Locomotion Beyond Feet

Tae Hoon Yang, Jiacheng Hu, Haochen Shi, Zhicong Zhang, Daniel Jiang, Weizhuo Wang,
Yao He, Zhen Wu, Yifan Hou, Monroe Kennedy, Shuran Song, and Karen Liu

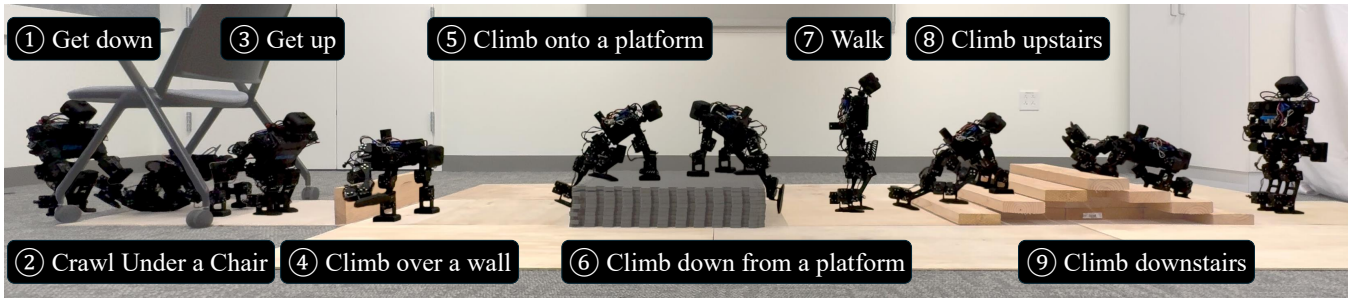


Fig. 1: **Locomotion Beyond Feet** enables whole-body humanoid locomotion on diverse and challenging terrains—including low-clearance spaces under chairs, knee-high walls, knee-high platforms, and steep ascending and descending stairs—through chaining nine distinct locomotion skills that actively engage body parts beyond the legs for stability and support.

***Abstract*— Most locomotion methods for humanoid robots focus on leg-based gaits, yet natural bipeds frequently rely on hands, knees, and elbows to establish additional contacts for stability and support in complex environments. This paper introduces Locomotion Beyond Feet, a comprehensive system for whole-body humanoid locomotion across extremely challenging terrains, including low-clearance spaces under chairs, knee-high walls, knee-high platforms, and steep ascending and descending stairs. Our approach addresses two key challenges: contact-rich motion planning and generalization across diverse terrains. To this end, we combine physics-grounded keyframe animation with reinforcement learning. Keyframes encode human knowledge of motor skills, are embodiment-specific, and can be readily validated in simulation or on hardware, while reinforcement learning transforms these references into robust, physically accurate motions. We further employ a hierarchical framework consisting of terrain-specific motion-tracking policies, failure recovery mechanisms, and a vision-based skill planner. Real-world experiments demonstrate that Locomotion Beyond Feet achieves robust whole-body locomotion and generalizes across obstacle sizes, obstacle instances, and terrain sequences.**

I. Introduction

Most locomotion methods for humanoid robots focus solely on leg-based movement [1], [2], [3], yet bipeds in nature frequently leverage contacts from all limbs and torso to stabilize and support their bodies in complex environments [4], [5]. For example, in environments such as low-clearance spaces under chairs, knee-high platforms, knee-high walls, and steep ascending and descending stairs, locomotion using only the feet becomes infeasible or necessitates abrupt motions. Humans naturally leverage additional body parts—such as hands, knees, and elbows—to establish extra contact points, enabling them to crawl, climb, and employ other whole-body strategies to overcome these obstacles.

We introduce a vision-based, hierarchical policy framework to enable highly diverse whole-body humanoid locomotion. Despite the benefits, whole-body locomotion remains underexplored in humanoid robots due to two main challenges: (1) Navigating complex environments requires strategic contact planning and robust control. (2) Different terrains require fundamentally different motor skills, such as walking, climbing, or crawling.

To address the first challenge, a key insight is that traditional keyframe animation and reinforcement learning (RL) are highly complementary for learning terrain traversal policies. Keyframe animation provides an intuitive approach to encode human knowledge of motor skills and physical interactions with the environment into robot control, such as specifying critical contact states and joint configurations [6]. Because natural human motion is typically low-frequency [7], keyframes serve as an effective abstraction.

Similar to prior approaches that retarget human motion capture (mocap) trajectories for motion tracking [8], [9], [10], [11], [12], [13], traditional keyframe animation provides kinematics but relies on RL trained in physics simulation to become dynamically viable robot policies. Importantly, unlike motion capture data, keyframes bypass the embodiment gap entirely by directly designing reference motions in the robot's state space. This frees us from carefully matching human and robot embodiments, and instead allows exploration of the robot's full hardware capabilities, producing motions not constrained by human demonstrations. Furthermore, the physical plausibility of keyframes can be verified in simulation and validated in the real world through open-loop execution, significantly accelerating the design iterations. In practice, once familiar with the tools, designing a physically consistent trajectory with keyframes typically requires only a few hours, even for challenging locomotion skills such as climbing over a wall, considerably more efficient than combined efforts of motion capture, human motion data retrargeting, and extensive reward shaping for RL training.

To address the second challenge that different terrain

requires different skills, we argue that a single vision-based policy is not necessary and likely less desirable: a hierarchical framework is more robust. Our hierarchical framework allows diverse motion tracking policies tailored to distinct terrains, robust failure recovery mechanisms for fall situations, and a general vision-based planner that classifies terrain with stereo fisheye cameras and learned depth estimation. While rapid responses to local disturbances require fast 50 Hz control loops, locomotion mode selection with vision input can robustly operate at a lower frequency (10 Hz).

As shown in Figure 1, Locomotion Beyond Feet is a comprehensive framework that enables traversal of extremely challenging obstacle courses through three categories of motor skills: (1) locomotion skills such as walking and crawling, (2) transition skills such as getting up from crawling, getting down to crawling, getting up from prone, and getting up from supine, and (3) terrain-specific skills such as climbing onto a platform, rotating on a platform, climbing down from a platform, climbing over a wall, climbing upstairs, and climbing downstairs. Extensive real-world experiments demonstrate the system's robustness to obstacle sizes, obstacle instances, and terrain sequences. All work will be open-sourced.

II. Related Works

A. Locomotion on Challenging Terrains

Biomechanical studies reveal clear distinctions between quadrupedal and bipedal locomotion modes: macaques walking bipedally adopt a wider step width, longer duty cycle, and extended double-support phase to compensate for upright posture and a shifted center of mass [14]. The key distinction lies in the size of the support polygon, defined as the convex hull of ground contact patches. Static balance is possible only when the center of mass (CoM) remains within this polygon. Leg-only locomotion yields a small support polygon that reduces to a single contact patch during footstep transitions, often necessitating abrupt motions on difficult terrain. In contrast, whole-body locomotion can employ three or more contact patches to form a larger and more consistent support polygon, yielding more stable and safer movement.

On the robotics side, RL has enabled robust locomotion on challenging terrains; for instance, Rudin et al. [15] demonstrated massively parallel deep RL. However, the same strategy is applied to both quadrupedal and bipedal robots without accounting for their distinct locomotion modes. Quadrupedal robots excel at terrain traversal through coordinated leg motion and have achieved agile parkour behaviors [16], [17], [18], [19], [20], [21]. Recent humanoid approaches [1], [2] largely adopt quadruped-inspired strategies, relying primarily on leg-based locomotion with minimal arm involvement and overlooking the distinct roles of arms and legs. By contrast, our approach leverages whole-body motion, with all the body parts actively contributing to stability during terrain traversal, akin to natural human strategies in extreme environments.

B. Keyframe Motion in Robotics

Keyframes provide intuitive human control over motion synthesis, originating from character animation [22], [23], [24]. In robotics, keyframes have been adapted for humanoid motion generation through optimization [25], as reference trajectories for learning motion tracking policies [26], [27], and as sparse rewards to achieve specific goals at predetermined times [28]. In these works, keyframes offer an intuitive mechanism for encoding human expertise and biomechanical insights into robotic motion synthesis [6]. The effectiveness of keyframe representation stems from the observation that natural human locomotion exhibits predominantly low-frequency characteristics [7], making sparse temporal sampling through keyframes a well-suited abstraction that captures essential motion dynamics. Inspired by prior work, we leverage keyframe motions as references for training terrain-specific whole-body locomotion skills.

Unlike traditional keyframe animation, our approach ensures physics-grounded motions through a MuJoCo-integrated tool [29] that allows interactive visualization and validation of dynamics and contacts. Keyframing further distills human knowledge of dynamics into motion priors by explicitly specifying contact transitions and support phases. In contrast, recent kinematic retargeting methods [8], [9], [10], [11], [12], [13] lack dynamics information [30] and suffer from embodiment gaps. Moreover, while retargeting pipelines make RL training and sim-to-real transfer difficult to verify, keyframes can be validated directly in simulation or via open-loop execution on the real robot, enabling faster iteration and clearer optimization.

C. Perception for Legged Locomotion.

Legged locomotion has employed diverse perception modalities, using lidar point clouds for geometric understanding [21], [31] and depth sensing for terrain perception [2], [18], [20]. Our approach adopts depth-based perception for its accessibility and computational efficiency. Notably, advances in learned stereo models such as Foundation Stereo [32] enable depth estimation directly from RGB inputs, even eliminating the need for depth sensors.

D. Policy Chaining

Policy chaining is typically achieved by treating motor controllers as modular skills and composing them through a high-level planner that determines when each controller is activated. The main challenge lies in the mismatch between the terminal state distribution of one policy and the start state distribution of the next. Prior methods address this either by carefully engineering compatible start and terminal states [33] or by leveraging deep learning frameworks to train composite behaviors with a meta-composer policy [34], [35], [36]. In our work, we found that a carefully designed state machine, in which all policies are trained to start and end in one of four canonical poses—standing, crawling, lying prone, or lying supine—was sufficient to ensure smooth transitions between skills during execution.

III. Method

Locomotion Beyond Feet enables whole-body locomotion through four key components: physics-grounded keyframe

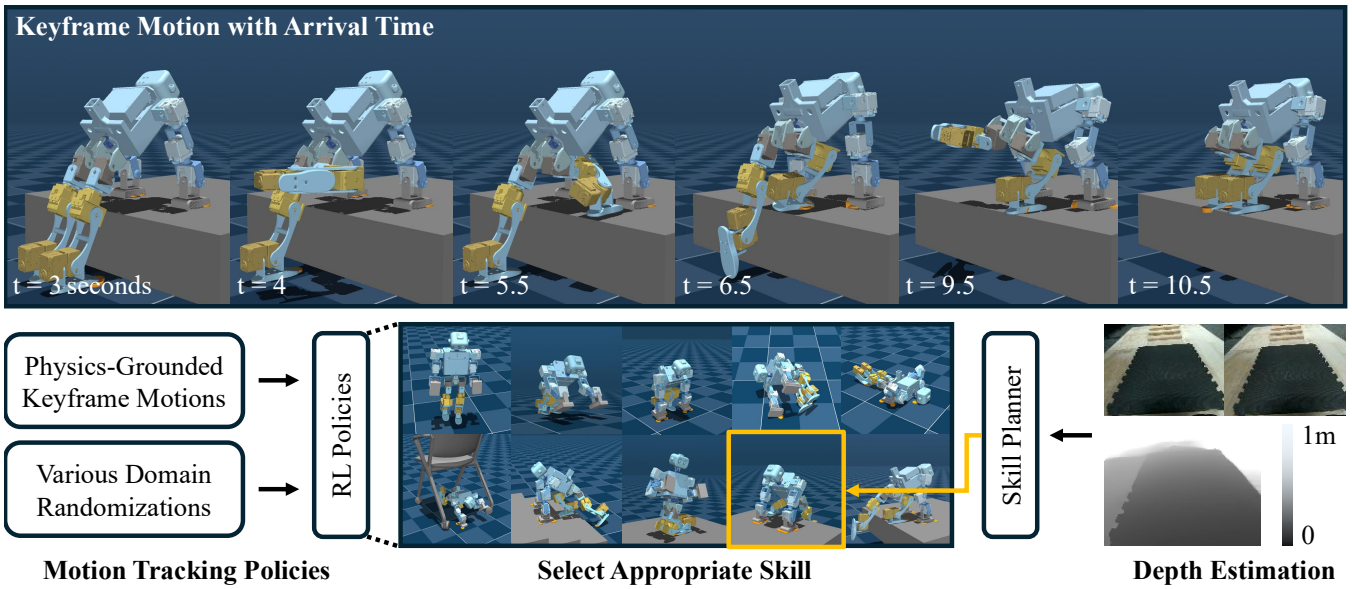


Fig. 2: **System Pipeline.** First, we generate physics-grounded keyframe motions with a physics-aware GUI application, where robot poses and arrival times are specified interactively. Second, we interpolate the keyframes to create reference motions, which serve as tracking rewards for RL policies. We further apply extensive domain randomization, such as initial robot states, obstacle dimensions, and IMU noise. Finally, a skill planner processes depth input from a learned depth estimation module at 10 Hz, along with IMU readings and the current skill, to select the next appropriate skill.

motion generation, DeepMimic-based motion tracking policies, a depth-conditioned visual skill classifier, and a hierarchical skill execution framework (Figure 2).

A. Physics-Grounded Keyframe Motion

We generate reference motions using a GUI tool based on MuJoCo [29] that allows intuitive design of physically plausible motions. In the app, the user specifies robot poses along with their execution order and arrival times. The resulting keyframe sequence is then linearly interpolated to generate a complete trajectory. Although specifying arrival times may seem difficult, we found that simple choices such as 0.5 seconds, 1 second, or 2 seconds are usually sufficient.

Keyframe motion is most criticized for the need for manual tuning [37]. To mitigate this, we streamline keyframe design with utilities for joint mirroring, aligning the robot's feet to the ground, and visualization of the center of mass, collisions, and contacts. Our tool enables quick validation of individual keyframes and full trajectories for balance and smoothness. In practice, for simple motions such as crawling, we design the entire trajectory to be physically valid and directly replayable in simulation. For more challenging motions such as climbing over a wall, we instead ensure that individual keyframes are statically stable, so that the linearly interpolated trajectory remains physically plausible.

Another limitation of keyframe motion is its open-loop nature—it cannot adapt to perturbations, modeling errors, or unexpected environmental changes. While it provides a strong prior, it lacks the reactive flexibility required for real-world deployment. To address this, we train motion-tracking policies with RL that robustly execute keyframe motions while adapting to various uncertainties.

B. Motion Tracking Policies

We categorize motion tracking policies into three types:

Locomotion skills provide continuous control for periodic locomotion skills that can be modulated by velocity commands. We implement command-conditioned policies for walking and crawling, ensuring reactive control based on high-level navigation commands.

Transition skills handle transitions between different poses, such as standing to crawling, crawling to standing, lying prone to standing, and lying supine to standing. Each policy is trained to execute the entire transition sequence autonomously once initiated.

Terrain skills handle specific terrains, including climbing onto a platform, rotating on a platform, climbing down from a platform, climbing over a wall, climbing upstairs, and climbing downstairs. Each policy autonomously executes the entire trajectory once triggered.

We train all three types of skills following a similar recipe: RL-based motion tracking policies $\pi(\mathbf{a}_t|\mathbf{s}_t)$ that output joint position setpoints $\mathbf{a}_t$ for proportional-derivative (PD) controllers. The observable state $\mathbf{s}_t$ includes:

$$\mathbf{s}_t = (\phi_t, \mathbf{c}_t, \Delta\mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}, \boldsymbol{\theta}_t, \boldsymbol{\omega}_t)\,, \tag{1}$$

where ϕ_t denotes the phase signal for temporal coordination. The locomotion policies employ a periodic signal, whereas the transition and terrain policies use a monotonically increasing signal. $\boldsymbol{c}_t$ represents optional velocity commands, $\Delta\boldsymbol{q}_t$ is the joint position offset from the neutral pose $\boldsymbol{q}_0$, $\dot{\boldsymbol{q}}_t$ is the joint velocity, $\boldsymbol{a}_{t-1}$ is the previous action, $\boldsymbol{\theta}_t$ is the torso orientation, and $\boldsymbol{\omega}_t$ is the torso angular velocity.

We train these policies using PPO [38] with reward

Algorithm 1 Skill Planner

Require: depth map $\boldsymbol{D}$; torso pitch and bounds $\theta, \theta_{\min}, \theta_{\max}$; depth extrema bounds $\delta_{\min}$, $\delta_{\max}$; confidence threshold c; current skill S_{curr}; recovery skill S_{rec}.

function SELECTSKILL($\boldsymbol{D}$, S_{curr}, θ)
  fallen $\leftarrow \big(\theta_{\text{pitch}} < \theta_{\min}$ **and** $\max(\boldsymbol{D}) < \delta_{\min}\big)$
    or $\big(\theta_{\text{pitch}} > \theta_{\max}$ **and** $\min(\boldsymbol{D}) > \delta_{\max}\big)$
  if fallen **then**
    return S_{rec}
  end if
  $\boldsymbol{p} \leftarrow$ CLASSIFYSKILL($\boldsymbol{D}$)
  $\bar{\boldsymbol{p}} \leftarrow 0.1 \cdot \bar{\boldsymbol{p}} + 0.9 \cdot \boldsymbol{p}$
  $S_{\text{best}} \leftarrow \arg\max_j \bar{\boldsymbol{p}}[j]$
  if $\bar{\boldsymbol{p}}[S_{\text{best}}] > c$ **then**
    return S_{best}
  else
    return S_{curr}
  end if
end function

functions following standard practices [26]:

$$\mathrm{r}_t = \mathrm{r}_t^{\text{imitation}} + \mathrm{r}_t^{\text{regularization}} + \mathrm{r}_t^{\text{survival}}. \tag{2}$$

The imitation reward $\mathrm{r}_t^{\text{imitation}}$ enforces accurate tracking of reference motions generated from our keyframe interpolation, with the exception of walking motions which use a closed-form Zero Moment Point (ZMP) solution [39]. The regularization term $\mathrm{r}_t^{\text{regularization}}$ incorporates heuristics to minimize joint torques, energy consumption, and action rate, while the survival reward $\mathrm{r}_t^{\text{survival}}$ prevents early termination.

The term $\mathrm{r}_t^{\text{imitation}}$ is defined as a weighted sum across several tracking rewards. We follow similar formulation conventions in DeepMimic [26]: the pose reward r_t^p encourages alignment of body orientations with the reference motion, the velocity reward r_t^v matches local body velocities, the end-effector reward r_t^e tracks the positions of the hands and feet, and the center-of-mass reward r_t^c penalizes deviations of the robot's center of mass from the reference trajectory.

$$\mathrm{r}_t^{\text{imitation}} = \mathrm{w}_t^m \mathrm{r}_t^m + \mathrm{w}_t^p \mathrm{r}_t^p + \mathrm{w}_t^v \mathrm{r}_t^v + \mathrm{w}_t^e \mathrm{r}_t^e + \mathrm{w}_t^c \mathrm{r}_t^c. \tag{3}$$

More specifically, r_t^m is the motor position tracking reward:

$$\mathrm{r}_t^m = \sum_g \mathrm{w}_g \exp\left(-\|q_g - q_g^*\|^2\right), \tag{4}$$

where $g \in \{\text{leg, arm, neck, waist}\}$, q_g is the actual motor position, q_g^* is the corresponding reference motor position, and w_g is the weight assigned to action group g. For physically verified keyframe motion, we use the commanded actions as the motor position reference to account for motor tracking errors. Thanks to physics-grounded keyframe motions, RL training requires no additional rewards and proceeds smoothly.

To enable seamless sim-to-real transfer, we employ extensive domain randomization during training, including ground friction, motor actuation parameters, initial robot states, starting positions and orientations to different terrains, IMU noise, and action delays. The IMU noise model combines colored noise, white noise, random-walk bias, and random amplitude scaling for both gyroscope and orientation signals, mimicking realistic IMU outputs.

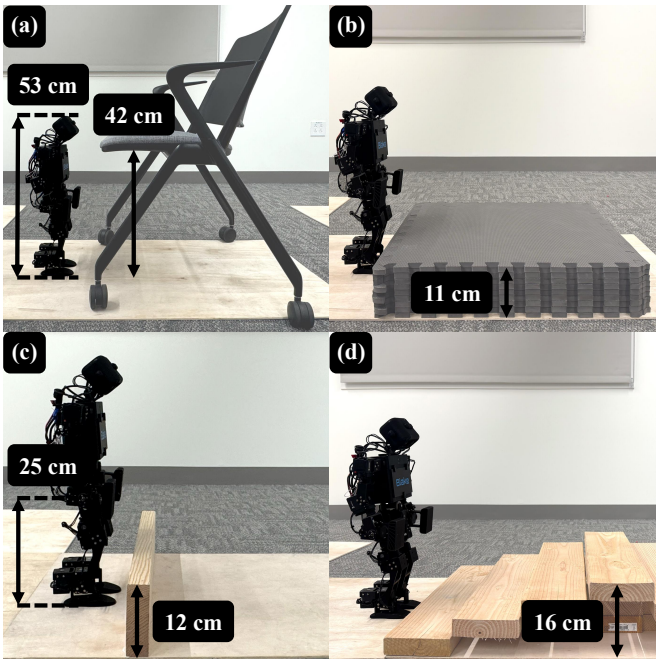


Fig. 3: **Test Obstacles**. We show the robot beside test obstacles, including (a) low-clearance spaces under chairs, (b) knee-high platforms, (c) knee-high walls, and (d) steep ascending and descending stairs. The space under the chairs is shorter than the robot (53 cm), requiring crawling. The wall is 48% of the robot's leg length (25 cm), requiring climbing. The platform height is 44% of the leg length, and each stair height is 16% of the leg length, all posing extreme challenges at the robot's scale.

All our polices are trained to start and end in either a standing pose, a crawling pose, a lying prone pose, or a lying supine pose. This design choice facilitates smooth transitions between different skills during execution.

C. Visual Skill Classifier

Our classifier enables autonomous skill selection by learning to classify appropriate skills from depth input.

Data Collection. Training data are first generated in simulation by pairing depth maps with skill labels. Obstacles are randomly positioned to create diverse terrains, and camera poses are slightly randomized within each rollout to increase diversity. Depth maps are primarily captured from head-mounted cameras, ensuring natural viewpoints consistent with real-world deployment. Locomotion skill data are collected throughout execution, while transition and terrain skill data are sampled only at the start of each skill. Ground-truth labels are obtained from a distance-based heuristic planner that triggers skills when obstacles fall within predefined thresholds. To address the remaining sim-to-real gap, we additionally collect a small amount of real-world data to

Fig. 4: **Terrain Policies.** We demonstrate our policies on traversing extremely challenging terrains—including (a) low-clearance spaces under chairs, (b) knee-high walls, (c) knee-high platforms, and (d) steep ascending and descending stairs—and additionally show (e) fall recovery from supine and prone positions in case of failure.

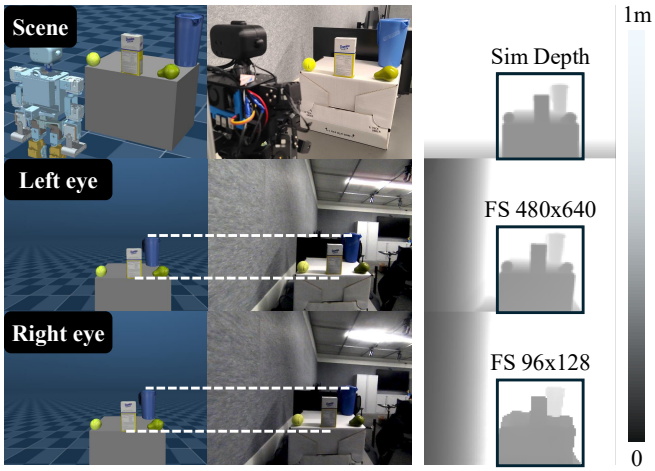
finetune the classifier, ensuring alignment with depth from real sensors.

Skill Classifier Training. We train a ResNet [40] classifier to select appropriate skills from depth input. To address class imbalance, transition and terrain skills are weighted proportionally, since locomotion data are more abundant. Simulated depth maps are processed with downsampling, cropping, clipping, noise, and blur to match the real camera. Unlike RGB-D sensors, our learning-based stereo system introduces distinct noise characteristics: although most depth values remain temporally consistent in a static scene, we observe both local and global flickering over time. We replicate this phenomenon in the simulator with local and global Gaussian noise and Gaussian blur at edges.

Real-world Deployment. We estimate depth with Foundation Stereo [32] from rectified dual-fisheye RGB images. Its disparity maps are converted to depth using rectification parameters and the baseline. Compared to conventional depth cameras, Foundation Stereo offers more accurate metric estimation and a much wider field of view—$97^\circ \times 76^\circ$ in our setup versus $87^\circ \times 58^\circ$ for a RealSense D435.

D. Hierarchical Policy Execution

We introduce a hierarchical framework that separates vision-based planning from proprioception-based control, enabling modularity and robustness. The design employs a



(a) Simulation (b) Real world (c) Depth map

Fig. 5: **Sim-to-real Depth Comparison**. We set up the same scene of YCB objects [41] (a) in MuJoCo [29] and (b) in the real world. The real-world RGB images are rectified after calibrating the fisheye cameras' intrinsics and distortion, with white dashed lines illustrating proper alignment. (c) On the right is a comparison of ground-truth depth with real-world estimates from Foundation Stereo [32] with resolution 480×640 and 96×128, respectively. We compute the quantitative results in the cropped region marked by the black box.

low-frequency visual classifier (10 Hz), motivated by the observation that locomotion mode switching occurs at low frequency, coupled with a high-frequency low-level policy (50 Hz) that enables rapid responses to local disturbances. Separate from the vision-based planner, our framework also detects falls from IMU readings and triggers the recovery policies, further enhancing system robustness.

Our execution strategy is shown in Algorithm 1: during testing, depth maps are continuously processed and skills are predicted at 10 Hz. For smooth deployment, skill predictions are temporally stabilized with an exponential moving average. The system continues executing locomotion skills (walking or crawling) until the smoothed confidence surpasses a threshold, at which point it switches to the corresponding transition or terrain policy. Some skills are chained, such as rotating on a platform and climbing down from a platform. During a transition, classifier outputs are ignored; once it completes, the system resumes locomotion control until the next confident transition is triggered.

IV. Experiments

A. Setup

We use the open-source humanoid platform ToddlerBot [42] for these whole-body locomotion tasks. With a compact form factor, 30 degrees of freedom, and human-like range of motion, ToddlerBot is well-suited for testing in complex environments consisting of diverse obstacles to evaluate terrain traversal skills (Figure 3). These terrains are constructed using common household items such as chairs, foam blocks, and wooden planks, allowing for easy setup and reconfiguration. Foam blocks and wooden planks are modeled with simple geometric primitives, while the chair's geometry is captured using an image-to-mesh generator [43] with hand-measured scales for accurate representation. We conduct controlled experiments on individual components as well as a holistic evaluation of the entire system.

B. Motion Tracking Policies

Figure 4 presents a selection of real-world photos, showcasing the effectiveness of our motion-tracking policies with zero-shot sim-to-real transfer, including crawling under a chair, climbing over a wall, climbing onto a platform, rotating on a platform, climbing down from a platform, climbing upstairs, climbing downstairs, getting up from supine, and getting up from prone. The full experiment, including an additional cart-exit skill acquired with only a few hours of keyframe tuning, is shown in the supplementary video.

C. Depth Estimation

To improve the inference speed, we downsample the RGB resolution from 480×640 to 96×128 and compile the model using TensorRT with float16 precision, achieving $10\times$ speedup from 1 Hz to 10 Hz on a Jetson Orin NX 16GB.

To demonstrate the effectiveness of depth estimation with Foundation Stereo [32], we present a qualitative comparison in Figure 5. Note that while many details are lost in the 96×128 version, they are unnecessary for terrain perception, as the obstacles are typically large structures. Quantitatively, our evaluation reports a pixel-wise mean absolute error (MAE) of 55 mm within the black box region, as shown in Figure 5 and a point cloud Chamfer Distance of 16 mm, which primarily arises from misalignment between the simulation and the real-world scene setup. Moreover, the accuracy at 96×128 is comparable to 480×640, with a pixel-wise MAE of 57 mm within the black box region in Figure 5 and a point cloud Chamfer Distance of 12 mm compared to the ground truth in simulation. Empirically, the accuracy at 96×128 is sufficient for the downstream visual skill classifier.

D. Visual Skill Classifier

We train primarily on 64,665 simulated depth map–skill pairs, supplemented with 9,952 real-world pairs, each labeled by obstacle distances using the same procedure as in simulation. To evaluate robustness, we collect an additional 2,401 real-world depth maps as a held-out test set.

Figure 6 shows the confusion matrix on the real-world test set, and Table I reports accuracy across different training regimes. A classifier trained only in simulation suffers from the sim-to-real gap, while one trained solely on real data performs well but is costly to collect. Combining large-scale simulation with limited real data yields the highest accuracy with minimum data collection effort. Although the best model still incurs a 3.3% error rate, this is further mitigated by the downstream skill planner by temporal smoothing and confidence-based skill selection (Algorithm 1).

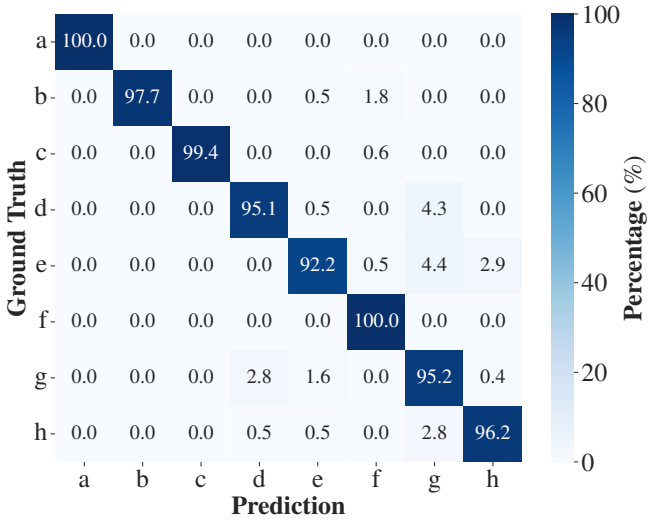


Fig. 6: **Visual Skill Classifier Accuracy.** We present a confusion matrix of the skill classifier's real-world accuracy after training in simulation and fine-tuning with a small amount of real-world data. The skills are marked with letters: (a) getting down to crawling, (b) crawling under a chair, (c) getting up from crawling, (d) climbing over a wall, (e) climbing onto a platform, (f) climbing down from a platform, (g) walking, (h) climbing upstairs.

TABLE I: We show the visual skill classifier's accuracy on a real-world test set when trained with different data sources.

Method	Sim data only	Real data only	Combined
Accuracy (%)	51.2	90.4	96.7

E. System Robustness

We further evaluate the robustness of our policies by running them on terrains of varying scales, focusing on challenging skills, including climbing over a wall, climbing onto a platform, and climbing down from a platform. As shown in Figure 7 (a) and (b), the terrain skills demonstrate strong robustness while relying solely on proprioceptive input, without visual information. Although the reference motions were designed for fixed terrain configurations (e.g., climbing over a 0.12 m wall, and climbing onto and down from a 0.11 m platform), policies trained with domain randomization on obstacle sizes generalize effectively to a wider range of obstacle heights. In contrast, simply replaying the keyframe animations and naive motion tracking policies will immediately fail when terrain configurations differ. Moreover, we evaluate our hierarchical framework on four obstacle orders: (1) chair → wall → platform → stairs, (2) stairs → platform → wall → chair, (3) wall → stairs → chair → box, and (4) platform → two chairs → stairs → wall (Figure 7). All of these are successfully solved in a zero-shot manner (see the supplementary video).

V. Conclusion

In conclusion, we present controlled experiments on individual components alongside a holistic system evaluation, demonstrating that Locomotion Beyond Feet achieves stable whole-body locomotion on challenging terrains—including low-clearance spaces under chairs, knee-high platforms, knee-high walls, steep ascending and descending stairs—by actively engaging hands, knees, elbows, and other body parts to increase terrain contact. Although evaluated on a miniature humanoid, given the system robustness, we expect that our framework transfers seamlessly to full-size humanoids.

While our system achieves robust whole-body locomotion, several limitations remain. First, keyframe design requires manual effort and domain expertise, though this enables rapid iteration compared to human motion retargeting. Automated design through optimization could enhance scalability. Second, linear interpolation between keyframes trades motion naturalness for simplicity—more sophisticated interpolation methods could improve motion quality. Third, extreme contact-rich strategies like climbing over a wall that succeed in simulation occasionally fail on hardware due to contact modeling approximations. These limitations highlight interesting avenues for future research, while our current system provides a practical solution for diverse terrain traversal.

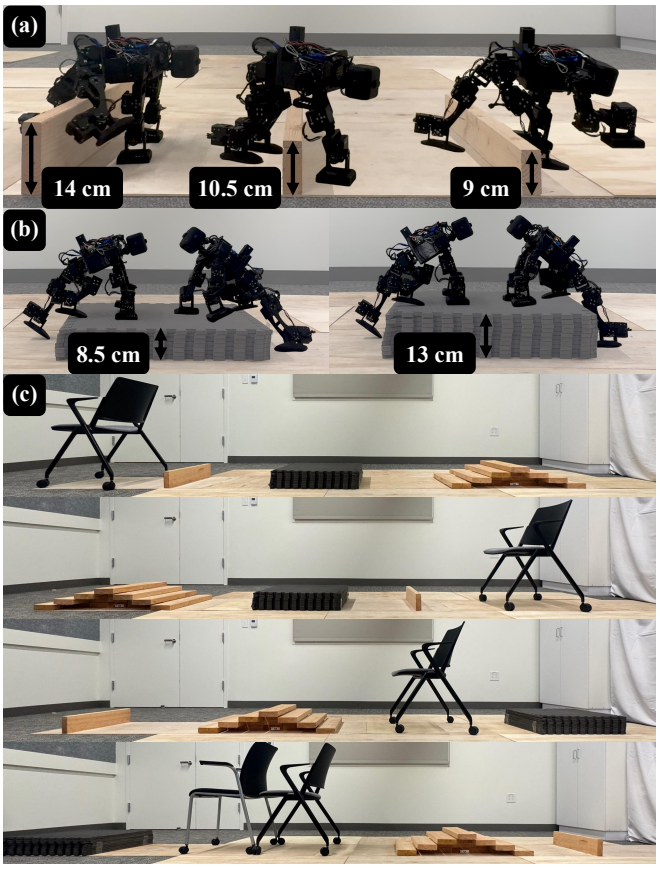


Fig. 7: **System Robustness.** (a) The climbing over a wall policy designed for a 12 cm wall generalizes to wall heights from 9 cm to 14 cm. (b) Similarly, the climbing onto a platform policy designed for a 11 cm platform generalizes between 8.5 cm and 13 cm. (c) We demonstrate zero-shot success of our method across four obstacle orders and varying obstacle counts, such as two chairs in a row.

REFERENCES

[1] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid Parkour Learning," in *8th Annual Conference on Robot Learning*, Sept. 2024.

[2] A. Allshire, H. Choi, J. Zhang, D. McAllister, A. Zhang, C. M. Kim, T. Darrell, P. Abbeel, J. Malik, and A. Kanazawa, "Visual Imitation Enables Contextual Humanoid Control," July 2025.

[3] K. Hu, H. Shi, Y. He, W. Wang, C. K. Liu, and S. Song, "Robot Trains Robot: Automatic Real-World Policy Adaptation and Learning for Humanoids," Aug. 2025.

[4] S. M. Syeda, C. J. Dunmore, M. M. Skinner, L. R. Berger, S. E. Churchill, B. Zipfel, and T. L. Kivell, "Phalangeal cortical bone distribution reveals different dexterous and climbing behaviors in Australopithecus sediba and Homo naledi," *Science Advances*, vol. 11, no. 20, p. eadt1201, May 2025.

[5] C. Mandery, J. Borras, M. Jochner, and T. Asfour, "Analyzing whole-body pose transitions in multi-contact motions," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. Seoul, South Korea: IEEE Press, Nov. 2015, pp. 1020–1027.

[6] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz, "Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective," in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2012, pp. 391–398.

[7] R. Khusainov, D. Azzi, I. E. Achumba, and S. D. Bersch, "Real-Time Human Ambulation, Activity, and Physiological Monitoring: Taxonomy of Issues, Techniques, Applications, Challenges and Limitations," *Sensors (Basel, Switzerland)*, vol. 13, no. 10, pp. 12 852–12 902, Sept. 2013.

[8] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "HumanPlus: Humanoid Shadowing and Imitation from Humans," in *8th Annual Conference on Robot Learning*, Sept. 2024.

[9] Q. Liao, T. E. Truong, X. Huang, G. Tevet, K. Sreenath, and C. K. Liu, "BeyondMimic: From Motion Tracking to Versatile Humanoid Control via Guided Diffusion," Aug. 2025.

[10] S. Chen, Y. Ye, Z.-A. Cao, J. Lew, P. Xu, and C. K. Liu, "Hand-Eye Autonomous Delivery: Learning Humanoid Navigation, Locomotion and Reaching," Aug. 2025.

[11] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon, R. Hoque, L. Paulsen, G. Yang, J. Zhang, S. Yi, G. Shi, and X. Wang, "Humanoid Policy ˜ Human Policy," Mar. 2025.

[12] Y. Ze, Z. Chen, J. P. Araújo, Z.-a. Cao, X. B. Peng, J. Wu, and C. K. Liu, "TWIST: Teleoperated Whole-Body Imitation System," May 2025.

[13] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. M. Kitani, C. Liu, and G. Shi, "OmniH2O: Universal and Dexterous Human-to-Humanoid Whole-Body Teleoperation and Learning," in *2nd Workshop on Dexterous Manipulation: Design, Perception and Control (RSS)*, July 2024.

[14] Y. Higurashi, M. A. Maier, K. Nakajima, K. Morita, S. Fujiki, S. Aoi, F. Mori, A. Murata, and M. Inase, "Locomotor kinematics and EMG activity during quadrupedal versus bipedal gait in the Japanese macaque," *Journal of Neurophysiology*, vol. 122, no. 1, pp. 398–412, July 2019.

[15] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning," in *Proceedings of the 5th Conference on Robot Learning*. PMLR, Jan. 2022, pp. 91–100.

[16] Z. Zhuang, Z. Fu, J. Wang, C. G. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot Parkour Learning," in *7th Annual Conference on Robot Learning*, Aug. 2023.

[17] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme Parkour with Legged Robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 11 443–11 450.

[18] N. Rudin, J. He, J. Aurand, and M. Hutter, "Parkour in the Wild: Learning a General and Extensible Agile Locomotion Policy Using Multi-expert Distillation and RL Fine-tuning," May 2025.

[19] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, "Learning quadrupedal locomotion on deformable terrain," *Science Robotics*, vol. 8, no. 74, p. eade2256, Jan. 2023.

[20] H. Lai, J. Cao, J. Xu, H. Wu, Y. Lin, T. Kong, Y. Yu, and W. Zhang, "World Model-based Perception for Visual Legged Locomotion," Sept. 2024.

[21] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "ANYmal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, Mar. 2024.

[22] J. Lasseter, "Principles of traditional animation applied to 3D computer animation," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 35–44, Aug. 1987.

[23] A. Witkin and M. Kass, "Spacetime constraints," in *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '88. New York, NY, USA: Association for Computing Machinery, June 1988, pp. 159–168.

[24] M. Gleicher, "Motion editing with spacetime constraints," in *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, ser. I3D '97. New York, NY, USA: Association for Computing Machinery, Apr. 1997, pp. 139–ff.

[25] M. Antonelli, F. Dalla Libera, E. Menegatti, T. Minato, and H. Ishiguro, "Intuitive Humanoid Motion Generation Joining User-Defined Key-Frames and Automatic Learning," in *RoboCup 2008: Robot Soccer World Cup XII*, L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou, Eds. Berlin, Heidelberg: Springer, 2009, pp. 13–24.

[26] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "DeepMimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 143:1–143:14, July 2018.

[27] R. Grandia, E. Knoop, M. Hopkins, G. Wiedebach, J. Bishop, S. Pickles, D. Müller, and M. Bächer, "Design and Control of a Bipedal Robotic Character," in *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation, July 2024.

[28] F. Zargarbashi, J. Cheng, D. Kang, R. Sumner, and S. Coros, "RobotKeyframing: Learning Locomotion with High-Level Objectives via Mixture of Dense and Sparse Rewards," in *8th Annual Conference on Robot Learning*, Sept. 2024.

[29] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 5026–5033.

[30] M. Izani, Aishah, A. Eshaq, and Norzaiha, "Keyframe animation and motion capture for creating animation: A survey and perception from industry people," in *Proceedings. Student Conference on Research and Development, 2003. SCORED 2003.*, Aug. 2003, pp. 154–159.

[31] B. Ma, N. Xu, C. Qi, X. Liu, Y. Mo, J. Wang, and C. Lu, "PPL: Point Cloud Supervised Proprioceptive Locomotion Reinforcement Learning for Legged Robots in Crawl Spaces," Aug. 2025.

[32] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield, "FoundationStereo: Zero-Shot Stereo Matching," Apr. 2025.

[33] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu, "RoboCook: Long-Horizon Elasto-Plastic Object Manipulation with Diverse Tools," Oct. 2023.

[34] Y. Lee, J. J. Lim, A. Anandkumar, and Y. Zhu, "Adversarial Skill Chaining for Long-Horizon Robot Manipulation via Terminal State Regularization," Nov. 2021.

[35] P. Xu, X. Shang, V. Zordan, and I. Karamouzas, "Composite Motion Learning with Task Control," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 93:1–93:16, July 2023.

[36] G. Christmann, Y.-S. Luo, and W.-C. Chen, "Expert Composer Policy: Scalable Skill Repertoire for Quadruped Robots," Mar. 2024.

[37] P. Goel, H. Zhang, C. K. Liu, and K. Fatahalian, "Generative Motion Infilling From Imprecisely Timed Keyframes," Mar. 2025.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *ArXiv*, July 2017.

[39] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, "A closed-form solution for real-time ZMP gait generation and feedback stabilization," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. Seoul, South Korea: IEEE, Nov. 2015, pp. 936–940.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778.

[41] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and Model set: Towards common benchmarks for manipulation research," in *2015 International Conference on Advanced Robotics (ICAR)*, July 2015, pp. 510–517.

[42] H. Shi, W. Wang, S. Song, and C. K. Liu, "ToddlerBot: Open-Source ML-Compatible Humanoid Platform for Loco-Manipulation," Feb. 2025.

[43] Y. Chen, Z. Li, Y. Wang, H. Zhang, Q. Li, C. Zhang, and G. Lin, "Ultra3D: Efficient and High-Fidelity 3D Generation with Part Attention," July 2025.