater body height of humanoid robots introduces challenges in visual perception, expanding their field of view and capturing a broader range of distances. This increased perceptual complexity exacerbates the misalignment between environmental sensing and physical contact, further complicating decision-making, navigation, and motor execution processes.

Refer to Appendix for discussion on HRL, and Privileged Learning, which form foundational pillars of our approach.

3. LEGO-H Framework

Sec. 3.1 concretes the definition of *hiking* task. Sec. 3.2 provides a concise system overview of LEGO-H. Sec 3.3 - 3.5 unfold the details of LEGO-H's learning process.

3.1. Task Definition

Drawing from human hiking paradigm [2], and considering the payloads a humanoid robot can carry in the real world (*e.g.,* a front-mounted Depth camera, and a GPS antenna), we formalize the definition of hiking task as follows: *Traversing a trail from a designed starting point P_A to endpoint P_B, with efficiency, safety, and all-level autonomy.* The given conditions for the humanoid robot to hike are: (1) relative position distance D_{rb} between endpoint B and humanoid robot's root at the moment based on GPS; (2) GPS-based M relative distances $\{D_{rm}\}_{m=1}^{M}$ between m_{th} intermediate sparse waypoints along the trail and humanoid robot's root; Here, we set $M = 1$, designating a single waypoint at the trail's midpoint to account for potential forks in real-world trails. (3) The onboard proprioception data $\mathcal{X}_{pro}$ from internal variables like velocity and torque, providing feedback on robot's physical state. (4) Vision sensor data $\{C_k\}_{k=1}^{K}$ [1] from a depth camera that lies in the humanoid robot's eye region. For ideal hiking, whole-body control would enable optimal balance and support in trail scenarios where hands provide additional contact points to coordinate with feet. However, as a baseline prototype for this new task—and noting that many trails can still be traversed with leg movement alone—this study simplifies the task by freezing humanoid's upper-body pose, focusing on lower-body functionality.

3.2. LEGO-H System Overview

The **Primary** goal for hiking is to reach P_B with safety. From a framework perspective, the humanoid robot needs to be designed with two key requisites: (1) to *autonomously* assess and adapt its local path based on environmental conditions spanning multiple time scales, as well as the motor skills executable at each moment; (2) to enable motor skills that, while *unrestricted* to any pre-designed modes, but execute *reasonably* to prevent damage, exhibiting flexible adaptability to foster emerged behaviors that optimally meet dynamic environmental demands.

[1] k is the index of k_{th} depth frame.

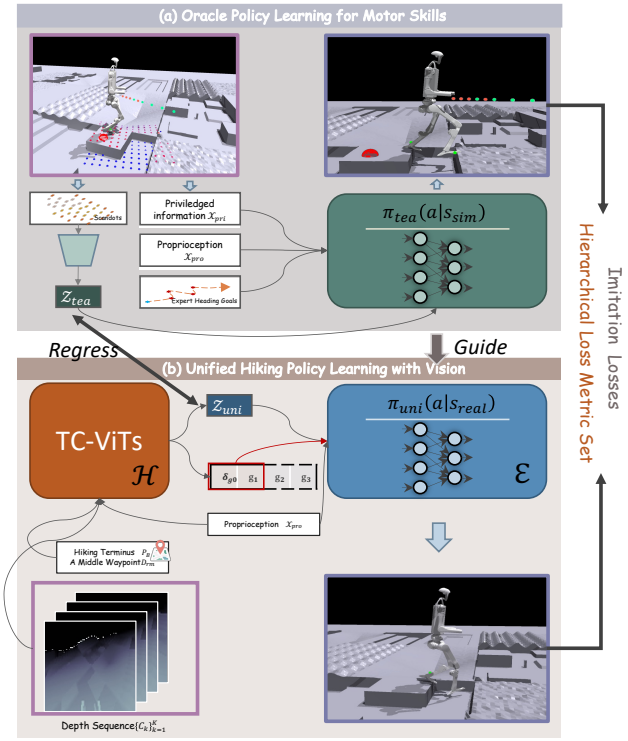

Figure 3. **LEGO-H Framework Overview.** LEGO-H equips humanoid robots with adaptive hiking skills by integrating navigation $\mathcal{H}$ and locomotion $\mathcal{E}$ in a unified, end-to-end learning framework (b). To foster the versatility of motor skills, we train the unified policy via privileged learning from oracle policy (a).

To this end, we present LEGO-H, as outlined in Fig 3. The robot employs two levels of modules (*i.e.,* navigation module $\mathcal{H}$ and motor skill module $\mathcal{E}$) within a unified pipeline (Fig 3(b)) to fulfill the first requisite. Specifically, given the state $\mathbf{s_{real}}$ – depth sensor data $\{C_k\}_{k=1}^{K}$, proprioception date $\mathcal{X}_{pro}$, endpoint information P_B, and one middle waypoint M, our navigation module $\mathcal{H}$, implemented as TC-ViTs (Sec 3.3), generates the implicit latent representation $\mathbf{z}_{uni}$ of the surrounding trail, and anticipates N *future* local navigation goals $G = \{g_n\}_{n=1}^{N}$ [2] as well as the residual δ_{g_0}, which captures the difference between the *last step*'s anticipated goal and the actual result after execution. The latent $\mathbf{z}_{uni}$, proprioception $\mathcal{X}_{pro}$, residual δ_{g_0}, and *only* the local goal g_1 that is *nearest* to the robot, flow to the low-level motor skill module $\mathcal{E}$ to *softly* guide the emergence of current executable actions $\mathbf{a}_t$ towards the trail's endpoint, rather than enforcing rigid alignment with $\mathcal{H}$'s sequential local goal anticipation. This unified pipeline enables the robot to autonomously select, adapt, and navigate local paths within traversable regions, avoiding entrapment in challenging trail sections, and collisions with obstacles through visual perception and physical feedback.

[2]In practice, $g_n \in [0, 2\pi]$, indicating the goal direction, is represented as yaw angle, and measured from the robot's root.

To address the second requisite, we tailor the privileged learning scheme. Concretely, before training the unified pipeline, we first train an oracle motor skill policy $\pi_{tea}(a|s_{sim})$ with privileged information $\mathcal{X}_{pri}$ (*e.g.,* terrain type, ground friction, and precise state measurements that are unavailable for unified pipeline stage) and expert navigation goals as oracle inputs (Fig 3(a)). Although this teacher stage operates without vision, $\mathcal{X}_{pri}$ and scandots (which represent scanned heights around the robot's feet) provide clear and precise data to facilitate high-performance motion skill learning. Then, in the unified pipeline training, the teacher policy is distilled into $\mathcal{E}$ using a Hierarchical Loss Metric Set (Sec. 3.6) to ensure both diversity and robustness in final motor skills from the policy $\pi_{uni}(a|s_{real})$. This extended regulation for privileged learning scheme ensures robots' stable and efficient movements with adaptive locomotion skills across diverse trail terrains.

3.3. TC-ViTs for Local Navigation Anticipation

Within LEGO-H, TC-ViTs (*Temporal Information Conditioned Vision Transformer Variants*) serves as central mechanism to achieve unified policy learning with visual perception, by addressing three critical questions to navigation module: (1) how to cognize the *past*,*current* and *future* states of the environment to balance both short-term reactivity with long-term goal alignment? (2) How to predict the *future* that remains adaptable to dynamic changes of environment and motor actuation? and (3) how to achieve seamless interplay between navigation and motor actuation, given their functional differences in time scale within real-world hardware settings? Fig 4 shows a detailed illustration of TC-ViTs.

Cognize surroundings with final goal. An intuitive approach to perceiving the surrounding environment would assue Markovian observations and parse depth images between adjacent frames via methods like explicit 3D modeling [19]/reconstruction [46], temporal visual feature extractor [6], or secondary task like semantic-aware traversable region prediction [10]. However, two critical issues arise when applying these approaches to hiking. First, the *time scale* challenge: both short-term dynamics and long-term environmental dependencies must be considered simultaneously. Second, the *specificity* requirement: perceived visual information should directly support the execution of the immediate next step while ensuring alignment with final goal.

A straightforward solution is to leverage a temporal vision transformer with goal-oriented conditioning. Thus, the first part of TC-ViTs (Fig 4 (a)) absorbs the encoder of a classic temporal vision transformer, ViViT [3], to capture the information with both spatial and long-range dependencies via extracting spatio-temporal tokens from the input depth sequences (16 frames, and downsample to 4), and processing them through 6 transformer layers. Each layer contains multi-headed self-attention in both spatial and temporal di-

mensions, with spatial attention applied first, followed by temporal attention. To ensure a continuous, tight association between each pixel (spatial and temporal ones) and the final target, we unfold the goal information P_B as an image channel by tiling it to the size of $(1, H, W)$[3], carpeting together with spatial and temporal features via tokenization. A flattened feature vector $\boldsymbol{\alpha}(\{C_k\}_{k=1}^{K}, P_B)$ is obtained from the final layer of the encoder. Compared to simply treating the transformer as a depth sequence extractor and feeding P_B directly to the later module, this design ensures that the goal information is embedded alongside spatial and temporal features through tokenization, maintaining a cohesive alignment with the target throughout the navigation process.

Anticipate the near future. However, the above design might be well-suited for embodied-agnostic, long-horizon navigation task [37] that making decisions in *coarse* level, but not insufficient for humanoid hiking, which demands *multiple granularity* decisions. As shown in Fig 1, hiking scenarios often encompass complex, uneven terrains, and sudden obstacles, where rapid adjustments are necessary, relying solely on temporal transformers for spatial and temporal understanding dilute the immediate spatial detail needed for precise foot placement and reactive balance adjustments. Thus, integrating spatially precise information that directly reflects the current state is essential. To this end, the second part of TC-ViTs (Fig 4(b)) provides dual processing on the current depth image. Specifically, each current depth is processed independently through a shallow CNN, yielding high-resolution spatial features $\boldsymbol{\beta}(C_{k=t})$ that retain critical, near-field spatial information. Since this branch is focused on immediate dynamics, goal conditioning is not applied here to master.

This dual-processing setup preserves fine-grained spatial details needed for near-term navigation, while goal conditioning temporal transformer manages broader patterns across time scales. The outputs of both are concatenated and passed through MLPs to complete each other, where $\boldsymbol{\gamma} = MLPs(concat(\boldsymbol{\alpha}, \boldsymbol{\beta}))$. While, beyond capturing the dynamics of surrounding environment, it's essential to account for how motor actuation and physical body states influence decision-making. Thus, in the third part of TC-ViTs (Fig 4(c)), we implement a recurrent goal adaptation mechanism that integrates visual awareness, goal information, and proprioception. This allows TC-ViTs to learn a latent representation that encodes the world with both visual perception and embodiment, as well as anticipated navigation goals and current goal residual between execution and the last step prediction. These serve as soft guidance for the locomotion module. Concretely, the input to this component includes visual representation $\boldsymbol{\gamma}$, endpoint P_B, a middle waypoint $\mathcal{D}_{rm}$, and proprioception $\mathcal{X}_{pro}$. These inputs flow into two-layer

[3]In practice, we set H and W to 128. We patchify both image and frame with size 16×16.

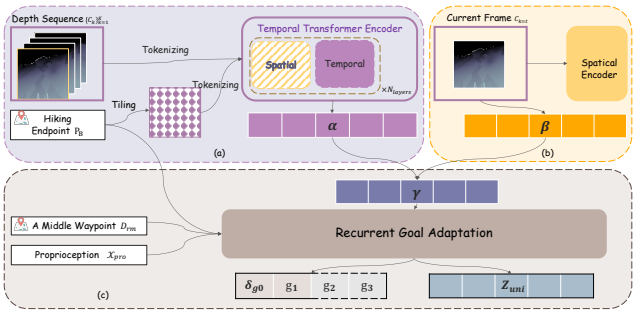

Figure 4. **Architecture of TC-ViTs.** Three key components: (a) a goal-orientated temporal transformer encoder for robots cognizing surroundings with the final goal; (2) a dual process on the current depth frame for integrating spatially precise information to reflect the current state (c) a recurrent goal adaptation mechanism that integrates visual awareness, goal information, and proprioception.

MLPs, and following with a GRU to model sequential dependencies between past, present, and anticipated future states: $\mathbf{z}_{uni}, \delta_{g_0}, G = GRU(MLPs(\boldsymbol{\gamma}, P_B, \mathcal{D}_{rm}, \mathcal{X}_{pro}))$. This structure enables TC-ViTs to capture visual cues with proprioceptive insights, ensuring the navigation goal suggestion is both perceptive and responsive to the robot's physical capabilities (as shown in Fig. 5).

Seamless interplay between the two levels. Still and all, due to hardware limitations in real robots, navigation and locomotion modules operate distinct time scales. Take Unitree H1 as an example, RealSense D435i depth sensor in the real world functions at 10 ± 2Hz to capture depth sequences, and the policy's computation frequency is set at 50 Hz on Jetson NX. This disparity introduces a gap in synchronism, where visual information and navigation decisions can lag behind the motor actuation. We use two strategies in TC-ViTs to tackle the issues: (1) *Nearest Goal Forwarding*. For the anticipated goals, we *only* forward the local goal g_1 that is *nearest* to the robot to the locomotion module. This prevents long-term error accumulation in navigation decisions and allows for timely adjustments to account for dynamic changes and motor execution, as demonstrated in the Appendix. (2) *Latent State Tiling*. $\mathbf{z}_{uni}$ is tiled five times before flowing to locomotion module. This tiling ensures continuous message broadcasting and a stable connection between navigation and locomotion processes. Together, these two strategies allow for seamless interplay between the two levels, ensuring the two modules work in harmony across time scales.

3.4. Oracle Policy Learning for Motor Skills

With TC-ViTs, local navigation and locomotion could be bridged in a unified manner. While, before training the unified policy pipeline, we first leverage privileged information to learn an *Oracle* policy for locomotion (Fig 3(a)), to ensure the diversity of motor skills. Concretely, $\mathcal{X}_{pro}$, current navigation goal, the latent $\mathbf{z_{tea}}$ of the scandots $\mathcal{S} \in \mathbb{R}^{66\times2}$ and $\mathcal{X}_{pri}$, are input to the oracle policy. To encourage

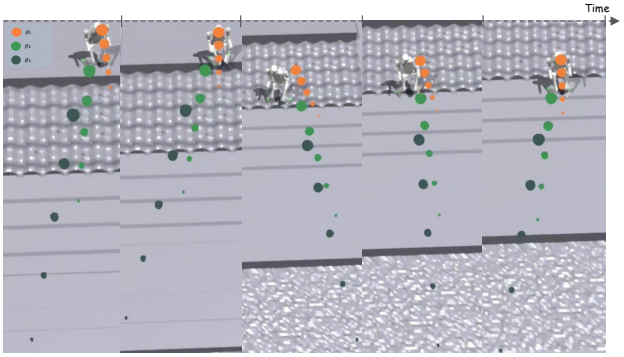

Figure 5. **Dynamic adjustments of near goal predictions.** Snapshots from left to right show a robot traversing mixed terrains along a trail. TC-ViTs does not provide a fixed trajectory that the locomotion module must rigidly follow. Instead, it predicts several near-future goals (g_1, g_2, g_3), which dynamically adapt to the robot's current state, reflecting real-time adjustments to its navigation decisions. Bubble size represents the predicted local navigation direction (from large to small).

self-emergent motor behavior rather than per-defined motor modes, and promote upright locomotion, rewards on three aspects are essential in this stage: velocity tracking alignment wrt waypoint direction r_{tracking}, soft torso height range constrain $r_{\text{base\_height}}$, and accumulation on feet air time $r_{\text{air\_time}}$. Due to space limits, please refer to Appendix for details.

3.5. Unified Hiking Policy Learning with Vision

Once oracle policy $\pi_{tea}(a|s_{sim})$ is learned, we proceed to distill it into student stage, simultaneously learning navigation and motor skills from visual observations via LEGO-H. Specifically, as shown in Fig 3(b), LEGO-H first processes depth sequences to generate latent $\mathbf{z}_{uni}$ and anticipates near-future navigation goals via TC-ViTs. Then, $\mathbf{z}_{uni}, \delta_{g_0}, g_1$ flow to locomotion module to obtain $\pi_{uni}(a|s_{real})$ that predict a_t. Both π implemented as MLPs, and basic training losses for the unified pipeline are reconstructions for imitation in goal, latent, and action levels from teacher stage:

$$\begin{aligned}\mathcal{L}_{im} = w_1\|\mathbf{z}_{tea} - \mathbf{z}_{uni}\|^2 + w_2\,\text{SmoothL1}(\mathbf{G}_{tea}, \mathbf{G}_{uni}) \\ + w_3\,\text{SmoothL1}(\mathbf{a}_{tea}, \mathbf{a}_{uni}) \end{aligned} \tag{1}$$

3.6. Hierarchical Loss Metric Set

To mitigate challenges introduced by modality and representation disparities between teacher and student stages, the constraints on students' action space are essential during policy distillation. We bound the student's action space from the structural rationality aspect, with loss metrics that reflect the structural similarity between teacher and student actions at each execution. Specifically, we employ a Variational Autoencoder with masking to implicitly learn hierarchical relationships in the teacher's action space. This enables student policy to internalize structured dependencies among joints, fostering rational and coordinated actions.

Hierarchical action structure prior learning. Concretely, during distillation, the VAE is *iteratively* trained on teacher actions with randomly masked portions, requiring it to reconstruct the full action vector from partial inputs, where:

$$\mathcal{L}_{rec} = w_4\mathcal{L}_{KL} + w_5\mathcal{L}_{self} + w_6\mathcal{L}_{mask} \tag{2}$$

$$\mathcal{L}_{KL} = KL\left(q(\mathbf{z}_{\mathbf{vae}}|\mathbf{a}_{\text{tea}}) \parallel \mathcal{N}(0, I)\right) \tag{3}$$

$$\mathcal{L}_{self} = \text{SmoothL1}(\text{Dec}(\text{Enc}(\mathbf{a}_{\text{tea}})), \mathbf{a}_{tea}) \tag{4}$$

$$\mathcal{L}_{mask} = \text{SmoothL1}(\text{Dec}(\text{Enc}(\mathbf{a}_{\text{tmask}})), \mathbf{a}_{tea}) \tag{5}$$

The w_x are the scaling weights, z_{vae} is the latent space of the autoencoder, a_{tmask} is the masked teacher action with randomly selected masking portion, and the Kullback-Leibler term *KL* follows the VAE formulation [16]. As the joint actions are unordered, we add positional embedding with sine and cosine functions to each joint. Upon the compactness of latent space driven by the underlying normal distribution, the masking further encourages the latent space of the VAE to learn inter-joint dependencies and structural consistencies that *align closely with the robot's physical mechanism, rather than motion prior from human data.*

Hierarchical action prior penalizer. Once trained, the VAE encoder is used to measure alignment between teacher and student actions in structured feature space. For each student action vector $\mathbf{a}_{\text{uni}}$, we compute hierarchical consistency loss $\mathcal{L}_{\text{ts}}$ relative to the corresponding teacher action:

$$\mathcal{L}_{ts} = 1 - \text{cos\_sim}(\text{Enc}(\mathbf{a}_{tea}), \text{Enc}(\mathbf{a}_{uni})) \tag{6}$$

$$= 1 - \frac{\text{Enc}(\mathbf{a}_{tea}) \cdot \text{Enc}(\mathbf{a}_{uni})}{\|\text{Enc}(\mathbf{a}_{tea})\|\|\text{Enc}(\mathbf{a}_{uni})\|} \tag{7}$$

To further enhance hierarchy, we introduce the triplet distance with masking:

$$\mathcal{L}_{trip} = c_{mt}(1 - \text{cos\_sim}(\text{Enc}(\mathbf{a}_{tea}), \text{Enc}(\mathbf{a}_{umask}))) \tag{8}$$

$$+c_{ms}(1 - \text{cos\_sim}(\text{Enc}(\mathbf{a}_{uni}), \text{Enc}(\mathbf{a}_{umask}))) \tag{9}$$

where $\mathbf{a}_{\text{umask}}$ is a randomly masked student action, and c_{mt} and c_{ms} are scaling factors for the triplet distance terms. Together, the hierarchical loss metric set is:

$$\mathcal{L}_{hie} = w_7\mathcal{L}_{ts} + w_8\mathcal{L}_{trip} \tag{10}$$

See Appendix for hyperparameters. As shown in Tab 1, student robots trained without these losses display task-completing motor behaviors that risk mechanical integrity due to frequent collisions. In contrast, with hierarchical losses, robots exhibit more refined, collision-free movements that align better with internal structural consistency.

4. Experiments

We evaluate the effectiveness of LEGO-H across several dimensions. First, we conduct ablations (Sec 4.2) to assess individual components. Then, we analyze robot's emerged behaviors across different levels (Sec 4.3). Finally, as a new task, we benchmark humanoid hiking in diverse simulated trail environments, covering LEGO-H, and other representative methodologies tailored to this task (Sec 4.4). We detail experimental setup on robot configurations/models/evaluation metrics (Sec 4.1). *Refer to Appendix for more details.*

4.1. Experimental Settings

Robots. We use Unitree H1 [43] and G1 [42] humanoids, chosen for their distinct differences in body scale and mechanism: H1, at adult size (5.9 ft/47kg), contrasts with kid-sized G1 (4.26 ft/35kg), with notable variations in torque density and morphology. These inherent differences impact key factors like visual perception range/motor stability/overall movement complexity even within identical trails.

Implementations. *Proprioception* ($\mathcal{X}_{pro} \in \mathbb{R}^{40}$): covers lower-body joint positions, velocities, torso roll and pitch, foot contact indicators, and previous action a_{t-1} for both robots. *Actions* ($a_t \in \mathbb{R}^{10}$): the learned policy uses position control for joints, with positions converted to torque via a PD controller $\tau = K_p(\hat{q} - q) + K_d(\dot{\hat{q}} - \dot{q})$ with fixed gains (K_p and K_d follow default configuration of Unitree). *Training*: for both oracle and unified policy training, we use PPO [36], supported by Dagger [34] and Actor-Critic [17] for privileged learning. Rewards follow those introduced in method section, with additional basic elements from [6, 12]. All physics simulations perform in Isaac Gym simulator [25].

Metrics. We evaluate models based on three core criteria with levels of granularity: *goal completeness*, *safeness*, and *efficiency*. Concretely, we use 6 evaluation metrics – (1) *Goal Completeness*: Success Rate (%) measuring the percentage of episodes where robots reach the hiking endpoint; Trail Completion (%) indicating the portion of the trail route a robot passed; and Traverse Rate (%) reflecting the distance from robot's final position (if not complete goal) to endpoint relative to total trail length. (2) *Safeness*: MEV (%) assessing foot-edge collisions; and TTF ($seconds$) evaluating robot stability based on episode duration before a fall occurs. (3) *Efficiency*: Time-to-Reach ($seconds$) measuring average time required for successful episodes to reach endpoint. Unless specified, experiments are conducted with 512 randomly spawned robots over 30 seconds on 5 distinct trail types, each featuring 5 difficulty levels. Results are averaged over 5 runs to minimize random biases and verify robustness.

4.2. Ablation Study

Settings. We compare full LEGO-H with following designs: (1)*Oracle*: trained with access to privileged info and expert-designed navigation goals, representing an upper-bound performance.(2) *w TC-ViTs*: LEGO-H trained without Hierarchical Loss Metric set (HLM). (3)*Vanilla*: LEGO-H variant where TC-ViTs is replaced by a ConvGRU to predict latent and goal, altering the navigation mechanism. *We draw key observations here. Refer to Appendix for more detailed comparisons and analysis.*

Table 1. **Ablation study of LEGO-H's main Components on H1 robot.** for best goal completeness; for most safeness; for best efficiency. Refer to Appendix for more ablations.

Metrics	Oracle	LEGO-H	w TC-ViTs	Vanilla
Success Rate (SR) (%) ↑	71.20 ± 0.72	68.40 ± 1.34	64.73 ± 2.22	42.97 ± 0.67
Trail Completion (TC) (%) ↑	77.73 ± 0.92	52.78 ± 1.30	52.50 ± 1.52	32.01 ± 0.61
Traverse Rate (TR) (%) ↑	73.60 ± 0.81	71.96 ± 2.37	72.04 ± 0.98	60.26 ± 0.94
MEV (%) ↓	7.12 ± 0.92	7.84 ± 0.92	10.40 ± 1.50	9.41 ± 1.27
TTF (s) ↑	7.25 ± 0.09	7.46 ± 0.17	7.00 ± 0.20	5.36 ± 0.10
T2R (s) ↓	4.59 ± 0.08	4.95 ± 0.12	5.13 ± 0.12	6.50 ± 0.07

Results. Tab 1 indicates several insights. (1) *TC-ViTs is essential for basic hiking functionality.* The consistent, significant performance advantage of *w TC-ViTs* over *Vanilla* across all metrics, except MEV, reveals the essence of balancing the goal, physical state, and visual perception, which is crucial for coordination between navigation and locomotion.(2) *Structural action behavior helps more efficient goal accomplishment and better stability.* The absence of HLM (*w TC-ViTs)* results in behaviors that complete tasks but compromise stability, often leading to mechanical risks (worse MEV than others). Including HLM (*LEGO-H)* ensures coordinated joint actions that align with the robot's physical structure, promoting both task success (SR rises from 64.73% to 68.40%) and mechanical integrity (MEV goes from 10.40% to 7.84%, TTF increase to $7.46s$), leading to more efficient task accomplishment (T2R improves from $5.13s$ to $4.95s$). (3) *LEGO-H rivals oracle in efficiency and safety.* Compared to oracle which has perfect observation conditions and expert navigation goals, LEGO-H falls behind on success rate and trail completion. But surprising aspects are the efficiency and safeness, where LEGO-H's performances are comparable to or slightly better than oracle. This stresses again LEGO-H's effectiveness and capacity.

4.3. Emerged Behaviors in Different Situations

We further explore the behaviors that emerge in humanoid robots to unfold how robots autonomously adapt their motor skills and decision-making in response to various factors.

Locomotion in diverse trail terrains. As shown in Fig 6, different terrain types trigger distinct locomotion behaviors, like *walking, stepping, jumping, leaping, and leaning sideways*. Key observations include: (1) H1 robots typically opt for a walking gait on continuous surfaces, regardless of variations in friction, adjusting their body tilt as needed to maintain balance (Fig 6 (a)). (2) Irregular surfaces, like fractured or sloped terrains, prompt gaits like stepping, jumping, or leaping, depending on slope and gap size (Fig 6 (b)). (3) In tight spaces, such as cracks between large obstacles, H1's

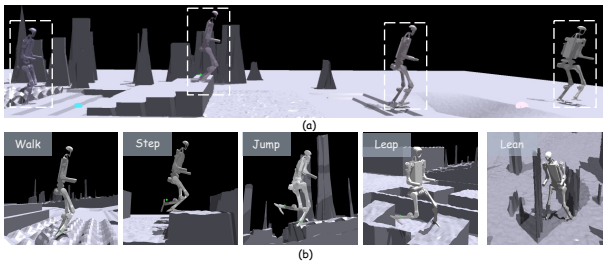

Figure 6. **Locomotion in diverse trail terrains.** Robots developed distinct motor skills to tackle different terrains, *e.g.,* walking on rough surfaces/leaping across ditches/leaning away high obstacles.

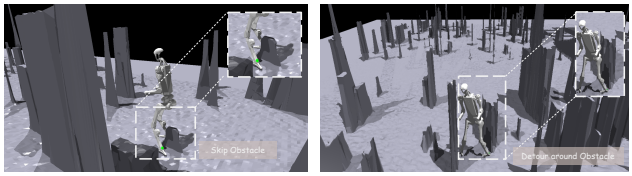

Figure 7. **Navigation in diverse situations.** Robots developed different navigation skills to tackle different situations, such as directly skipping a small obstacle and detouring around a high obstacle to edge through.

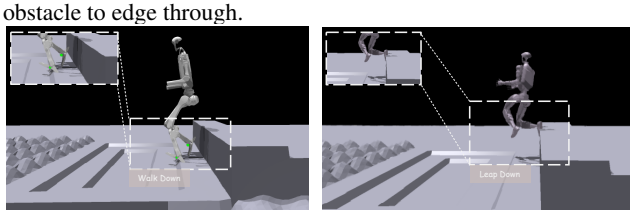

Figure 8. **Motor behavior differences between robots.** Robots with different structures developed unique skills – H1, which is higher and heavier, chooses to "walk down" step, while G1, which is shorter and more lightweight, chooses to "leap down" the step.

adapt by leaning sideways to navigate through these confined areas (*Lean* in Fig 6 (b)).

Navigation in blocked paths. Two key behaviors are evident from Fig 7: (1) When faced with tall or large obstacles, the robots typically choose to detour, maintaining a safe clearance from the obstacles. (2) For obstacles below hip height, the robots initially attempt to stride or step over; if unsuccessful, they then choose to detour. These phenomena reveal the embodied character in high-level decisions. Refer to the *Appendix* for more evaluations.

Motor behavior differences between robots. As shown in Fig 8, when encountering identical trails like transitions between platform and flat ground, H1 and G1 exhibit different behaviors. H1 navigates down smoothly, while G1 bends its knees to jump down. This difference highlights the impact of physical mechanisms on emergent motor styles.

4.4. Humanoid Hiking Benchmark

Settings. Since current research does not directly support humanoid hiking, we selected two representative quadruped pipelines, adapting them to this task using the same input structure and oracle policy as LEGO-H. This setup allows us to investigate several key factors essential for effective humanoid hiking. The first adapted pipeline, *EP-H*, represents a modified humanoid-hiking version of EP [6]. The main methodological difference between EP-H and LEGO-H is that EP-H handles visual-aware navigation and locomotion by processing each depth frame independently, disregarding farther depth data to avoid distributional shifts. *RMA-H* and *RMA-B* are the adapted pipeline from RMA [18] – the former has vision inputs, and the later is blind. This pipeline originally supports blind locomotion, and employs a frozen oracle policy with an adapter network to map real-world sensory data to the oracle's latent space for policy adaptation.

Results. We focus on three vital questions from the benchmark: 1) *Is visual perception essential for integrated navigation and locomotion?* 2) *What type of visual information is most effective?* 3) *Is unified cross-level learning necessary?* Key findings in Tab 2 and visualizations in Appendix revel the answers:(1) *Vision is essential.* Without vision, RMA-B struggles across all metrics, highlighting the need for visual feedback. (2) *Goal-aligned, multi-scale visual perception is critical. EP-H*, which processes each depth frame independently without continuous goal alignment, and brute-force cutoff distance information, results in frequent circles and fails to lock onto navigation paths. The performance gap between LEGO-H and EP-H across metrics underscores the importance of structured visual information. (3) *Unified learning is vital for adaptability. RMA-H* performs adequately on straight paths but fails with turns or obstacles, showing that locomotion feedback alone is insufficient for embodied-aware decision-making. A unified learning framework supports essential cross-level interaction, enabling adaption and effectiveness across all levels.

Table 2. **Hiking benchmark for Humanoid Robot H1 across all different trail categories.** for best goal completeness; for most safeness; for best efficiency.

Metrics	LEGO-H	EP-H	RMA-H	RMA-B
Success Rate (%) ↑	68.40 ± 1.34	28.80 ± 0.88	65.17 ± 2.05	48.11 ± 0.72
Trail Completion (%) ↑	52.78 ± 1.30	25.98 ± 0.22	52.51 ± 1.41	41.92 ± 0.34
Traverse Rate (%) ↑	71.96 ± 2.37	64.16 ± 0.48	74.61 ± 0.93	69.85 ± 1.50
MEV (%) ↓	7.84 ± 0.92	12.44 ± 1.32	8.70 ± 1.55	10.74 ± 1.13
TTF (s) ↑	7.46 ± 0.17	4.64 ± 0.13	6.97 ± 0.17	5.22 ± 0.03
Time-to-Reach (s) ↓	4.95 ± 0.12	9.79 ± 0.16	4.98 ± 0.11	6.19 ± 0.05

5. Conclusion

LEGO-H stresses the importance of integrative multi-level development in advancing humanoid robot autonomous capabilities for complex tasks like hiking. It unifies locomotion and navigation within an end-to-end policy framework, achieved by: (1) a Temporal Vision Transformer variant in HRL, re-framing navigation as a sequential anticipation to softly guide rather than rigidly enforce locomotion; (2) a hierarchical metric set leveraging robot's inherent structure for task-agnostic supervision to policy distillation.

References

[1] Junhyeok Ahn, Steven Jens Jorgensen, SeungHyeon Bang, and Luis Sentis. Versatile locomotion planning and control for humanoid robots. *Frontiers Robotics AI*, 2021. 2

[2] AllTrails. https://www.alltrails.com/. 3

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *International Conference on Computer Vision (ICCV)*, 2021. 4

[4] André Brandenburger, Diego Rodriguez, and Sven Behnke. Mapless humanoid navigation using learned latent dynamics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. IEEE, 2021. 3

[5] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *RSS*, 2024. 3

[6] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *IEEE International Conference on Robotics and Automation, ICRA*, 2024. 4, 7, 8

[7] Matthew Chignoli, Donghyun Kim, Elijah Stanger-Jones, and Sangbae Kim. The MIT humanoid robot: Design, motion planning, and control for acrobatic behaviors. In *IEEE-RAS International Conference on Humanoid Robots, Humanoids*, 2021. 3

[8] Octavian A. Donca, Chayapol Beokhaimook, and Ayonga Hereid. Real-time navigation for bipedal robots in dynamic environments. *CoRR*, 2022. 3

[9] Boston Dynamics. https://bostondynamics.com/blog/leaps-bounds-and-backflips/. 2018. 2, 3

[10] Jonas Frey, Matias Mattamala, Libera Piotr, Nived Chebrolu, Cesar Cadena, Georg Martius, Marco Hutter, and Maurice Fallon. Wild Visual Navigation: Fast Traversability Learning via Pre-Trained Models and Online Self-Supervision. In *Arxiv*, 2024. 4

[11] Johannes Garimort, Armin Hornung, and Maren Bennewitz. Humanoid navigation with dynamic footstep plans. In *IEEE International Conference on Robotics and Automation, ICRA*, 2011. 2

[12] Xinyang Gu, Yen-Jen Wang, and Jianyu Chen. Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer. 2024. 7

[13] Xinyang Gu, Yen-Jen Wang, Xiang Zhu, Chengming Shi, Yanjiang Guo, Yichen Liu, and Jianyu Chen. Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning. In *Robotics: Science and Systems*, 2024. 3

[14] Tuomas Haarnoja, Ben Moran, Guy Lever, Sandy H. Huang, Dhruva Tirumala, Jan Humplik, Markus Wulfmeier, Saran Tunyasuvunakool, Noah Y. Siegel, Roland Hafner, Michael Bloesch, Kristian Hartikainen, Arunkumar Byravan, Leonard Hasenclever, Yuval Tassa, Fereshteh Sadeghi, Nathan Batchelor, Federico Casarini, Stefano Saliceti, Charles Game, Neil Sreendra, Kushal Patel, Marlon Gwira, Andrea Huber, Nicole Hurley, Francesco Nori, Raia Hadsell, and Nicolas Heess. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Sci. Robotics*, 2024. 3

[15] David Hoeller, Lorenz Wellhausen, Farbod Farshidian, and Marco Hutter. Learning a state representation and navigation in cluttered and dynamic environments. *IEEE Robotics Autom. Lett.*, 2021. 3

[16] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations, ICLR*, 2014. 2, 6

[17] Vijay R. Konda and John N. Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems 12, [NIPS*, 1999. 7

[18] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021. 2, 8

[19] Joonho Lee, Marko Bjelonic, Alexander Reske, Lorenz Wellhausen, Takahiro Miki, and Marco Hutter. Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Sci. Robotics*, 2024. 3, 4

[20] Chenhao Li, Elijah Stanger-Jones, Steve Heim, and Sangbae Kim. FLD: fourier latent dynamics for structured motion representation and learning. In *ICLR*, 2024. 2, 3

[21] Qiayuan Liao, Bike Zhang, Xuanyu Huang, Xiaoyu Huang, Zhongyu Li, and Koushil Sreenath. Berkeley humanoid: A research platform for learning-based control. *CoRR*, abs/2407.21781, 2024. 3

[22] Yu-Chi Lin and Dmitry Berenson. Humanoid navigation in uneven terrain using learned estimates of traversability. In *IEEE-RAS International Conference on Humanoid Robotics, Humanoids*. IEEE, 2017. 3

[23] Yu-Chi Lin and Dmitry Berenson. Long-horizon humanoid navigation planning using traversability estimates and previous experience. *Auton. Robots*, 2021. 3

[24] Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris M. Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. In *ICLR*, 2024. 3

[25] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In *NeurIPS Datasets and Benchmarks*, 2021. 7

[26] Marcell Missura, Arindam Roychoudhury, and Maren Bennewitz. Polygonal perception for mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. IEEE, 2020. 2, 3

[27] Denise Mitten, Jillisa R Overholt, Francis I Haynes, Chiara C D'Amore, and Janet C Ady. Hiking: A low-cost, accessible intervention to promote health benefits. *American journal of lifestyle medicine*, 12(4):302–310, 2018. 1

[28] Omid Mohareri and Ahmad B. Rad. Autonomous humanoid robot navigation using augmented reality technique. In *2011 IEEE International Conference on Mechatronics*, 2011. 2, 3

[29] Ingeborg Nordbø and Nina K Prebensen. Hiking as mental and physical experience. In *Advances in hospitality and leisure*, pages 169–186. 2015. 1

[30] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for styl-

ized physics-based character control. *ACM Trans. Graph.*, 2021. 2, 3

[31] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Trans. Graph.*, 2022. 3

[32] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Learning humanoid locomotion with transformers. *CoRR*, abs/2303.03381, 2023. 2, 3

[33] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Sci. Robotics*, 2024. 2, 3

[34] Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, 2011. 7

[35] Nikita Rudin, David Hoeller, Marko Bjelonic, and Marco Hutter. Advanced skills by learning locomotion and local navigation end-to-end. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2022. 3

[36] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, 2017. 7

[37] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. In *Conference on Robot Learning, CoRL*. PMLR, 2023. 5

[38] Filippo M. Smaldone, Nicola Scianca, Leonardo Lanari, and Giuseppe Oriolo. From walking to running: 3d humanoid gait generation via MPC. *Frontiers Robotics AI*, 9, 2022. 3

[39] Sait Sovukluk, Johannes Englsberger, and Christian Ott. Highly maneuverable humanoid running via 3d slip+foot dynamics. *IEEE Robotics Autom. Lett.*, 9(2):1131–1138, 2024. 3

[40] Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH Conference Proceedings*, 2023. 2

[41] Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions on Graphics (TOG)*, 2024. 3

[42] UnitreeG1. https://www.unitree.com/g1/. 7

[43] UnitreeH1. https://www.unitree.com/h1. 2, 7

[44] Jin Wang, Arturo Laurenzi, and Nikolaos G. Tsagarakis. Autonomous behavior planning for humanoid loco-manipulation through grounded language model. *CoRR*, abs/2408.08282, 2024. 2, 3

[45] Ruihan Yang, Minghao Zhang, Nicklas Hansen, Huazhe Xu, and Xiaolong Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. In *ICLR*, 2022. 3

[46] Ruihan Yang, Ge Yang, and Xiaolong Wang. Neural volumetric memory for visual locomotion control. In *CVPR*, 2023. 4

[47] Chong Zhang, Wenli Xiao, Tairan He, and Guanya Shi. Wococo: Learning whole-body humanoid control with sequential contacts. *CoRL*, 2024. 3

[48] Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning. *CoRR*, abs/2406.10759, 2024. 3

Let Humanoids Hike! Integrative Skill Development on Complex Trails Appendix

Kwan-Yee Lin Stella X. Yu
University of Michigan, Ann Arbor
{junyilin,stellayu}@umich.edu

Abstract

In the appendix, we provide a comprehensive elaboration of LEGO-H. Section 1 recaps the positioning of the Humanoid Hiking task and highlights how LEGO-H departs from the current trends in humanoid robotics. Section 2 expands on related work. Section 3 delves into extended ablation studies, analyzing detailed design choices of each component in LEGO-H. Section 4 explores the framework's universality through experiments on the integration of LEGO-H components into alternative frameworks. Section 5 introduces the simulated environments developed for training and evaluation in this new hiking paradigm. Section 6 specifies implementation details. Section 7 extends evaluations on critical questions in humanoid hiking. Lastly, section 8 discusses future work.

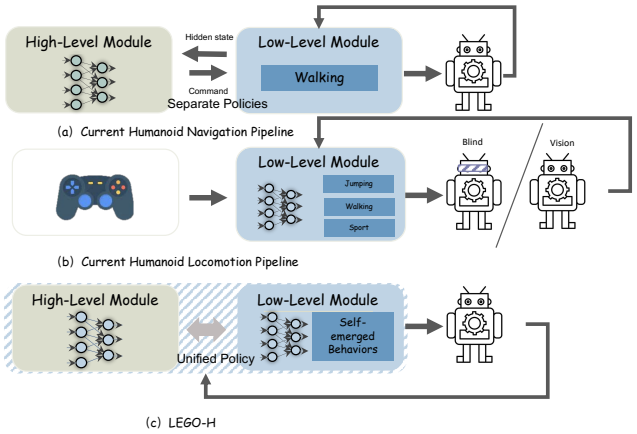

Figure 1. **The conceptual framework differences.** We summarize the key conceptual level differences between our work and current humanoid robot trends for better positioning of LEGO-H.

1. The Positioning of LEGO-H

To better understand LEGO-H's positioning, we present a conceptual framework comparison in Fig 1. LEGO-H advances humanoid robotics by seamlessly integrating navigation and locomotion into a unified policy learning framework (Fig 1(c)). This contrasts with existing pipelines, which either separate these modules (Fig 1(a)) or reduce environmental complexity by relying on external commands for action execution (Fig 1(b)).

This work emphasizes the importance of integrative development of navigation and locomotion for humanoid robots to operate effectively in complex real-world environments. Humanoid hiking provides an ideal testbed to evaluate this coordination. LEGO-H, as a baseline prototype, demonstrates how unified learning fosters self-emerged behaviors, enabling dynamic adaptation to diverse trails and challenges.

2. Additional Related Work

2.1. Hierarchical RL

It is widely adopted to decompose a complex RL problem into multiple layers of policies [4, 13]. This paradigm naturally structures in hierarchy, where a decision-making/control module at higher levels manages temporal (longer time scale) and behavioral abstraction, while a low-level module focuses on atomic skills to execute momentary actions in the environment, guided by the high-level module. HRL includes two main methodologies: (1) explicit goal setting [9], where the high-level policy assigns target goals to the low level, enhancing reusability but limiting adaptability, and (2) latent space policies [7], where high-level module guides the low-level policy by providing latent sub-goals at a lower frequency, offering flexibility but often limiting generalization. However, HRL are generally not end-to-end trainable due to complexity and distinct objectives of each level. Our *LEGO-H*, is also hierarchical but avoids strict goal adherence or explicit skill definitions. Instead, it presents a unified, end-to-end policy learning framework, where high-level module offers latent representations and intermediate goals as flexible guidance, allowing low level to reference them adaptively rather than following rigidly. This soft guidance supports adaptability and coherence in complex environments, addressing traditional HRL limitations.

2.2. Privileged Learning

It is a two-stage technique in robotics, often employed to address sim-to-real transfer challenges [2, 6, 15]. For first *teacher* stage, the robot agent learns an *oracle* policy via additionally accessing privileged information from human demonstrations [2], or GT exteroceptive measurements from simulator [6]. Since extra information reduces ambiguity via precise physical states/terrain details/expert trajectories, the agent could learn more precise actions. However, as this information is unavailable in real-world deployment, in the second *student* stage, the robot agent learns to imitate the teacher's behavior using only accessible data [1] through knowledge distillation. Common distillation losses target element-wise difference [2], distribution alignment [11] or latent space alignment [5]. However, studies rarely address the structural consistency of actions, a critical factor for humanoid hiking, where the robot's high articulation requires precise coordination across joints.

3. Additional Ablation Studies

In this section, we delve into the detailed designs of TC-ViTs (Section 3.1) and the Hierarchical Loss Metric Set (Section 3.2). Additionally, we example and analyze further emergent behaviors focusing on the *safeness* aspect, which were not covered in the main paper due to space limits.

3.1. Efficiency of TC-ViTs

In this subsection, we further analyze the efficiency behind TC-ViTs' recurrent goal adaptation module design.

Why Recurrent Goal Adaptation? As mentioned in the main paper, this module, implemented via a GRU and grafted at the end of TC-ViTs – integrates motor actuation and physical body states, enhancing visual cue processing with proprioceptive insight. While recent advances like CausalTransformers (CTs) [12, 17] have shown promising results in temporal modeling, we intentionally adopt a GRU-based design due to its better computational efficiency: TC-ViTs has Flops-$0.686G$/Params-$31.25M$, while

[1] It often includes proprioception, user commands, and visual sensor inputs.

replacing its GRU to CTs increase to $0.785G/55.92M$. Besides, CTs require significantly more computational resources for sufficient training, leading to performance degradation under the same memory constraints (Tab. 1). Since most visual information is already processed by the preceding ViViT-style encoder, CTs would introduce redundancy in such a later stage. An additional finding is that our HLM helps improve CTs performance—*e.g.*, reducing CT's collision (MEV) from 10.48% to 8.61%.

Table 1. **GRU vs CTs at the end of TC-ViTs.**

Metrics	w GRU	w CTs
Success Rate (%) ↑	68.40 ± 1.34	27.85 ± 1.02
TTF (s) ↑	7.46 ± 0.17	5.44 ± 0.34

3.2. How Hierarchical Loss Metric Set (HLM) Work

In this subsection, we further analyze the Hierarchical Loss Metric (HLM) by addressing two key questions: (1) *How does the structural rationality of actions impact the safety of the robot's movements?* (2) *Is a vanilla VAE sufficient to capture and reflect the rationality of the robot's actions?* Through these investigations, we aim to provide deeper insights into the design choices and contributions of HLM for promoting self-coordinated and safe humanoid movements across complex trails.

Ablation on w/wo HLM. We show the quantitative comparison between w/wo HLM in Tab 1 of the main paper with metric MEV. Here, as a complementary, we show qualitative samples. As shown in Fig 2, while LEGO-H without HLM achieves successful traversal over the hurdle, the mechanical risks are significantly higher. The robot's right leg collides with the hurdle during the stepping motion, and the minimal clearance further demonstrates unsafe and inefficient movement patterns. In contrast, with HLM incorporated, the robot executes structurally rational and safe movements. It first steps onto the hurdle with its left leg, ensuring sufficient clearance for the right leg, and then transitions to a stable hop onto the opposite leg. This coordinated behavior highlights the role of HLM in enabling stability, safety, and effective traversal strategies.

Vanilla VAE or full HLM? The latent space of a vanilla VAE is commonly employed for prior regularization, promoting outputs that align with the normal distribution of the data. This proves effective for tasks like approximating averages in large-scale or in-the-wild datasets, as seen in human pose reconstruction [10]. However, vanilla VAE falls short when structural dependencies and inter-joint dynamics are critical, like humanoid robot actions. Specifically, humanoid hiking with safety demands fine-grained understanding of hierarchical relationships of robots' own physical mechanism, which vanilla VAE lacks. By contrast, as demonstrated in Tab 2, full HLM introduces additional masked reconstruction and hierarchical losses that implicitly enforce inter-joint structural rationality, enabling safer and more efficient robot movement in complex tasks like humanoid hiking.

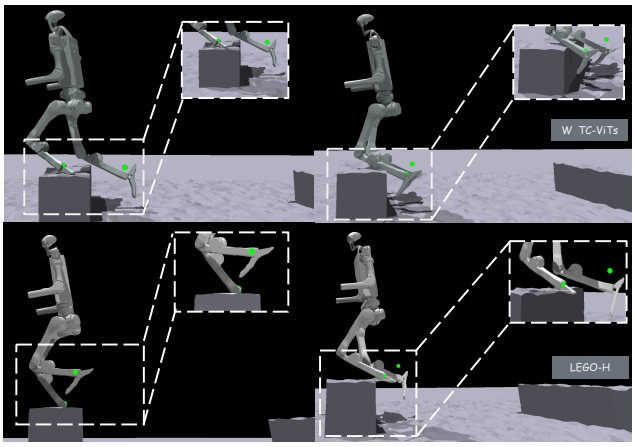

Figure 2. **Qualitative ablation on with/without HLM.** Snapshots from right to left depict two time steps of a robot traversing a hurdle obstacle. The top row illustrates behaviors without HLM, where unsafe movements lead to right leg collisions with the hurdle. The bottom row showcases behaviors with HLM, exhibiting coordinated and structurally rational actions that ensure stability and successful traversal with safe clearance.

Table 2. **Ablation of HLM.** for best goal completeness; for most safeness; for best efficiency. The results highlight the insufficiency of using a vanilla VAE as a prior. Additionally, compared with Tab. 1 in the main paper, the vanilla VAE collapses actions into average motions. While this slightly improves MEV compared to the setting without any prior (w TC-ViTs), it sacrifices performance across all other metrics.

Metrics	full HLM	Vanilla VAE
Success Rate (%) ↑	68.40 ± 1.34	53.49 ± 1.61
Trail Completion (%) ↑	52.78 ± 1.30	43.00 ± 0.96
Traverse Rate (%) ↑	71.96 ± 2.37	64.52 ± 1.02
MEV (%) ↓	7.84 ± 0.92	9.26 ± 1.08
TTF (s) ↑	7.46 ± 0.17	6.30 ± 0.15
Time-to-Reach (s) ↓	4.95 ± 0.12	6.02 ± 0.05

3.3. Emergent Behavior Analysis

In this subsection, we explore a critical question: *How do robots behave to ensure safety?* We will list three examples, considering both high-level navigation behaviors and low-level motor skill execution, to show how LEGO-H prioritizes safety in dynamic and challenging environments.

Navigation in blocked paths. As discussed in the main paper Section 4.3, robots typically opt to detour around large, tall obstacles and skip over smaller ones. Here, we show the phenomena from another aspect. In Fig 3, the traversed

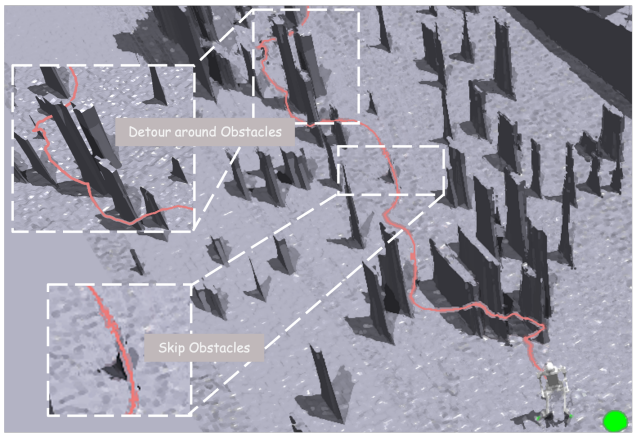

Figure 3. **Navigation in blocked paths over different obstacles.** The colored trajectory illustrates the robot's torso position as it traverses the trail. Zoomed-in regions highlight distinct navigation behaviors: when encountering crowded, tall obstacles, the robot opts to detour, whereas for smaller obstacles, the robot leaps over, demonstrating adaptive navigation strategies.

Figure 4. **Behaviors over difference terrains.** The robots exhibit diverse integrative navigation and locomotion skills tailored to varying trail terrains. (a) The robot adopts a lateral "crab" walking style to navigate a long, rugged gully, maintaining stability while progressing toward the hiking terminus. (b) The robot faces the final terminus directly and jumps over a short, smooth gully. The orange directional lines show the terminus directions.

trajectory shows substantial clearance maintained from tall obstacles (zoomed-in block: detour over obstacles) and efficient traversal above smaller ones (zoomed-in block: skip obstacles). This demonstrates the robot's ability to prioritize collision avoidance while exhibiting adaptive decision-making based on the encountered environment.

Behavior over different terrains. In the main paper Section 4.3, we discussed how diverse and distinct locomotion skills emerge to tackle different terrains. Here, we present two examples demonstrating how terrains influence the robots' integrative navigation decisions and motor execution. As shown in Fig 4: (1) for a *long, rugged* gully, the robot adopts a lateral "crab walk" strategy to maintain balance and progress towards the terminus. (2) For a *short, smooth* gully, the robot directly faces the terminus and leaps over it, showcasing adaptive integrative navigation and motor behavior responses to varying trail challenges.

Re-balancing. The ability to re-balance is critical for humanoid robots traversing complex trails. As shown in Fig 5, the robot stumbles due to uneven terrain (red timeline), triggering a sequence of emergent lateral motions that dynamically counteract the imbalance (yellow timeline). After that, the robot shows seamless coordination between re-balancing and task continuity (green timeline). This example highlights that, rather than relying on predefined recovery motions, the robot adapts its behavior dynamically to the context. Such adaptability underscores the robustness of LEGO-H's unified learning framework in fostering emergent, and context-aware integrative navigation and motor skills with safeness.

4. The Universality of LEGO-H

In this section, we explore the universality of LEGO-H by demonstrating its flexibility in two ways: (1) integrating key components like HLM into other policy learning pipelines, and (2) transferring the entire framework to a morphologically distinct humanoid robot, Unitree G1, without architecture changes.

4.1. HLM as a Plug-in Supervision

HLM focuses exclusively on maintaining structural similarity between the oracle locomotion policy's actions and the student's, making it agnostic to the student's framework design. This modularity allows HLM to be seamlessly integrated as a plug-in supervision component into different policy architectures, ensuring structural rationality and coordination without requiring changes to the underlying framework. We demonstrate this property by adding it to EP-H. The results are shown in Tab 3.

Table 3. **HLM as a plug-in supervision for other framework.**

Metrics	EP-H	EP-H + HLM
Success Rate (SR) (%) ↑	28.80 ± 0.88	35.53 ± 1.30
Trail Completion (TC) (%) ↑	25.98 ± 0.22	30.36 ± 0.89
Traverse Rate (TR) (%) ↑	64.16 ± 0.48	58.23 ± 0.76
MEV (%) ↓	12.44 ± 1.32	10.98 ± 1.40
TTF (s) ↑	4.64 ± 0.13	5.04 ± 0.16
T2R (s) ↓	9.79 ± 0.16	7.80 ± 0.37

4.2. Transfer to G1 Robot

To further evaluate the universality of LEGO-H, we retrain the framework on the Unitree G1 humanoid robot without any architectural modification — demonstrating its agnosticism to specific robot morphology. As shown in Tab 4, two key observations emerge from this transfer: (1) Framework generalization: LEGO-H can adapt to G1, despite differences in body structure and joint configuration from H1. LEGO-H on G1 preserves reasonable integrative navigation and locomotion performance. (2) Performance shift. Compared to H1, G1 exhibits lower performance in general. This is primarily due to its shorter leg length and reduced camera height, which constrain both physical reach

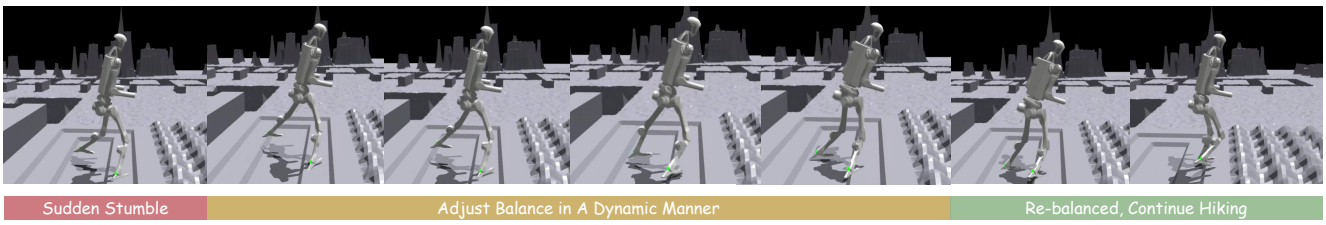

Figure 5. **Self re-balance.** The robot stumbles unexpectedly (red timeline), swiftly adjusts its balance through a sequence of emergent lateral motions (yellow timeline), and seamlessly regains stability (green timeline).

and perceptual field. Thus, on tasks requiring large clearance—such as jumping over ditches, G1 typically struggles more. A possible solution to mitigate these limitations is to extend LEGO-H with effective whole-body control (WBC) designs, allowing more expressive coordination across the upper body and the lower body. This could compensate for morphological constraints and unlock more agile, full-body responses to complex hiking trails.

Table 4. **LEGO-H on Humanoid G1 robot.** We list H1's result as a reference. The results highlight the universality of our proposed learning framework for different robot types.

Metrics	H1	G1
Success Rate (%) ↑	68.40 ± 1.34	63.96 ± 1.03
Trail Completion (%) ↑	52.78 ± 1.30	38.94 ± 0.63
Traverse Rate (%) ↑	71.96 ± 2.37	62.21 ± 0.97
MEV (%) ↓	7.84 ± 0.92	5.33 ± 0.68
TTF (s) ↑	7.46 ± 0.17	7.24 ± 0.22
Time-to-Reach (s) ↓	4.95 ± 0.12	8.10 ± 0.08

5. Simulated Hiking Trail Constructions

To establish a robust testbed for humanoid hiking tasks, we design diverse trails in the Nvidia Isaac Gym Simulator [8] using a procedural generation approach. The construction process is detailed in Section 5.1, while Section 5.2 outlines the goal and waypoint design methodology.

5.1. Trail Scene Generation

To simulate diverse trail environments for humanoid hiking, we design 16 basic terrain primitives. Each primitive is extended into multiple variants by randomly sampling terrain properties such as slope, height, and surface friction, as well as their positions, using a procedural terrain generation mechanism. These primitives form the foundation for constructing five distinct trail types, each presenting a unique combination of terrain challenges and navigation complexity. Specifically:

- *RandomMix* trail category features unobstructed views, testing the robot's ability to navigate long distances while adapting multiple motor skills to various mixed terrain types.
- *Ditch* category introduces uneven, middle-distance trails with diverse slopes and gaps, challenging the robots to decide and execute quick turns and agile leaps.
- *Hurdle* category includes trails with long, cubic obstacles, focusing on testing the robot's ability to avoid foot collisions while navigating middle distances.
- *Gap* trails with uneven jumping platforms, including varying gap distances and straight or staggered stones, evaluating the robot's balance and jumping ability during middle-distance navigation.
- *Forest* trails densely populated with variously sized and positioned obstacles, simulating obstructed views and tight navigation spaces. These test the robot's ability to detour, effectively traverse crowded paths, and maintain balance under constrained conditions.

Each trail category covers five hiking difficulty levels, with additional variants generated through the randomization of terrain properties and obstacle placement. These diversities ensure a comprehensive testbed across a wide spectrum of challenges. To expand the evaluation scope, we also construct out-of-domain hiking trails by combining multiple trail types into complex, long-distance hill scenarios. These trails test the robots' adaptability, and integrative capabilities under extended and unpredictable hiking conditions. We show the zero-shot ability of LEGO-H on the out-of-domain trails in the supplemental video.

5.2. Oracle Navigation Goal Design

The design of expert navigation goals for the oracle stage follows these criteria:

- *Unobstructed-view trails*: For trails with clear visibility, such as *RandomMix*, expert navigation goals are set as evenly spaced waypoints within the traversable regions, aligning directly with the trail direction. These goals ensure smooth long-distance navigation.
- *Obstructed-view trails*: For complex trails like *Forest*, navigation goals are dynamically set to detour around obstacles, following feasible paths with a degree of randomness to promote diverse path exploration. These goals maintain sufficient clearance to prevent collisions and en-

courage obstacle-aware navigation strategies.

- *Terrain-specific trails*: For specialized challenges like *Hurdle*, *Ditch*, and *Gap*, navigation goals are positioned to encourage the emergence of specific motor behaviors, such as agile leaps, balanced stepping, or jumping within safe zones. These goals are carefully tailored to meet the unique demands of each terrain type, ensuring both adaptability and safety.

These navigation goals establish a robust foundation for oracle policy training.

6. Experimental Details

All experiments are conducted on a single A40 GPU, though the policy can also be deployed on a more cost-effective GPU, such as the 4080. The oracle policy training requires approximately $\sim$ 18 GPU hours, while the unified policy training takes $\sim$ 2 GPU days. For camera placement, if the humanoid robots are equipped with a head-mounted camera, we use the default configuration. Otherwise, an additional camera is attached approximately at eye level. This section provides additional implementation details of LEGO-H: Section 6.1 details the architecture specifications, and Section 6.2 elaborates on the training procedures and hyperparameter configurations.

6.1. Network Architectures

This section details the network architectures of: the scandot encoder, the oracle policy, and the masked Variational Autoencoder (VAE) used in the Hierarchical Loss Metric (HLM).

Scandot Encoder. It is three layers of MLPs, with the hidden layer dimension of $[128, 64, 32]$. The activation functions are eLU for hidden layers and Tanh for the output layer.

Oracle Policy. The Actor network takes proprioceptive data, encoded scan features from the Scandot Encoder, privileged information, and encoded privileged features as inputs, and flows them into three layers of MLPs, where the dimension is $[512, 256, 128]$. The activation functions are eLU for hidden layers and Tanh for the output layer.The Critic network shares the same architecture as the Actor network. The encoder dimension for privileged information is $[64, 20]$.

Masked VAE for HLM. The architecture of the Variational Autoencoder (VAE) employed for the Hierarchical Loss Metric (HLM) consists of fully connected residual layers. The encoder includes multiple ResidualFC layers followed by two linear layers to produce the mean and log variance of the latent variable. ReLU activations are used in both the encoder and decoder, with the decoder's output layer utilizing a sigmoid activation function to ensure bounded outputs.

6.2. Training Procedure

The training process begins with the development of *oracle* policy using privileged information and expert navigation goals. Subsequently, the unified policy, incorporating TC-ViTs and the locomotion module, is trained with visual information as inputs. This stage excludes privileged information and distills motor knowledge from the oracle policy into the unified framework.

6.2.1. Oracle Policy Training

The goal of this stage is to develop an oracle locomotion policy that facilitates the training of the unified policy in the subsequent stage. Since the environment properties will be unknown in the second stage, we adopt the strategy from [3, 16] to train an adaptation module capable of estimating environment properties. The detailed training procedure is outlined below.

Curriculum Learning. To ensure stable training, we leverage curriculum learning [3, 6, 7], progressively increasing the complexity of traversable terrains based on the robots' acquired skills. This method enables gradual adaptation and robust policy development for challenging trails. Specifically, the robot's distance from the origin is tracked and compared against a threshold determined by its commanded velocity and the episode length. Terrain levels are adjusted as follows: (1) if the robot's distance exceeds 80% of the threshold, the terrain level advances to a more challenging stage; (2) if the robot's distance falls below 40% of the threshold, the terrain level reverts to an easier stage; and (3) upon completing all levels, the robot is randomly reassigned to a level to maintain diversity in training.

Domain Randomization. To increase the sim-to-real transfer ability, we follow the common strategy in robotics to use the [14]. The detailed parameters are listed in Tab 5.

Table 5. **Domain randomization parameters.**

Term	Value
Friction	$\mathcal{U}(0.6, 2.0)$
Base Mass offset	$\mathcal{U}(0.0, 3.0)$
Base CoM offset	$\mathcal{U}(-0.2, 0.2)$
Push robot–interval	8s
Push robot–max push vel_xy	0.5m/s
Motor strength range	$\mathcal{U}(0.8, 1.2)$
Delay update global steps	24×8000

Rewards. Please refer to Tab 6 for the detailed formula definitions and corresponding weights.

Termination Conditions. To maintain meaningful training and testing environments, we define termination conditions to prevent invalid episodes. An episode ends if any of the following occur: (1) *Soft pose check*: the robot's absolute roll or pitch exceeds a predefined threshold, or its height falls below a defined lower bound; (2) *Goal reach*

Table 6. **Rewards' definition and weight**.The symbol $*$ means the term only used in unified policy training stage.

Term	Mathematical Expression	Weight
Tracking Goal Velocity	$\frac{\min(\mathbf{v}_{\text{target}} \cdot \mathbf{v}_t, \text{cmd}_x)}{\text{cmd}_x + \epsilon}$	10.0
Tracking Yaw	$\exp\left(-\lvert\psi_{\text{target}} - \psi_t\rvert\right)$	0.5
Linear Velocity (Z)	v_z^2	-2.0
Angular Velocity (XY)	$\sum\left(\omega_x^2 + \omega_y^2\right)$	-1.0
Orientation	$\sum\left(g_x^2 + g_y^2\right)$	-1.0
DOF Acceleration	$\sum\left(\frac{\dot{q}_{t-1} - \dot{q}_t}{\Delta t}\right)^2$	-3.5e-8
Collision	$\sum\left(\lVert\mathbf{F}_{\text{contact}}\rVert > 0.1\right)$	-10.0
Action Rate	$\lVert\mathbf{a}_{t-1} - \mathbf{a}_t\rVert$	-0.01
Delta Torques	$\sum\left(\tau_t - \tau_{t-1}\right)^2$	-1.0e-7
Torques	$\sum \tau_t^2$	-1.0e-5
Hip Position	$\sum\left(q_{\text{hip}} - q_{\text{hip-default}}\right)^2$	-0.5
DOF Error	$r_{\text{dof\_error}} = \sum\left(q_{\text{dof}} - q_{\text{default}}\right)^2$	-0.04
Feet Stumble	$\bigvee_{(\lVert\mathbf{F}_{\text{contact}}\rVert > 4 \cdot \lvert F_{\text{contact}}\rvert)}$	-1
Feet Edge	$(\text{terrain\_level} > 3) \cdot \sum(\text{feet\_at\_edge})$	-1
Feet Air Time	$\sum\left(T_{\text{air}} - 0.5\right) \cdot (\text{first\_contact})$	1.0(H1)/0.5(G1)
Base Height	$\left(h_{\text{base}} - h_{\text{target}}\right)^2$	-100.0 (H1)/-35.0 (G1)
Point Navigation Distance*	$r_{\text{pn\_distance}} = \begin{cases} 1 & \lVert\mathbf{p}_{\text{rel}}\rVert < \theta_{reach} \\ -\lVert\mathbf{p}_{\text{rel}}\rVert \cdot 0.75 & \text{otherwise} \end{cases}$	1.0
DOF Position Limits	$\sum\left(-\max\left(0, \text{dof} - \text{dof\_lim}_{low}\right) + \max\left(0, \text{dof} - \text{dof\_lim}_{up}\right)\right)$	0.0 (H1)/-5.0 (G1)
Tracking Sigma	$\exp(-track_{err}^2/\sigma)$	0.5

Table 7. **Loss weight hyperparameters.**

Parameter	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	c_{mt}	c_{ms}
Value	1.0	1.0	1.0	1.0	1.0	1.0	100.0	2.0	0.85	0.15

check: the robot is within a specific distance from the final goal. We adopt the goal navigation criteria from [1], setting the goal distance to roughly twice the robot's body width. Specifically, the goal distance is set to 0.89 during testing and 0.5 during training to encourage precise task execution. (3) *Timeout*: The robot exceeds maximum episode length.

6.2.2. Unified Policy Training

To train the unified policy, we use the rewards listed in Tab 6, and losses introduced in the main paper, where the hyperparameters are listed in Tab 7.

7. Humanoid Hiking Benchmark

This section provides: (1) Qualitative comparisons of robot behaviors in response to varying trail challenges, demonstrating how different policy learning methodologies influence navigation and locomotion strategies tailored to humanoid tasks; (2) Detailed quantitative results for each trail type between EP-H and RMA-B, offering insights into specific strengths and weaknesses of the approaches under distinct terrain and navigation conditions.

Visualization. Fig 6 presents qualitative comparisons of LEGO-H with other benchmarked methods across five distinct trail examples, expanding on the key findings from Section 4.4 of the main paper. Additional insights include:(1) without vision, RMA-B frequently fails to adapt to changing terrain properties (e.g., slope and surface friction) and falls over more often, as observed in Fig 6(a)-(b). It also struggles to navigate obstacles effectively, often becoming stuck, as shown in Fig 6(c). The higher MEV on Ditch and Hurdle, and lower trail completion on Forest in Tab 8 also demonstrate this. (2) EP-H, which processes depth frames independently and applies brute-force cutoff for distant depth information, exhibits "circling" behaviors due to its inability to maintain scene continuity. This limitation hinders quick decision-making and recovery from self-induced distribution shifts, as demonstrated in Fig 6(b), and results in inefficient navigation paths, as illustrated in Fig 6(c). (3) While leveraging vision, RMA-H lacks dynamic adaptability in navigation due to its separation of locomotion and navigation learning. This results in inefficient behaviors on trails requiring sharp turns or obstacle avoidance, as seen in Fig 6(a)-(b). Additionally, its inefficient embodiment leads to unsafe detours, with trajectories that closely rub against obstacles, as highlighted in the zoomed-in trajectory in Fig 6(c). (4) The clean and safe-clearance trajectories of LEGO-H across all examples highlight the necessity and importance of integrative navigation and lo-

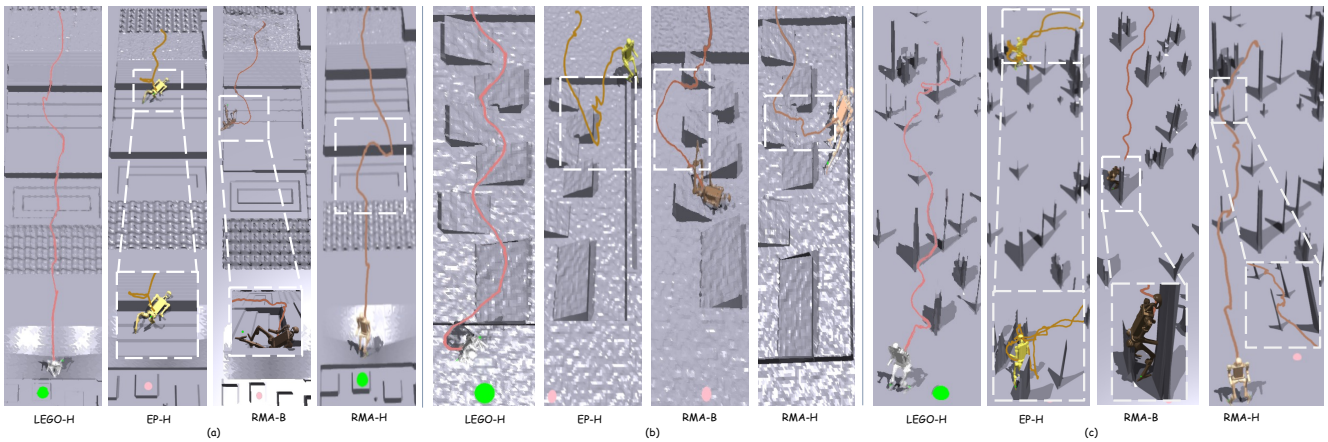

Figure 6. **Qualitative comparisons between LEGO-H and other benchmarked methods.** The trajectories, visualized through dynamically updated colored lines, depict the robots' torso position as they traverse diverse trail environments. (a) illustrates the performance on a *RandomMix* trail featuring unobstructed views with varied terrain types. (b) highlights the results on a *Ditch* trail, where uneven terrain with slopes and gaps demands quick turns and agile leaps. (c) showcases the performance on a *Forest* trail, where extensive obstacles of different sizes and heights block the robot's view. The zoom-in regions highlight the issues of the robots.

Table 8. **EP-H vs RMA-B on each trail category.** This table employs a distinct protocol for fine-grained analysis: 256 randomly initialized robots are evaluated for 30 seconds *per* trail category, spanning 25 scenes (5 difficulty levels, each with 5 variants). Results are averaged over 5 runs to minimize random biases and ensure robustness.

Methods	Success Rate (%) ↑	Trail Completion (%) ↑	Traverse Rate (%) ↑	MEV (%) ↓	TTF (s) ↑	Time-to-Reach (s) ↓
RandomMix						
EP-H	16.98 ± 0.85	2.67 ± 0.14	70.88 ± 1.41	11.32 ± 1.83	3.33 ± 0.13	9.73 ± 0.19
RMA-B	30.99 ± 0.95	3.60 ± 0.37	76.74 ± 1.13	10.95 ± 1.70	4.14 ± 0.20	6.79 ± 0.09
Ditch						
EP-H	16.12 ± 0.66	17.90 ± 0.62	55.75 ± 0.58	22.75 ± 1.63	3.50 ± 0.08	11.88 ± 0.33
RMA-B	32.80 ± 1.56	30.77 ± 0.59	63.49 ± 1.42	23.66 ± 1.63	4.56 ± 0.18	6.37 ± 0.37
Hurdle						
EP-H	46.54 ± 2.64	57.04 ± 1.25	68.95 ± 1.79	8.77 ± 0.46	6.44 ± 0.21	5.94 ± 0.14
RMA-B	83.04 ± 0.27	76.72 ± 0.47	83.04 ± 1.17	12.90 ± 1.92	9.24 ± 0.28	4.22 ± 0.04
Gap						
EP-H	18.13 ± 1.19	32.26 ± 0.48	58.74 ± 1.21	31.84 ± 2.00	4.36 ± 0.20	12.15 ± 0.34
RMA-B	39.93 ± 1.55	44.27 ± 1.06	65.30 ± 1.32	24.10 ± 2.09	5.44 ± 0.24	7.99 ± 0.17
Forest						
EP-H	63.29 ± 1.50	1.04 ± 0.16	82.61 ± 1.18	6.18 ± 1.57	8.96 ± 0.49	13.65 ± 0.08
RMA-B	64.81 ± 2.43	1.86 ± 0.38	81.59 ± 3.20	5.69 ± 1.04	10.18 ± 0.89	13.20 ± 0.24

comotion development through unified learning.

Insufficient Vision vs Blind. Tab 8 show the comparison between EP-H and RMA-B. It indicates insufficient vision sometimes worse than blind vision.

8. Discussion

Future work. (1) Kilometer-scale hiking. In this paper, we investigate humanoid robots on prototype trails to establish a baseline on the importance of integrative high-level navigation and low-level motor skills. However, real-world trails are considerably more complex, with long-distance traverse challenges. Future work could expand the framework to handle kilometer-scale trails, where sustained adaptability, energy efficiency, and long-term planning become crucial. (2) Whole-body control for integrative navigation and locomotion skills. Expanding control across the entire body would enable a wider spectrum and adaptive behaviors, enhancing the robot's flexibility in complex, obstacle-rich environments. Our preliminary results suggest that while robots exhibit distinct motor styles based on physical constraints(Fig. 7), *direct* involvement of the upper body does not significantly impact performance. This opens opportunities for future work on exploring how coordinated whole-body strategies can enhance performance. (3) Sim-

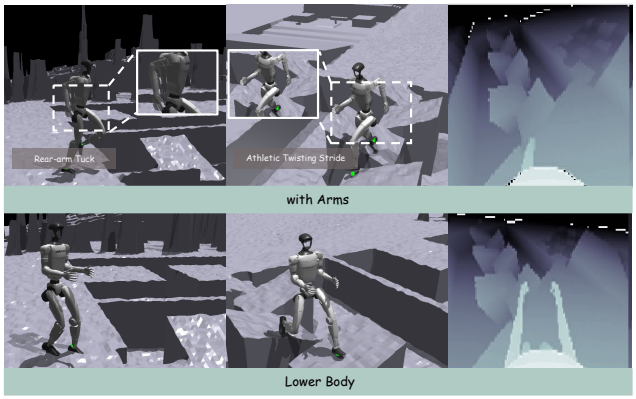

Figure 7. **Preliminary observations for future work on WBC.** G1 exhibits distinct motor behaviors over *with arms* vs *only lower body*. Besides, G1 emerges a rear-arm tuck posture while walking, likely to minimize arm interference with vision (see depth map).

ulated environment upgrading. Our current simulated trails are primarily for foot contact; Future work could upgrade the simulated environment to better incorporate whole-body interactions, enabling a better testbed for future hiking studies. (4) Real-world deployment. In this paper, we conduct experiments on the simulator, enabling controlled benchmarking, rapid iteration, and reproducibility — *key prerequisites* for real-world deployment. However, applying LEGO-H to real-world scenarios remains a vital next step toward closing the sim-to-real gap and realizing field-ready humanoid hikers.

References

[1] Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. *CoRR*, abs/1807.06757, 2018. 7

[2] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *CoRL*, 2019. 2

[3] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *IEEE International Conference on Robotics and Automation, ICRA*, 2024. 6

[4] Peter Dayan and Geoffrey E. Hinton. Feudal reinforcement learning. In *Advances in Neural Information Processing Systems 5, [NIPS Conference*, 1992. 2

[5] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021. 2

[6] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Sci. Robotics*, 2020. 2, 6

[7] Joonho Lee, Marko Bjelonic, Alexander Reske, Lorenz Wellhausen, Takahiro Miki, and Marco Hutter. Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Sci. Robotics*, 2024. 2, 6

[8] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In *NeurIPS Datasets and Benchmarks*, 2021. 5

[9] Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *NeurIPS*, 2018. 2

[10] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 3

[11] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Trans. Graph.*, 2022. 2

[12] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Learning humanoid locomotion with transformers. *CoRR*, abs/2303.03381, 2023. 2

[13] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 1999. 2

[14] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2017. 6

[15] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: similarity control and knowledge transfer. *J. Mach. Learn. Res.*, 2015. 2

[16] Qi Wu, Zipeng Fu, Xuxin Cheng, Xiaolong Wang, and Chelsea Finn. Helpful doggybot: Open-world object fetching using legged robots and vision-language models. In *arXiv*, 2024. 6

[17] Kuo-Hao Zeng, Zichen Zhang, Kiana Ehsani, Rose Hendrix, Jordi Salvador, Alvaro Herrasti, Ross B. Girshick, Aniruddha Kembhavi, and Luca Weihs. Poliformer: Scaling on-policy RL with transformers results in masterful navigators. In *CoRL*, 2024. 2