IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Learning Autonomous and Safe Quadruped Traversal of Complex Terrains Using Multi-Layer Elevation Maps

Yeke Chen, Ji Ma, Zeren Luo, Yimin Han, Yinzhao Dong, Bowen Xu, and Peng Lu†

***Abstract*—Legged robots hold great promise for agile and flexible mobility across diverse and unstructured terrains, inspired by the remarkable adaptability of bipeds and quadrupeds in nature. However, achieving robust autonomous locomotion in cluttered and complex environments remains a significant challenge. In this work, we present a hierarchical control framework for quadrupedal robots that enables safe and autonomous traversal of cluttered terrains. Central to our approach is a novel multi-layer elevation map representation, which is generalized enough to capture a wide range of terrains. To further improve policy generalization and maneuverability, we incorporate terrain augmentation, knowledge distillation, and carefully designed reward functions. Extensive simulation experiments demonstrate that each component contributes to improved policy generalization, and that our terrain representation is more efficient and informative than existing alternatives. By training a terrain compressor in simulation, we successfully deploy our system on a low-cost quadrupedal robot in real-world environments, showcasing the practicality and robustness of our approach.**

***Index Terms*—Legged locomotion, robot learning, reinforcement learning**

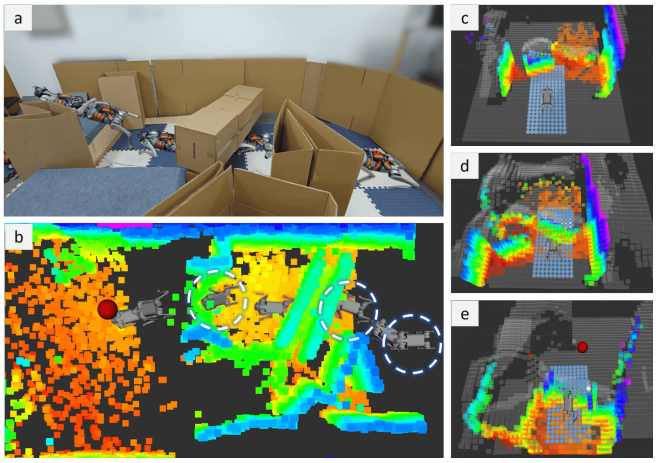

Fig. 1: **Real-world Deployment of the System** (a) A quadrupedal robot autonomously and safely traverses a cluttered terrain to reach a target point. (b) Bird's-eye view of the robot's trajectory. (c)(d)(e) The terrain compressor extracts the multi-layer elevation maps from various occupancy grid maps with generalization, despite numerous missing areas and noise. The spheres in the picture are the multi-layer elevation maps, the rainbow-colored squares are the occupancy grid maps, and the gray ones are the top layer of the larger multi-layer elevation map.

I. Introduction

Animals, especially bipeds and quadrupeds, can reach a wide variety of places and have the flexibility to adapt their behavior to their surroundings, showing amazing athletic intelligence. As their artificial counterparts, legged robots are designed to have the potential to traverse a variety of terrains. Although legged robots have been successfully deployed in a variety of indoor and outdoor environments, achieving flexible and safe mobility in diverse and cluttered environments remains a significant challenge in legged robotics.

Recent developments in learning-based approaches have allowed legged robots to move on various terrains [1] [2] [3]. Robots learn robust skills through trial and error without the need for precise modeling, and massively parallel simulation [4] greatly improves this process. By leveraging proprioceptive sensors, legged robots can implicitly estimate terrain properties and system states, achieving reliable locomotion through a variety of off-road terrains [1]. Some recent works show legged robots equipped with exteroceptive sensors, such as cameras and LiDAR, capable of moving through more challenging and risky environments such as stairs and gaps, even performing difficult parkour tasks with great agility [5] [6] [7] [8] [9] [10].

However, few of the above works have attempted to achieve autonomous robot movement in cluttered scenarios. In a cluttered scenario, an intelligent robot should be able to safely avoid dangerous areas, quickly traverse necessary obstacles, and flexibly adjust its behaviors. Its policies should be able to maintain generalization capabilities on various unstructured terrains. Accordingly, its external perception should be concise, generalized, and informative. That is, the perceptual input can represent a variety of terrains, including gaps and blocks, as well as confined areas, compatible with a variety of sensor modalities and mounting positions. It should have a wide field of view without excessive density. For reconstructed perception, it should not degrade in an unstructured environment.

This paper introduces a hierarchical system for quadrupedal robots to traverse cluttered and challenging terrains with obstacle avoidance autonomously. The outer local navigation policy focuses on reaching long-distance target points safely and agilely, computing high-level commands, while the inner direction-aware locomotion policy follows these commands.

We also propose a multi-layer elevation map as an efficient, informative, and generalized terrain representation. To enable real-world applicability, a neural terrain compressor is trained to convert occupancy grid maps into this specialized elevation map format. This approach inherently supports generalization across multiple sensor modalities and facilitates the integration

Manuscript received: May, 2, 2025; Revised July, 15, 2025; Accepted July, 27, 2025.

This paper was recommended for publication by Editor Jens Kober upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the General Research Fund under Grant 17204222, and in part by the Seed Fund for Collaborative Research.

The authors are with the Adaptive Robotic Controls Lab (ArcLab), Department of Mechanical Engineering, The University of Hong Kong, Hong Kong SAR 999077, China. chen-yeke@connect.hku.hk, maji@connect.hku.hk, zerluo@connect.hku.hk, ymhan2023@connect.hku.hk, dongyz@connect.hku.hk, link.bowenxu@connect.hku.hk.

†Corresponding author: lupeng@hku.hk.

The supplementary video is available at https://youtu.be/nlDQpzXRUSw

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

of mapping methods that preserve historical information.

To further promote generalization, maneuverability, and robustness, we use carefully designed terrains, tailored rewards, terrain augmentation, and knowledge distillation. Experiments in large-scale, unseen randomized environments demonstrate the superior generalization and maneuverability of our approach. Our framework is also validated on a real-world, low-cost quadrupedal robot in both indoor and outdoor environments.

The main contributions of this paper are summarized as follows:

- **A novel and concise terrain representation** is introduced to cover a wide range of scenarios, with a neural terrain compressor making it applicable in real-world contexts.
- **A hierarchical navigation system** is proposed for quadrupedal robots to move safely and autonomously in cluttered environments.
- **Carefully designed terrains and rewards** are employed to enhance policy generalization and maneuverability.

II. Related Work

A. Perceptive Legged Locomotion

Perceptive legged locomotion is crucial for legged robots to move successfully on challenging terrains, and many impressive results have been achieved. Combined with elevation maps, model-based methods enable robots to traverse uneven terrains [11] [12]. However, these methods often lack agility, robustness to external disturbances, and the ability to generalize to varying ground physical properties.

Today, learning-based controllers are widely used due to their lack of reliance on precise modeling and their strong robustness to disturbances [1] [2] [3] [13]. Leveraging elevation maps, robots achieve perceptive locomotion, capable of traversing more challenging environments [14] [15] [9] [16]. However, elevation maps fail to represent confined areas. Consequently, recent works use the depth image as external perception, employing RNNs to implicitly estimate surrounding terrains, achieving walking on risky terrains and even performing difficult parkour tasks with great agility [5] [6] [10] [8]. Since a depth image includes excessive redundant information in a limited field of view and changes with camera models and mounting locations, [7] [17] use occupancy voxels as input, which can provide a more comprehensive representation of the surrounding environment and are not dependent on sensor modalities. Beyond directly utilizing external perception. [18] uses depth images to predict ray distances, [19] maintains neural volumetric memory using depth images, while [7] [20] reconstruct occupancy voxels to remove noise and occlusion.

In this work, we reconsider the perception input and propose a multi-layer elevation map as a terrain representation. It is efficient, concise, and generalized, similar to a single-layer elevation map, while also being able to represent confined scenarios with overhanging obstacles, like a depth image or an occupancy grid map. We design and train a neural terrain compressor to extract maps, reduce noise, and handle missing areas, which is deployable in the real world.

B. Legged Local Navigation

To navigate a robot to a goal, a common approach is to explicitly plan a path or a trajectory and then have the controller track it. This can be achieved through classic planning methods [21] [22] and learning-based methods [23] [24]. However, these methods tend to neglect the dynamics of the robots, which particularly hinders quadrupedal robots from traversing complex terrains, leading to a suboptimal situation.

In order to fully exploit the locomotion ability of quadrupedal robots, another intuitive idea is learning an end-to-end goal-based policy. [15] [9] formulate a locomotion task as a goal-conditioned local navigation problem, which allows robots to freely modulate their actions in a given time, achieving remarkable performance. A similar formulation of a goal-conditioned problem can also be found in [25] [26] [27] where velocity commands are difficult to assign in the task. A common difficulty in goal-based problems is sparse reward signals that complicate exploration. To address this issue, reward shaping methods [25] and special rewards [9] [15] are used to facilitate efficient exploration. Some works use a hierarchical learning method, their navigation policy outputs a single twist [28], twist residual [17] or point goal [7] [16], fully making use of the existing locomotion policies through reinforcement learning (RL).

In our task, training an end-to-end policy from scratch is quite challenging. Therefore, we adopt a goal-based navigation formulation to encourage flexible behaviors, while also utilizing a hierarchical structure to preserve fundamental locomotion skills. Our approach shares this hierarchical scheme with [16], but differs by using a multi-layer elevation map to capture overhanging obstacles and a neural terrain compressor to handle noisy and incomplete LiDAR data. In hierarchical methods, the locomotion policy is frozen once pretrained, which is known to potentially degrade performance when encountering unseen or out-of-distribution (OOD) scenarios. To address this, our locomotion policy is obtained through joint distillation of multiple expert policies, demonstrating better adaptability to a new environment. Besides, carefully designed shaping rewards and procedurally generated random terrains with augmentation are used to mitigate the sparse reward situation and enhance policy robustness, respectively.

III. Method

A. Overview

Our approach employs a hierarchical structure that enables quadrupedal robots to autonomously traverse cluttered terrains with safety and agility, as illustrated in Fig. 2.

The locomotion policy takes the proprioceptive observation $\mathbf{o}_t^p$ and the multi-layer elevation map $\mathbf{e}_t$ as input, and outputs joint target positions $\mathbf{a}_t \in \mathbb{R}^{12}$. In general, $\mathbf{o}_t^p \in \mathbb{R}^{45}$ remains consistent with previous blind locomotion works, including body yaw velocity $\omega_t \in \mathbb{R}^3$, projected gravity $\mathbf{g}_t \in \mathbb{R}^3$, desired body velocity command $\mathbf{c}_t \in \mathbb{R}^3$, joint positions $\mathbf{q}_t \in \mathbb{R}^{12}$,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

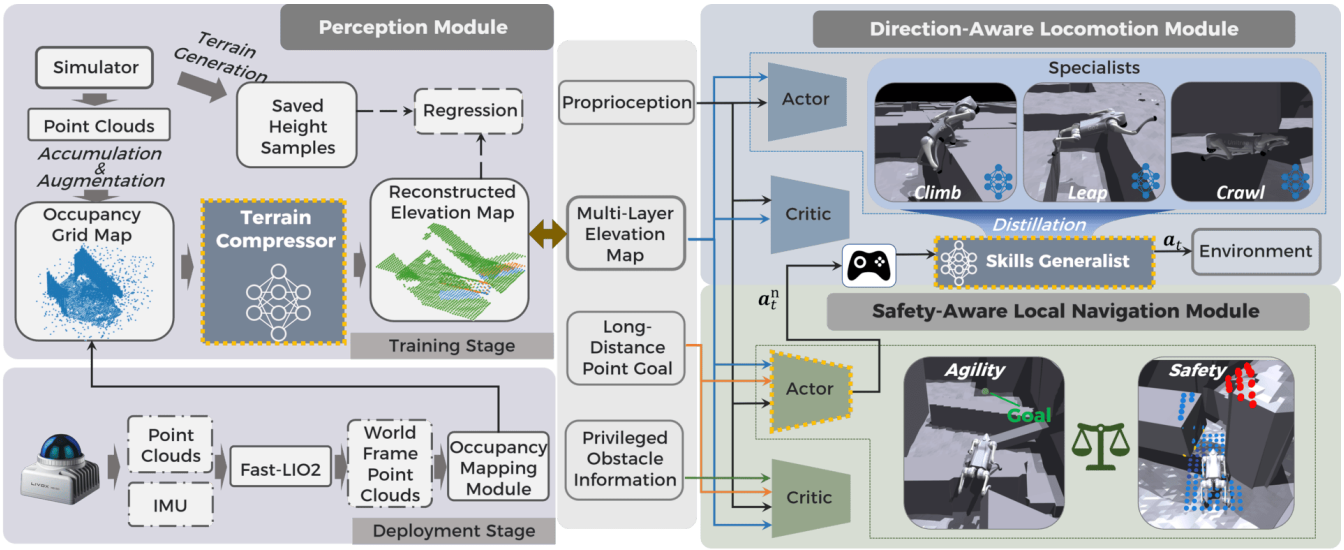

Fig. 2: **Overview of the System** We first train several locomotion skills (Specialists), which are then distilled into a single locomotion policy (Generalist). The local navigation policy is then trained to reach a long-distance target point with safety and agility. A terrain compressor, which is used to extract the multi-layer elevation map from the occupancy grid map, is trained **independently** in simulation and deployed in the real world. Modules with yellow dashed lines are used for real-world deployment.

joint velocities $\dot{\mathbf{q}}_t \in \mathbb{R}^{12}$, and the action of the last step $\mathbf{a}_{t-1} \in \mathbb{R}^{12}$. This can be expressed as:

$$\mathbf{o}_t = [\mathbf{o}_t^p, \mathbf{e}_t], \tag{1a}$$

$$\mathbf{o}_t^p = [\omega_t, \mathbf{g}_t, \mathbf{c}_t, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}]. \tag{1b}$$

We train a diverse set of specialized locomotion skills (Specialists), each meticulously optimized for distinct terrains. Through a comprehensive distillation process, these terrain-specific experts are fused into a unified locomotion policy (Generalist). This consolidated policy dynamically integrates their individual strengths, enabling seamless adaptation across heterogeneous environments by implicitly activating the most relevant sub-skills.

The local navigation policy is responsible for efficiently guiding the agent toward a target point in real time, while strictly enforcing collision-free motion to ensure safe operation in dynamic environments. It outputs the velocity command $\mathbf{c}_t$ given observation $\mathbf{o}_t^n$ that reuses the locomotion observation $\mathbf{o}_t^p$. The multi-layer elevation map is replaced by a larger map $\mathbf{e}_t^n$, the velocity command is replaced by a long-distance point goal $\mathbf{c}_t^n \in \mathbb{R}^3$, and the last locomotion action is replaced by the last navigation policy output $\mathbf{a}_{t-1}^n$. This can be represented as:

$$\mathbf{a}_t^n = [\mathbf{c}_t], \tag{2a}$$

$$\mathbf{o}_t^n = [\mathbf{o}_t^{p_n}, \mathbf{e}_t^n], \tag{2b}$$

$$\mathbf{o}_t^{p_n} = [\omega_t, \mathbf{g}_t, \mathbf{c}_t^n, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}^n]. \tag{2c}$$

In the asymmetric actor-critic framework, the navigation critic receives privileged observations, including the body linear velocity $\mathbf{v}_t \in \mathbb{R}^3$ and obstacle information $\mathbf{i}_t$, while the locomotion critic only receives the body linear velocity $\mathbf{v}_t$. $\mathbf{i}_t$ consists of posture and geometric information for up to $k \leq k_{\max}$ obstacles near the robot, which will be detailed in Section III-D.

High-dimensional observation components are first compressed into feature vectors via dedicated encoders. In particular, $\mathbf{e}_t$ and $\mathbf{e}_t^n$ are processed by convolutional neural networks, which are well-suited for data with grid-like structures. $\mathbf{i}_t$ is processed by a Transformer encoder since $\mathbf{i}_t$ represents a variable-length information sequence where each obstacle can be viewed as a token. These feature vectors are then concatenated with the other observation components and fed into an output network to get the action or value.

B. Multi-Layer Elevation Map

Our proposed multi-layer elevation map extends the conventional elevation map to better represent complex terrains, particularly those with confined or overhanging structures. We simplify a constrained region as a ground surface overlaid by N non-intersecting overhanging convex polyhedra, where a vertical line intersects the terrain at most $1 + 2 \times N$ times. Thus, the multi-layer elevation map has $1 + 2 \times N$ layers. In this work, we simplify it to a 3-layer elevation map (Figs. 3(j) and 3(k)) by treating overhangs as a single convex polyhedron, which can handle most situations.

In simulation, the 3-layer elevation map is obtained by storing the lowest three z-values per grid cell from terrain meshes; missing layers are filled by repeating existing values (Fig. 3(k)). Since this direct extraction is infeasible in real-world scenarios, we **independently** train a lightweight UNet-like neural network terrain compressor that converts occupancy grid maps into 3-layer elevation maps.

In real-world scenarios, the perception process encounters substantial complexity. The occupancy grid maps produced by the mapping module frequently exhibit numerous voids, primarily due to occlusions and limited LiDAR scanning frequency. Furthermore, these occupancy maps are often corrupted by noise stemming from pose estimation jitter and intrinsic sensor inaccuracies. Beyond these challenges, the cluttered and unstructured nature of real-world environments induces input distributions that deviate from those encountered during simulation-based training.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

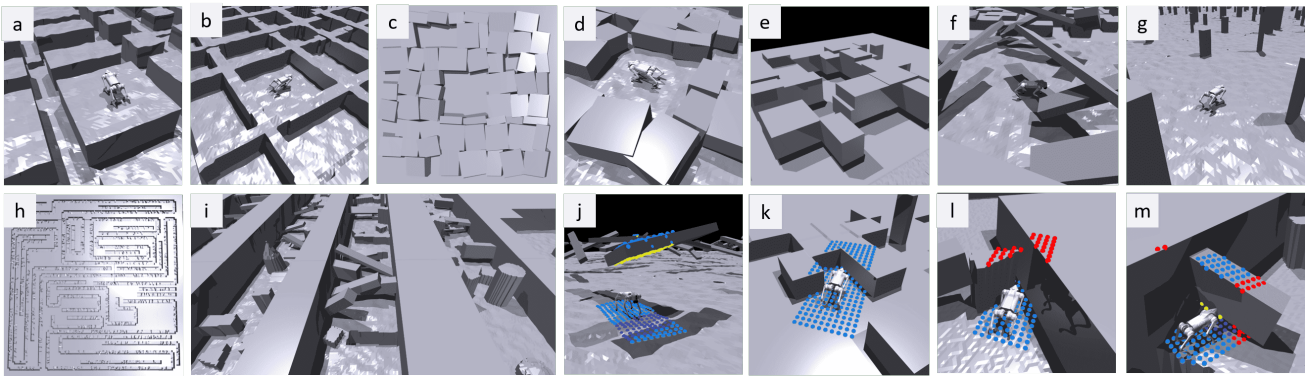

Fig. 3: **Illustration of Terrain Generation and the Multi-Layer Elevation Map** (a), (b) Enclosure terrains. (c) A bird's-eye view of a box terrain. (d) A box terrain for climbing. (e) A box terrain for crawling. (f), (g) Terrain augmentation. (h) A bird's-eye view of the navigation terrain. (i) The navigation terrain. (j) An example of a 3-layer elevation map, where the purple sphere represents the lowest layer, the yellow represents the second layer, and the blue represents the highest layer. (k) The 3-layer elevation map serves as a conventional elevation map when there are no overhanging obstacles. (l), (m) Demonstration of the dangerous points, where the red points are dangerous points that are not traversable.

To gather realistic data, we generate diverse terrains and collect data pairs within Gazebo simulation environments. Occupancy grid maps are obtained from the accumulated LiDAR point clouds, while ground-truth 3-layer elevation maps are precomputed during terrain generation. To enhance the model's generalization, we apply data augmentation techniques such as random clearing of point clouds and injecting Gaussian and impulsive noise.

Once available in real-world scenarios, the 3-layer elevation representation offers a promising approach for complex terrain modeling. It reduces GPU memory and computational overhead compared to ray casting during policy training, due to the decoupling of terrain compressor and policy training. Moreover, it is more data-efficient than occupancy grid maps and provides a richer representation of the surrounding environment than depth images.

The input occupancy grid has a resolution of $0.1 \times 0.1\,\mathrm{m}^2$ per cell over a 20×10 XY grid with 100 height bins at 0.03 m resolution along Z. The 3-layer elevation map compresses this into three height layers at the same XY resolution. The terrain compressor is trained with an L1 loss on elevation values, achieving a mean absolute error (MAE) of 0.030 m on training data and 0.049 m on testing data.

C. Locomotion policy

The reward functions and terrains used for skill training largely follow those in previous works [4]. Below, we highlight the main differences.

1) Terrain Generation and Augmentation: Our terrain integrates a large height-field mesh with multiple overhanging box meshes, simulating a variety of complex environments such as rough ground, stairs, and gaps, as well as confined spaces. Building on prior work [4], we introduce several novel terrain types. **Enclosure terrains** (Figs. 3(a) and 3(b)), comprising closed curves of random width and height, foster climbing or leaping skills. **Box terrains** (Figs. 3(c), 3(d) and 3(e)), with boxes of varying heights and poses, is specifically used for training crawling and climbing behaviors.

We also employ **terrain augmentation** techniques to enhance policy robustness against OOD scenarios. In addition to randomized positions, postures, and sizes, the environment further incorporates multiple high walls, poles, and overhanging blocks that are randomly generated, as illustrated in Figs. 3(f) and 3(g). These elements are designed to introduce subtle physical perturbations to the robots' bodies, such as minor collisions, without causing catastrophic failures. However, they significantly impact robots' external observations, thereby greatly increasing the observation space variety.

Importantly, the terrain augmentation is applied during the distillation process such that these additional terrain elements are only visible to the student policy, while the teacher policies do not observe them. This separation enriches the student's observations and helps prevent the teacher from encountering OOD scenarios.

2) Direction-Aware Linear Velocity Tracking Reward: Velocity tracking rewards are widely adopted in legged locomotion tasks, particularly for encouraging robots to follow desired velocity commands across various terrains. However, in complex environments, such rewards can cause unintended behaviors. For example, robots may bypass difficult obstacles or sacrifice short-term tracking accuracy to obtain easier rewards later, sometimes losing directional control, as shown in Fig. 4.

To mitigate this issue, we introduce a direction-aware linear velocity tracking reward. Rather than randomly sampling angle velocity commands, our environment samples movement yaw directions, from which the target angular velocity is derived. The linear velocity tracking reward $r_{\text{vel\_tracking}}$ is further modulated by the angular error between the robot's actual orientation and the target direction, promoting consistent heading while tracking velocity commands. This can be written as:

$$r_{\text{vel\_tracking}} = \exp\left(-\frac{1}{\sigma}(\mathbf{v}_{xy} - \mathbf{c}_{xy})^2\right)\frac{\cos(\theta_{\text{error}}) + 1}{2}, \tag{3a}$$

$$\theta_{\text{error}} = |\theta - \theta_{\text{target}}|, \tag{3b}$$

where $\mathbf{v}_{xy} \in \mathbb{R}^2$ denotes the robot's body linear velocity in the x-y plane, $\mathbf{c}_{xy} \in \mathbb{R}^2$ is the target body linear velocity, σ is a hyperparameter that controls the sensitivity of the tracking error, λ is a hyperparameter used to calculate the angular velocity command, θ represents the robot's current yaw angle, while θ_{target} denotes the target yaw angle. The yaw velocity

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

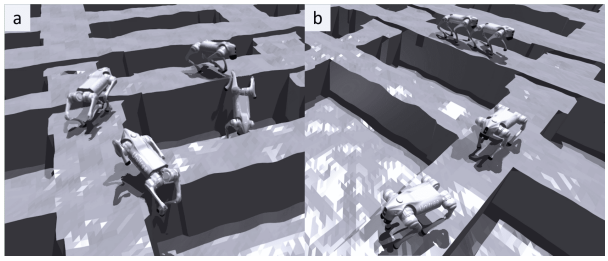

Fig. 4: **Effect of the Reward on Steering Behavior** Both robots on the left and right receive linear and angular velocity commands to move forward while turning. (a) Using the direction-aware linear velocity tracking reward, the robot exhibits more controlled steering. (b) With the original reward, the robot fails to maintain directional control and only turns when encountering crossed paths.

command c_ω is computed as a scaled yaw error, which can be written as $c_\omega = \lambda\theta_{\text{error}}$.

3) Training Details: We train our policies in simulation through Isaac Gym. Crawling, leaping, and climbing skills similar to [5] are trained. A generalist policy is then distilled using DAgger [29] on a mixed terrain, where the environment is partitioned into sub-regions, and each sub-region is managed by a dedicated specialist policy that provides expert actions based on the robot's current location.

A curriculum is carefully designed for robots to progressively enhance abilities during skill training: The lower surface is lowered for crawling, the upper surface is raised for climbing, and gaps are widened for leaping.

D. Local Navigation policy

1) Navigation Terrain: The navigation terrain consists of path sections flanked by tall walls and clustered obstacles of varied types and sizes (Figs. 3(h), and 3(i)). Owing to its procedurally generated nature, the navigation terrain offers high complexity and diversity; even minor changes in the random seed produce fundamentally different layouts. Therefore, it serves both as a training ground and a benchmark for generalization and traversal performance. Since the locomotion policy remains frozen during the local navigation policy training, it is crucial to ensure that the locomotion policy is robust enough to adapt to new navigation terrains.

Each obstacle's type (a one-hot vector), position, orientation, and size are stored together in an information vector. Each path section maintains a K-D tree for efficient querying of nearby obstacles, which returns the information vectors of the $k \leq k_{\max}$ nearest obstacles within a specified range. At each training step, the current position of each robot is used to perform a query.

2) Rewards: In the context of local navigation, the robot is required to avoid obstacles and reach the target point, as shown in Fig. 5. Thanks to the strong generalization ability of the locomotion policy, we can employ a relatively simple reward structure for the local navigation policy. The total reward is defined as:

$$r_{\text{nav}} = \sum \left(w_\star \cdot r_\star\right), \tag{4}$$

where $\star \in \{\text{close}, \text{collision}, \text{avoidance}, \text{arrival}, \text{termination}\}$. The terms $r_{\text{collision}}$, r_{arrival}, and $r_{\text{termination}}$ denote the penalty for collisions, the reward for successfully reaching the goal, and the penalty for early episode termination, respectively. These components are straightforward. The term r_{close} is a simple shaping reward designed to encourage the robot to move closer to the goal at each timestep. It is formulated as:

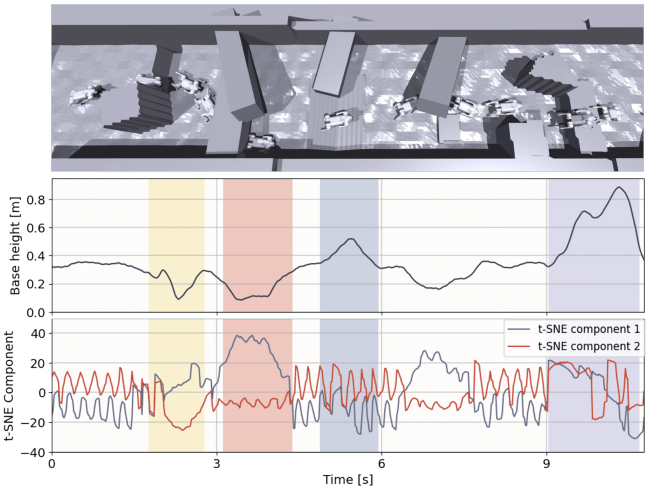

Fig. 5: **An Example of the Local Navigation Task** The robot autonomously navigates in a cluttered environment. The curves of the base height and the t-SNE embedding of the joint angles are shown, with yellow, red, blue, and purple shaded areas representing the gap, overhanging, slope, and climbing regions, respectively.

$$r_{\text{close}} = d_{\text{last}} - d, \tag{5}$$

where d represents the current distance to the goal, and d_{last} is the distance to the goal at the previous timestep. This reward is positive when the robot makes progress towards the goal and negative when it moves away.

The term $r_{\text{avoidance}}$ is also a shaping reward that encourages obstacle avoidance and selecting the safest, easiest route. Multiple $(x,\ y)$ points are defined under the robot's body frame, with denser sampling in the front. Each point has an associated danger score, and a point is considered a dangerous point if it is non-traversable (Figs. 3(l) and 3(m)). The score is computed from height samples h_0, h_1, h_2 as:

$$\Delta h = h_1 - \max(h_0, 0), \tag{6a}$$

$$s_i = \begin{cases} 0.1, & \Delta h \geq 0.2, \\ \max\begin{pmatrix} |\operatorname{clip}(h_0, -0.4, 0.7)|, \\ |\operatorname{clip}(h_2, -0.4, 0.7)| \end{pmatrix}, & \text{otherwise}, \end{cases} \tag{6b}$$

where s_i is the danger score of the i-th point. Points with $s_i = 0.7$ are non-traversable. The reward is defined as:

$$r_{\text{avoidance}} = -\, p_{\text{danger}} \mathbb{I}(p_{\text{danger}} > T_{\text{danger}}) \|\mathbf{v}\| - \lambda_s \sum_{i=1}^{N} s_i, \tag{7a}$$

$$p_{\text{danger}} = \frac{n_{\text{danger}}}{N}, \tag{7b}$$

where p_{danger} is the fraction of dangerous points among the total N points, n_{danger} is the number of dangerous points, T_{danger} is a threshold that determines when the danger probability p_{danger} is considered significant enough to trigger avoidance behavior, λ_s is a weighting hyperparameter, and $\mathbb{I}$ is the indicator function (1 if the condition holds, 0 otherwise). The function encourages robots to proactively avoid dangerous

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

obstacles, slow down when a collision is upcoming, and choose the safest route.

3) Training Details: During simulation, long-distance point goals are randomly sampled at a distance between 4 and 20 meters from the robot in any direction. In real-world deployment, the target is typically set at approximately 4 meters away from the robot.

IV. Experimental Results

A. Implementation Details

We utilize the MID360, a low-cost and lightweight LiDAR sensor, to acquire point cloud data. Odometry estimation is performed leveraging [30], while occupancy grid maps are constructed via [31].

Domain randomization parameters are summarized in Table I, including robot body mass, initial joint positions, and motor strength. For better sim-to-real transfer, we incorporate system delays in both actions and the 3-layer elevation map.

Parameters	Range	Unit
Added base mass	[-2, 2]	Kg
Ground Friction	[0.25, 1.75]	-
Motor Latency	[0, 30]	ms
3-Layer Elevation Latency	[60, 120]	ms
Motor Offset	[-0.02, 0.02]	Rad
Motor Strength Factor	[0.9, 1.1]	-

TABLE I: **Randomization range of critical parameters**

B. Simulation Experiments

To evaluate the generalizability and robustness of a policy, designing a single "track" is insufficient; instead, a large number of random and novel scenarios are required. Therefore, to comprehensively assess the generalization and maneuverability of our proposed hierarchical system in cluttered terrains, we generate large-scale randomized cluttered navigation terrains with varying levels of difficulty. Leveraging the high randomness of the terrain generation process, we apply the same generation scheme for evaluation while adjusting parameters to ensure that the evaluation scenarios are completely distinct from the training scenarios. We create 500 robots, each performing the navigation task and being evaluated over 1500 time steps. The following metrics are collected to assess the policy performance:

- **Average Termination Count per Robot (ATC):** Average episodes terminated due to falls or unsafe states per environment.
- **Average Success Count per Robot (ASC):** Average successful goal reaches per environment.
- **Average Goal-directed Distance per Robot (AGD):** Mean distance progressed toward the goal per environment.
- **Success Path Sum per Robot (SPS):** Average sum of $\text{success}_i \cdot \text{path\_length}_i$ per environment, where success_i is a binary goal-reaching indicator, and path_length_i is the corresponding segment length.

To validate each component, we compare our proposed method with the following baselines:

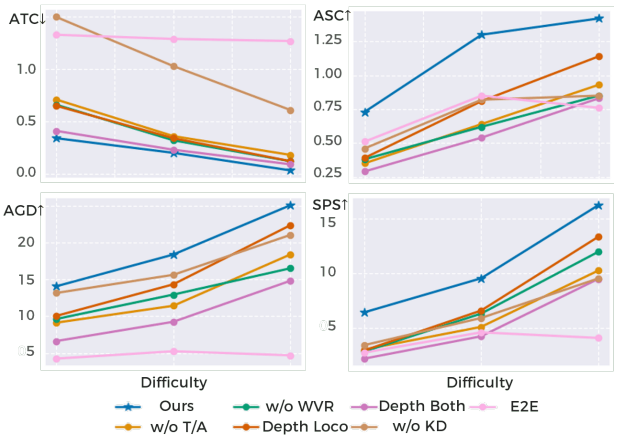

Fig. 6: **Evaluation Results on Navigation Terrain**

- **No Knowledge Distillation (w/o KD):** A hybrid local navigation policy that selects among specialist skills while outputting velocity commands is trained using hybrid PPO, similar to [7].
- **End-to-End (E2E):** A non-hierarchical policy trained end-to-end, without explicit hierarchical decomposition.
- **No Terrain Augmentation (w/o TA):** The locomotion policy is trained without terrain augmentation.
- **Depth Locomotion (Depth-Loco):** The locomotion policy replaces the 3-layer elevation map representation with depth images and uses a GRU to implicitly estimate the surrounding states, similar to [5] [6].
- **Depth Both Locomotion and Navigation (Depth-Both):** Both the locomotion and local navigation policy use depth images as input.
- **No Weighted Velocity Reward (w/o WVR):** The locomotion policy is trained without the direction-aware linear velocity tracking reward.

The performance of each method on the benchmark is summarized in Fig. 6. The results demonstrate the generalizability and maneuverability of our approach on cluttered terrain, as it performs best in all metrics. Both terrain augmentation and knowledge distillation effectively improve the policy's generalization ability by enhancing observational diversity. Additionally, the hierarchical system is more effective than a single end-to-end policy in these challenging environments because it decomposes the task and facilitates learning for both policies. The locomotion policy utilizing a 3-layer elevation map outperforms the depth-based policy, primarily due to the excessive redundancy of depth images within a limited field of view and the implicit way they estimate the surrounding terrain with GRUs. Furthermore, the direction-aware linear velocity tracking reward significantly enhances the locomotion policy's maneuverability, resulting in a higher success rate, longer traveled distance, and fewer termination counts, mainly due to better danger avoidance.

Policy Type	Leap ↑	Leap-Stop ↑	Climb-Back ↑
Elevation-Based	**1.00**	**1.00**	**1.00**
Depth-Based	0.76	0.48	0.00

TABLE II: **Evaluation Results of Skills in Three Scenarios**

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

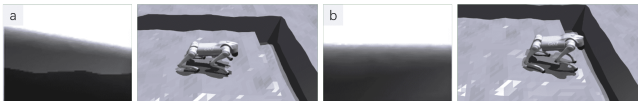

Fig. 7: **Depth Observations of the Robot at Different Positions** (a) The robot is far from the gap, which is clearly visible in the depth image (black area). (b) The robot is close to the gap, but the gap is missing in the depth image (mostly gray, indicating ground), causing perception difficulties.

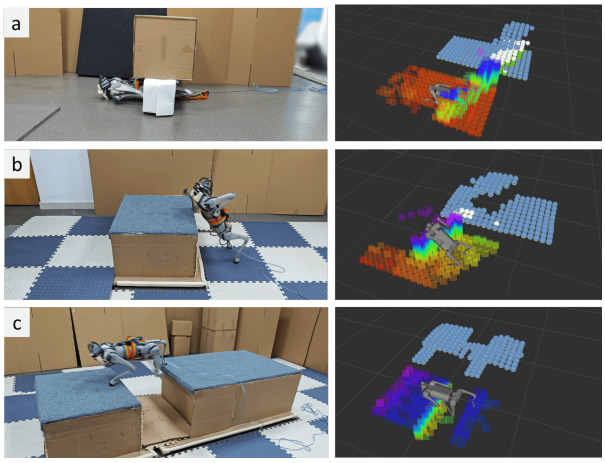

Fig. 8: **3-Layer Elevation Map Outputs in Various Skill-Training Scenarios** The spheres represent the 3-layer elevation maps: blue spheres correspond to the top layer, white spheres to the second layer, and green spheres to the lowest layer. The rainbow-colored regions represent the occupancy grid map input. (a) Crawling under an overhanging block. (b) Climbing up to a block. (c) Leaping over a gap.

We also assess policies' ability to represent the terrain by testing them in three scenarios:(1) Moving forward to the gap all the time rapidly (**Leap**). (2) Stopping after walking to the edge of a gap and then moving forward again (**Leap-Stop**). (3) Climbing across an obstacle followed by moving backward (**Climb-Back**). The results, also shown in Table II, reveal that combining the depth map with an RNN makes reliable terrain characterization challenging. The depth camera's limited viewing angle prevents it from capturing the area directly beneath the robot, causing crucial details to be lost as the robot nears obstacles (Fig. 7(b)). Consequently, the RNN must rely on memory to infer terrain features, which becomes unreliable in sudden stops or backward motions rarely seen during training. This lack of relevant experience further degrades terrain representation and policy performance in the Leap-Stop and Climb-Back scenarios.

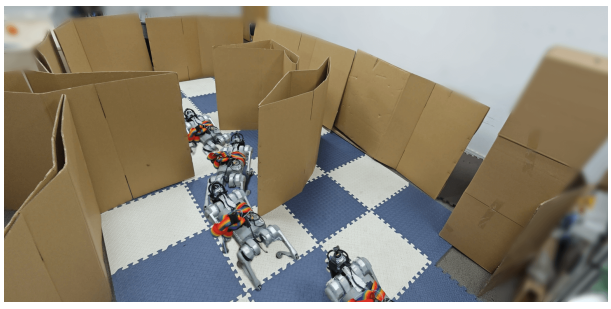

Fig. 9: **Continuous Obstacle Avoidance** The robot avoids obstacles continuously with precise movements.

C. Real-World Experiments

We conduct real-world experiments to validate that the proposed method can be successfully deployed in real-world scenarios with a terrain compressor, despite the unstructured and cluttered nature of the environment, as well as noisy and occluded perception inputs.

Fig. 8 demonstrates 3-layer elevation map outputs in various skill-training scenarios. The terrain compressor effectively predicts missing areas and reduces noise in the occupancy grid map. Specifically, the terrain compressor predicts the lower surface of the overhanging block and completes the ground obscured by the block. It also predicts the invisible ground when climbing up and accurately represents the elevation of the gap terrain, even though the LiDAR scans only part of the side of the opposite box.

We further evaluate the local navigation policy in a multiple-obstacle scenario shown in Fig. 9, where the robot avoids obstacles continuously to reach the goal, requiring accurate perception input and proper velocity commands. The results demonstrate the local navigation policy's obstacle avoidance capability and the locomotion policy's effective maneuverability.

We also evaluate the entire system on a randomly cluttered navigation terrain, where the robot autonomously avoids high walls and leverages different skills to traverse the terrain, as shown in Fig. 1. This demonstrates the locomotion policy's adaptability to various terrains, the local navigation policy's ability to select appropriate routes as well as avoid hazards, and the terrain compressor's generalizability to random and complex scenarios.

Key quantitative metrics from these real-robot experiments are summarized in Table III.

Task	Quantitative Metric	Success Rate
Climb	Height 0.45 m	5/5
Leap	Gap width 0.45 m	4/5
Crawl	Height 0.2 m	5/5
COA	Narrowest clearance between obstacles 0.5 m	4/5
NCT	Avg. time 7.4 s	4/5

TABLE III: **Quantitative Metrics from Real-Robot Tests for Climb (Climbing), Leap (Leaping), Crawl (Crawling), COA (Continuous Obstacle Avoidance), and NCT (Navigation on Cluttered Terrain)**

Finally, we assess the performance of our locomotion policy and terrain compressor in an outdoor environment, as illustrated in Fig. 10. The robot climbs a high step in an empty outdoor area, demonstrating that the mapping system can still operate in such conditions. The terrain compressor successfully predicts the lower surface of the bench by the occupancy grids at the front of it, enabling the robot to crawl beneath. Moreover, the robot ascends a slope in low light, proving that the terrain compressor can accurately extract slope terrain features and that the system functions well under low illumination.

V. Conclusion

We developed a hierarchical control system for quadrupedal robots that safely navigates cluttered and diverse terrains using

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

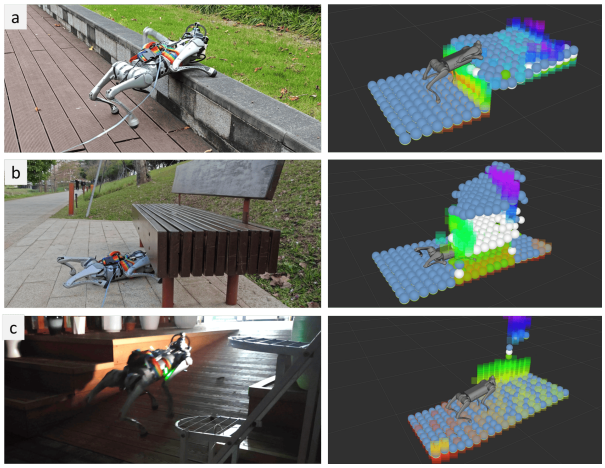

Fig. 10: **Deploying the System in Outdoor Environments** (a) Climbing up to a high step. (b) Crawling beneath a chair. (c) Walking on a slope at night.

a novel multi-layer elevation map representation. Our approach integrated terrain augmentation, knowledge distillation, and tailored reward functions to enhance policy generalization and maneuverability. Extensive simulations and real-world deployments on a low-cost robot demonstrated the robustness and practicality of the system. Nevertheless, the current terrain compressor is limited by its inability to learn directly from real-world data, whether in a manually supervised or self-supervised manner, which may restrict full adaptation to complex real-world terrain features. Our future work will focus on leveraging real-world data to further enhance the generalization of both the policy and perception modules.

REFERENCES

[1] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[2] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *Robotics: Science and Systems XVII*, 2021.

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

[4] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[5] Z. Zhuang, Z. Fu, J. Wang, C. G. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," in *Conference on Robot Learning*. PMLR, 2023, pp. 73–92.

[6] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 443–11 450.

[7] 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, 2024.

[8] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Conference on robot learning*. PMLR, 2023, pp. 403–415.

[9] C. Zhang, N. Rudin, D. Hoeller, and M. Hutter, "Learning agile locomotion on risky terrains," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 864–11 871.

[10] N. Rudin, J. He, J. Aurand, and M. Hutter, "Parkour in the wild: Learning a general and extensible agile locomotion policy using multi-expert distillation and rl fine-tuning," *arXiv preprint arXiv:2505.11164*, 2025.

[11] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2464–2470.

[12] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3402–3421, 2023.

[13] Z. Luo, Y. Dong, X. Li, R. Huang, Z. Shu, E. Xiao, and P. Lu, "Moral: Learning morphologically adaptive locomotion controller for quadrupedal robots on challenging terrains," *IEEE Robotics and Automation Letters*, 2024.

[14] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[15] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2497–2503.

[16] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter, "Learning robust autonomous navigation and locomotion for wheeled-legged robots," *Science Robotics*, vol. 9, no. 89, p. eadi9641, 2024.

[17] T. Miki, J. Lee, L. Wellhausen, and M. Hutter, "Learning to walk in confined spaces using 3d representation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8649–8656.

[18] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," in *Robotics: Science and Systems (RSS)*, 2024.

[19] R. Yang, G. Yang, and X. Wang, "Neural volumetric memory for visual locomotion control," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1430–1440.

[20] D. Hoeller, N. Rudin, C. Choy, A. Anandkumar, and M. Hutter, "Neural scene representation for locomotion on structured terrain," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8667–8674, 2022.

[21] L. Wellhausen and M. Hutter, "Rough terrain navigation for legged robots using reachability planning and template learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6914–6921.

[22] T. Dudzik, M. Chignoli, G. Bledt, B. Lim, A. Miller, D. Kim, and S. Kim, "Robust autonomous navigation of a small-scale quadruped robot in real-world environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3664–3671.

[23] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," in *Conference on Robot Learning*. PMLR, 2023, pp. 711–733.

[24] F. Yang, C. Wang, C. Cadena, and M. Hutter, "iplanner: Imperative path planning," *Proceedings of Robotics: Science and System XIX*, p. 064, 2023.

[25] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Robust and versatile bipedal jumping control through reinforcement learning," in *Robotics science and systems*. RSS, 2023.

[26] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li, "Multi-expert learning of adaptive legged locomotion," *Science Robotics*, vol. 5, no. 49, p. eabb2174, 2020.

[27] H. Kim, H. Oh, J. Park, Y. Kim, D. Youm, M. Jung, M. Lee, and J. Hwangbo, "High-speed control and navigation for quadrupedal robots on complex and discrete terrain," *Science Robotics*, vol. 10, no. 102, p. eads6192, 2025.

[28] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang, *et al.*, "Barkour: Benchmarking animal-level agility with quadruped robots," *arXiv preprint arXiv:2305.14654*, 2023.

[29] S. Ross, G. Gordon, and D. 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*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[30] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[31] Y. Ren, Y. Cai, F. Zhu, S. Liang, and F. Zhang, "Rog-map: An efficient robocentric occupancy grid map for large-scene and high-resolution lidar-based motion planning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 8119–8125.