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.

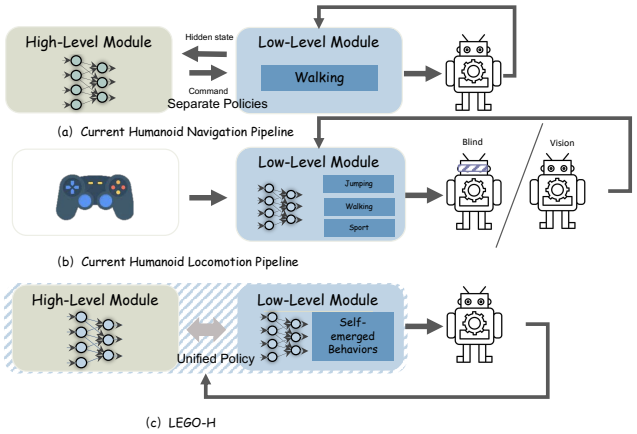

Figure 1. **The conceptual framework differences.** We summarize the key conceptual level differences between our work and current humanoid robot trends for better positioning of LEGO-H.

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. Flexibility and Efficiency of TC-ViTs

In this subsection, we further analyze the dynamic adjustment capability of TC-ViTs for near-goal prediction, and the efficiency behind its recurrent goal adaptation module design.

Dynamic near-goal adjustments. As discussed in Section 3.4 of the main paper, TC-ViTs does not provide a fixed trajectory that the locomotion module must rigidly follow. Instead, it predicts several near-future goals, dynamically adjusting them based on the robot's current state. Only the nearest goal is passed to the locomotion module as soft guidance, preventing long-term error accumulation in navigation decisions, and enabling flexibility and adaptability

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

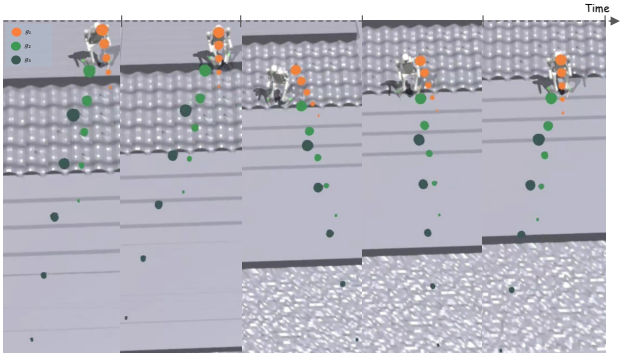

Figure 2. **Dynamic adjustments of near goal predictions.** Snapshots from left to right show a robot traversing mixed terrains along a trail. The predicted near goals g_1, g_2, g_3 dynamically adapt to the robot's current state, reflecting real-time adjustments to its navigation decisions. Bubble size represents the predicted local navigation direction (from large to small).

in response to changing environments. Here, as a complementary, in Fig 2, we illustrate how these goals dynamically adjust as the robot progresses through a trail with mixed terrains, demonstrating the TC-ViTs' responsiveness.

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 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.

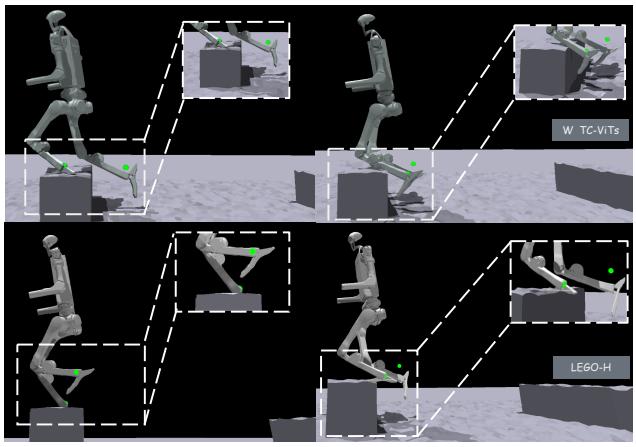

Figure 3. **Qualitative ablation on with/without HLM.** Snapshots from right to left depict two time steps of a robot traversing a hurdle obstacle. The top row illustrates behaviors without HLM, where unsafe movements lead to right leg collisions with the hurdle. The bottom row showcases behaviors with HLM, exhibiting coordinated and structurally rational actions that ensure stability and successful traversal with safe clearance.

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 3, 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

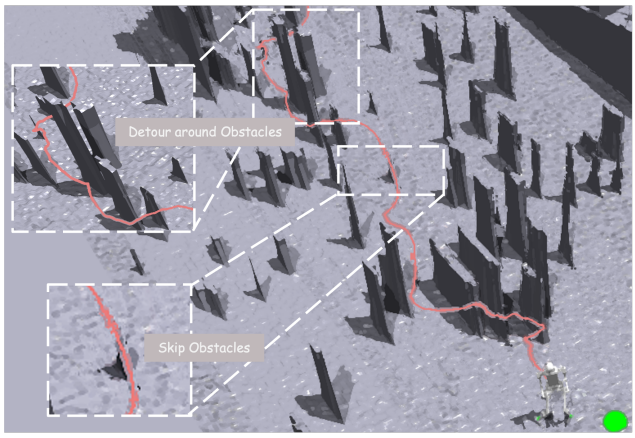

Figure 4. **Navigation in blocked paths over different obstacles.** The colored trajectory illustrates the robot's torso position as it traverses the trail. Zoomed-in regions highlight distinct navigation behaviors: when encountering crowded, tall obstacles, the robot opts to detour, whereas for smaller obstacles, the robot leaps over, demonstrating adaptive navigation strategies.

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.

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 4, the traversed 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.

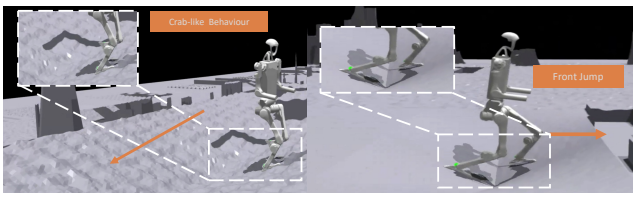

Figure 5. **Behaviors over difference terrains.** The robots exhibit diverse integrative navigation and locomotion skills tailored to varying trail terrains. (a) The robot adopts a lateral "crab" walking style to navigate a long, rugged gully, maintaining stability while progressing toward the hiking terminus. (b) The robot faces the final terminus directly and jumps over a short, smooth gully. The orange directional lines show the terminus directions.

Behavior over difference 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 5: (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 6, 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

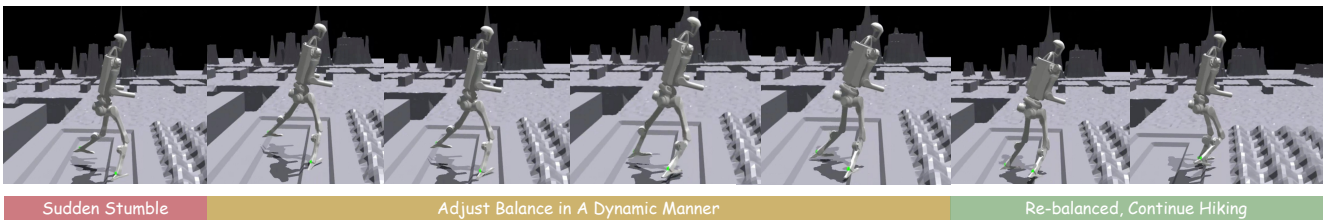

Figure 6. **Self re-balance.** The robot stumbles unexpectedly (red timeline), swiftly adjusts its balance through a sequence of emergent lateral motions (yellow timeline), and seamlessly regains stability (green timeline).

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 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 encourage 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 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 7 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 7(a)-(b). It also struggles to navigate obstacles effectively, often becoming stuck, as shown in Fig 7(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 7(b), and results in inefficient navigation paths, as illustrated in Fig 7(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 7(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 7(c). (4) The clean and safe-clearance trajectories of LEGO-H across all examples highlight the necessity and importance of integrative navigation and locomotion 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

Release and maintains. All code/models/benchmarks will be publicly accessible and continuously updated to incorporate more robots/environments/models, aiming to establish a standard evaluation testbed for humanoid hiking research.

Future work. (1) Kilometer-scale hiking. In this paper,

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

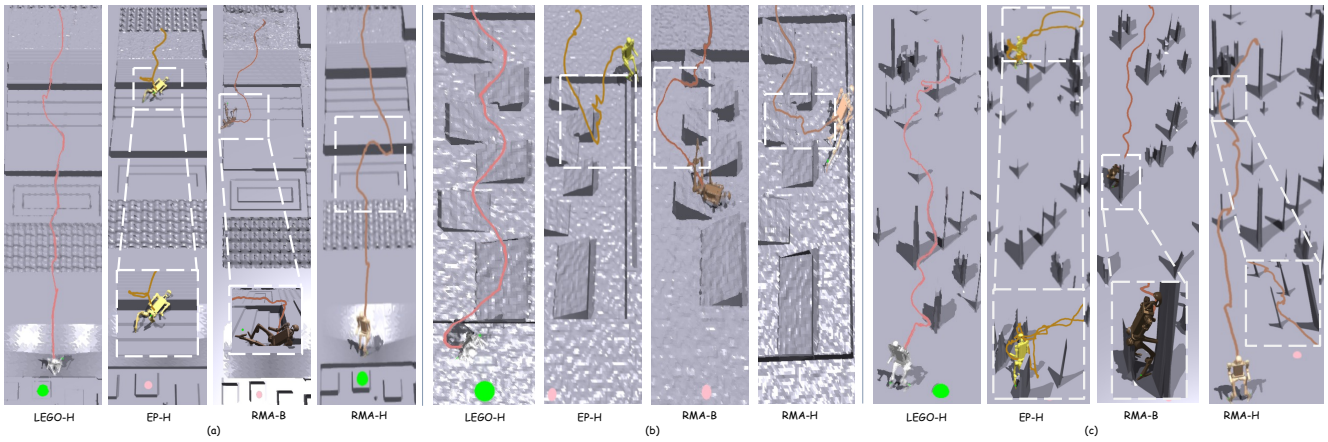

Figure 7. **Qualitative comparisons between LEGO-H and other benchmarked methods.** The trajectories, visualized through dynamically updated colored lines, depict the robots' torso position as they traverse diverse trail environments. (a) illustrates the performance on a *RandomMix* trail featuring unobstructed 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 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

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

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

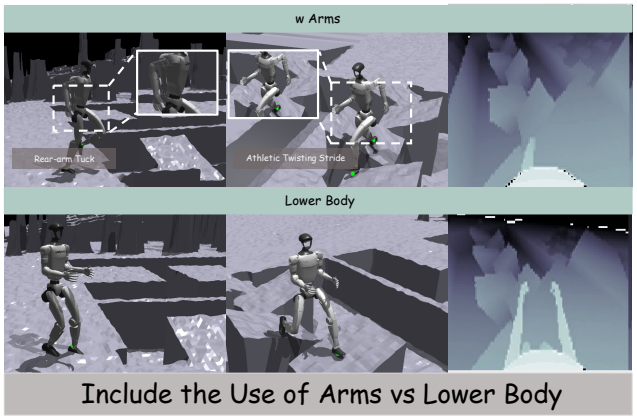

Figure 8. **Preliminary observations for future work on WBC.** G1 exhibits distinct motor behaviors over w arms vs only lower body. Besides, G1 emerges a rear-arm tuck posture while walking, likely to minimize arm interference with vision (see depth map).

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. 8), *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) Simulated 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. 7

[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, 7

[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, 7

[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. 3

[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. 7

[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. 7

[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. 3