IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

MARG: MAstering Risky Gap Terrains for Legged Robots with Elevation Mapping

Yinzhao Dong*, Ji Ma*, Liu Zhao, Wanyue Li, Peng Lu†

***Abstract*—Deep Reinforcement Learning (DRL) controllers for quadrupedal locomotion have demonstrated impressive performance on challenging terrains, allowing robots to execute complex skills such as climbing, running, and jumping. However, existing blind locomotion controllers often struggle to ensure safety and efficient traversal through risky gap terrains, which are typically highly complex, requiring robots to perceive terrain information and select appropriate footholds during locomotion accurately. Meanwhile, existing perception-based controllers still present several practical limitations, including a complex multi-sensor deployment system and expensive computing resource requirements. This paper proposes a DRL controller named MAstering Risky Gap Terrains (MARG), which integrates terrain maps and proprioception to dynamically adjust the action and enhance the robot's stability in these tasks. During the training phase, our controller accelerates policy optimization by selectively incorporating privileged information (e.g., center of mass, friction coefficients) that are available in simulation but unmeasurable directly in real-world deployments due to sensor limitations. We also designed three foot-related rewards to encourage the robot to explore safe footholds. More importantly, a terrain map generation (TMG) model is proposed to reduce the drift existing in mapping and provide accurate terrain maps using only one LiDAR, providing a foundation for zero-shot transfer of the learned policy. The experimental results indicate that MARG maintains stability in various risky terrain tasks.**

***Index Terms*—Legged Robots; Elevation Mapping; Deep Reinforcement Learning; Risky Gap Terrains.**

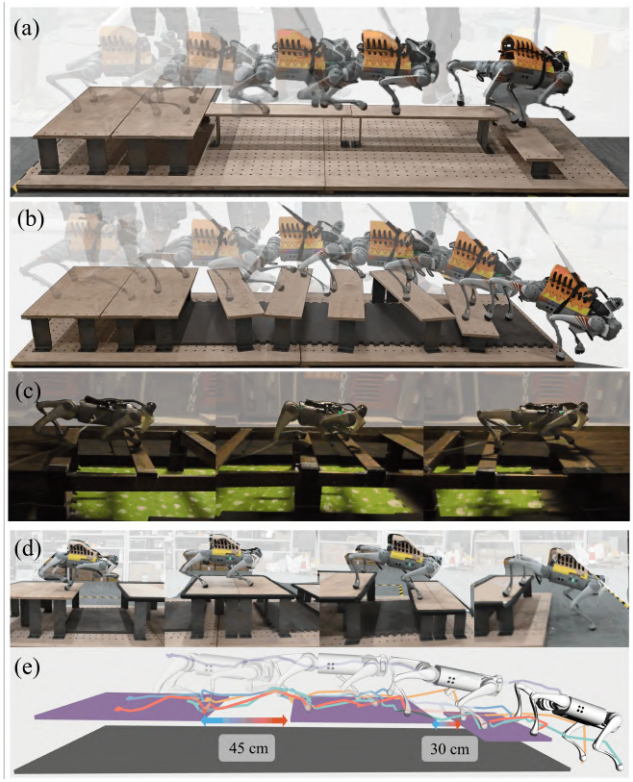


Fig. 1. Experiment of Unitree Go1 and Go2 on risky gap terrains, including (a) Single plank bridge, with the narrowest traversable width being 18 cm, validates the center-of-gravity control under narrow support. (b-c) Balance beams, where the narrowest beam width is 9 cm, to test the robot's stability in response to height variations, inclination changes, and edge perception. (d-e) Large gaps to demonstrate the capability to traverse gaps of varying widths (up to 65 cm in the real-world experiment).

I. Introduction

Legged robots have significantly advanced locomotion capabilities, demonstrating impressive skills across various movement modes, such as climbing stairs [1], [2], descending ramps [3], high-speed running [4], parkour [5], bipedal locomotion [6], and backflipping [7]. These abilities enable robots to perform well in continuous and highly challenging terrains, including rugged mountain paths, narrow passages, stairwells, slippery or unstable surfaces, etc. However, existing blind locomotion controllers often struggle to overcome risky gap terrains due to shortcomings in ensuring the safety and balance of quadruped robots.

Risky gap terrains exhibit numerous complex characteristics, imposing nearly stringent demands on robots regarding footholds and balance capabilities during locomotion. As shown in Fig. 1 (a), robots must not only strive to maintain the stability of their center of gravity on a narrow single-plank bridge but also respond in real time to potential lateral disturbances. Once a robot makes errors during locomotion, such as slipping or shifting its center of gravity, it may quickly step on the air or lose stability, leading to a fall and potentially causing severe damage to the robot. When traversing balance beams, the quadruped robot must not only accurately perceive terrain information such as height variations, gap width, and edges, but also select appropriate landing footholds and timing

Manuscript created 19 November 2024; Revised 31 July 2025; Accepted 20 September 2025. This work was supported by the General Research Fund under Grant 17204222, and in part by the Seed Fund for Collaborative Research and General Funding Scheme-HKU-TCL Joint Research Center for Artificial Intelligence. This paper was recommended for publication by Editor Jens Kober and Editor-in-Chief Wolfram Burgard upon evaluation of the reviewers' comments. (Corresponding author: Peng Lu).

The authors are with the Adaptive Robotic Controls Lab (ArcLab), Department of Mechanical Engineering, The University of Hong Kong, Hong Kong SAR, China (E-mail: dongyz@connect.hku.hk, maji@connect.hku.hk, zhaol@connect.hku.hk, liwy1024@connect.hku.hk, lupeng@hku.hk.

* Equal Contribution.

The supplementary video is available at https://youtu.be/NOVmjvWUM8Y

Digital Object Identifier (DOI): ras.tro.25-0713.aa6b0eec.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

for exertion based on its locomotion capabilities and current state to avoid missteps, as shown in Figs. 1 (b-c).

The majority of existing quadrupedal locomotion controllers are blind, which means that they do not utilize perception sensors like cameras and LiDARs [8], [9], [10], [11], [12]. It is nearly impossible for these controllers to traverse risky terrains as shown in Fig. 1. Recently, perception sensors have been used to obtain an elevation map of the environment [13], [1]. However, they do not take risky terrains into consideration. Only a few studies consider risky terrains, and they either rely on multiple sensors [14], which significantly increases the complexity of hardware deployment, or use motion capture systems to obtain prior information about the terrain [15]. In this paper, we only use one sensor to construct a robot-centered map and do not rely on motion capture systems.

A. Model-based Legged Locomotion

Existing model-based controllers rely on precise modeling of robots to calculate the optimal joint torques or footholds required for locomotion. For example, Singh et al. [16] compute second-order derivatives of rigid-body inverse and forward dynamics, achieving significant speed-ups over automatic differentiation in optimization-driven robot control. The CAFE-MPC framework [17] employs a cascaded-fidelity model predictive control scheme paired with a tuning-free whole-body controller, enabling quadruped robots to execute agile maneuvers without manual parameter tuning. Meduri et al. [18] splits the nonlinear MPC problem into biconvex centroidal dynamics and full-body kinematics, enabling real-time generation of dynamic whole-body motions for legged robots. These models can generate accurate control commands, enabling the robot to achieve stable and efficient locomotion in an ideal simulation and simple terrains [19]. However, uncertainty factors in real-world environments, such as terrain irregularity, changing friction, and external disturbances, present significant challenges to model-based methods. These factors are difficult to accurately incorporate into models, leading to potential mismatches between the model and the real world. Even slight discrepancies can cause robot locomotion failures, especially in risky gap terrains.

To address these challenges, researchers [20], [21] have attempted to simplify the dynamics model by utilizing Nonlinear Model Predictive Control (NMPC) to enhance the locomotion of robots in complex and dynamic environments. Yin et al. [22] propose an optimization algorithm to improve the robot's locomotion performance by transforming the discrete terrain height map into a continuous cost map to adjust the footholds dynamically. [23] proposes a novel control system that integrates adaptive control into a force-based control system for legged robots, enabling them to dynamically locomotion on uneven terrains. However, the computational complexity and slow convergence rates limit the applicability of robots in dynamic environments.

In addition, studies [24], [25] are also exploring the use of multiple sensors to enhance the accuracy and reliability of the model. Alongside the robot's inertial measurement unit (IMU), external perception devices such as depth cameras and LiDARs are employed to gather environmental information, including terrain width, height, and edge shape, and integrate this information into the dynamic model to assist robot control [26]. The synchronization of sensor data, the design of fusion algorithms for different sensor inputs, and the computational burden of data processing will further adversely affect the real-time control performance of robots.

B. Model-free Legged Locomotion

Model-free methods, such as deep reinforcement learning (DRL), have shown promise in enabling legged robots to adapt to complex terrains without relying on precise dynamic models. These methods focus on training robots to learn optimal policies through trial and error, allowing them to manage uncertainties and dynamic changes in their environment effectively [27]. The blind locomotion controllers [2], [10] have shown impressive progress in enabling robots to traverse challenging continuous terrains. However, these controllers often struggle in risky terrains due to the absence of environmental perception.

Integrating data from other external sensors, such as depth cameras, motion capture, etc, into the DRL framework is an effective way to help robots comprehensively understand their surrounding environment. Pioneering works [28], [5], [29] utilize deep learning models, such as GRU [30] and LSTM [31], to process depth images and extract terrain features, including height, slope, and distribution. Robots can successfully perform high-difficulty parkour tasks by incorporating these terrain factors into their decision-making processes. Challenges such as lighting changes, occlusion issues, high dimensionality, and complexity of images [32] may lead to inaccurate terrain feature extraction or high computational complexity during the training process, ultimately affecting the real-time performance of the robot [33]. Meanwhile, [34] and [15] use motion capture and an offline map to derive the height map around the robot's feet, which limits the practical applicability of this algorithm.

C. Elevation Mapping for Legged Robots

To obtain more accurate terrain information in the real world, previous DRL controllers [13], [35], [14] utilize multiple depth cameras or LiDARs simultaneously for elevation mapping, which can significantly enhance the accuracy of terrain representation. However, this approach increases the complexity of hardware deployment, as it requires sophisticated processing capabilities to handle the data from multiple sensors. Additionally, existing localization technologies [36], [37], [38], [39] heavily rely on the pose estimation of floating bases within the global frame. Any inaccuracy in this estimation may lead to map drift, thereby affecting the movement of legged robots in risky terrains. Thus, designing safe and reliable controllers, developing efficient algorithms to simplify deployment processes, and obtaining precise terrain maps remain challenging in risky gaps tasks.

In summary, we propose a DRL controller for quadrupedal locomotion—MAstering Risky Gap (MARG)—which integrates terrain maps, privileged information, and proprioceptive

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

into the policy to enhance the locomotion performance of quadrupedal robots in risky terrains. The key contributions of this work can be listed as follows:

- We propose a safe and robust robot controller for locomotion, which can predict the body velocity and the contact state of feet on each step, significantly enhancing the robot's stability in risky gap terrains.
- For risky tasks, we have designed three foot-related rewards: feet air time, feet stumble, and feet center, which promote the policy to explore safe footholds, enhancing the safety of movement.
- We propose a terrain map generation model that uses a single LiDAR to obtain the robot-centered height map. Our method minimizes drift compared to the traditional localization approaches while achieving zero-shot transfer capability and optimal computational efficiency.
- The MARG controller empowers quadruped robots to adeptly handle risky gap terrains in the real world, including 65 cm large gaps, 18 cm narrow single-plank bridges, and balance beams with varying sizes, heights, and inclinations.

II. MARG Locomotion Controller

A. Problem Formulation

Terrain-aware legged locomotion task is a type of sequential decision-making problem under uncertainty. However, even with exteroceptive sensors, such as LiDARs and depth cameras, certain privileged information (e.g., the body velocity, the mass of each link, and the friction) still cannot be accurately obtained in real robots. Therefore, our problem can be formulated as an infinite-horizon Partially Observable Markov Decision Process (POMDP), denoted as a 7-tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \Omega, \gamma\}$. Here, $\mathcal{S}$, $\mathcal{O}$, and $\mathcal{A}$ are the sets of states, observations, and actions, respectively. For each state $\boldsymbol{s}_t \in \mathcal{S}$, the learning agent interacts with the environment with an action $\boldsymbol{a}_t \in \mathcal{A}$ and receives a reward $\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$, leading to the transition of the environment to the next state $\boldsymbol{s}_{t+1}$ with the probability $\mathcal{P}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. Meanwhile, the observation $\boldsymbol{o}_{t+1} \in \mathcal{O}$ depends on the new state $\boldsymbol{s}_{t+1}$ and the action $\boldsymbol{a}_t$ with a conditional probability $\Omega(\boldsymbol{o}_{t+1}|\boldsymbol{s}_{t+1}, \boldsymbol{a}_t)$. The objective is to determine the optimal policy π^* that maximizes the accumulated rewards.

1) Observation Space: The proprioceptive observations $\widetilde{\boldsymbol{o}}_t = (\boldsymbol{\omega}_t,\ \boldsymbol{g}_t,\ \boldsymbol{v}_t^*,\ \boldsymbol{q}_t,\ \dot{\boldsymbol{q}}_t,\ \boldsymbol{a}_{t-1})$ in our task include body angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$, projected gravity $\boldsymbol{g}_t \in \mathbb{R}^3$, linear velocity command $\boldsymbol{v}_t^* \in \mathbb{R}^3$, joint angles $\boldsymbol{q}_t \in \mathbb{R}^{12}$, joint angular velocities $\dot{\boldsymbol{q}}_t \in \mathbb{R}^{12}$, and the action of the last step $\boldsymbol{a}_{t-1} \in \mathbb{R}^{12}$. Meanwhile, we define a temporal observations $\widetilde{\boldsymbol{o}}_t^H = [\widetilde{\boldsymbol{o}}_t, \widetilde{\boldsymbol{o}}_{t-1}, ..., \widetilde{\boldsymbol{o}}_{t-H}]$ to store the proprioceptive observations over the past H time step ($H = 5$ in this task). In addition, we also collect the exteroception observations (i.e., the egocentric terrain map of the robot body $\boldsymbol{h}_t \in \mathbb{R}^{187}$) and the privileged state $\widetilde{\boldsymbol{s}}_t$ defined as:

$$\widetilde{\boldsymbol{s}}_t = (\boldsymbol{v}_t,\ \boldsymbol{c}_t,\ \boldsymbol{m},\ \boldsymbol{\mu},\ \boldsymbol{\zeta},\ \boldsymbol{f}_t,\ \boldsymbol{k}_{pd}) \tag{1}$$

where $\boldsymbol{v}_t \in \mathbb{R}^3$, $\boldsymbol{c}_t \in \mathbb{R}^4$, $\boldsymbol{m} \in \mathbb{R}^4$, $\boldsymbol{\mu} \in \mathbb{R}^1$, $\boldsymbol{\zeta} \in \mathbb{R}^2$, and $\boldsymbol{f}_t \in \mathbb{R}^2$ denote the real linear velocity, the contact boolean of all feet, the critical links' masses (like trunk, thigh, and calf), the body friction coefficient, the center of mass in body space, and the disturbance force projection in x-o-y plane, respectively. $\boldsymbol{k}_{pd} \in \mathbb{R}^{26}$ includes the proportional gain $\boldsymbol{k}_p \in \mathbb{R}^1$, the derivative gain $\boldsymbol{k}_d \in \mathbb{R}^1$, the motor strength of each joint $\boldsymbol{\alpha} \in \mathbb{R}^{12}$ and the motor offset of each joint $\Delta\boldsymbol{q} \in \mathbb{R}^{12}$.

2) Action Space: The action $\boldsymbol{a}_t \in \mathbb{R}^{12}$ represents the desired increment of the joint angle w.r.t the initial pose $\mathring{\boldsymbol{q}}$, i.e. $\boldsymbol{q}_t^* = \mathring{\boldsymbol{q}} + \boldsymbol{a}_t$. The final desired angle $\boldsymbol{q}_t^*$ is tracked by the torque generated by the joint-level proportional–derivative (PD) controller of the actuation module in the simulator.

B. Neural Network design

As shown in Fig. 2, the MAstering Risky Gap (MARG) controller consists of four sub-networks: an actor net, a critic net, an estimator net, and an elevation net. Each component is crucial in enabling the controller to adapt in real-time to the challenges posed by risky gap terrains. Next, we will discuss each part of the framework in detail.

1) Learn to Extract the Critical Features: The ability of our controller to master risky terrains can be attributed to two networks: an elevation net that extracts elevation features $\boldsymbol{e}_t^h \in \mathbb{R}^{16}$ and an estimator net that predicts the critical privilege features $\boldsymbol{e}_t^o \in \mathbb{R}^7$. These features are utilized by actor and critic networks to make informed decisions, ensuring locomotion safely and efficiently under risky terrains.

The elevation net $\boldsymbol{E}_{\boldsymbol{\theta}_1}$ is designed to extract the elevation features $\boldsymbol{e}_t^h$ from the robot-centered area of 1.6 m × 1.0 m, which enable the robot to gain a comprehensive understanding of the height variations and contours, thereby facilitating its identification of safe zones. By incorporating the $\boldsymbol{e}_t^h$ into the learning process, the robot can better adapt to risky environments and enhance its locomotion ability. This network employs a Multilayer Perceptron (MLP) architecture to extract the terrain features $\boldsymbol{e}_t^h$. Specifically, we utilize the relative height map $\hat{\boldsymbol{h}}_t$ between the robot's base $\boldsymbol{z}_t$ and the surrounding terrain $\boldsymbol{h}_t$ as input, as follows:

$$\boldsymbol{e}_t^h = \boldsymbol{E}_{\boldsymbol{\theta}_1}(\hat{\boldsymbol{h}}_t) = \boldsymbol{E}_{\boldsymbol{\theta}_1}(\boldsymbol{z}_t - \boldsymbol{h}_t). \tag{2}$$

The estimator net $\boldsymbol{E}_{\boldsymbol{\theta}_2}$ aims to process the history of observations $\widetilde{\boldsymbol{o}}_t^H$ and extract the critical privilege features $\boldsymbol{e}_t^o$ (including the estimated linear velocity $\hat{\boldsymbol{v}}_t$ and the estimated contact boolean of all feet $\hat{\boldsymbol{c}}_t$), which plays a vital role in enhancing the robot's stability and robustness during locomotion on risky terrains, particularly when encountering gaps or obstacles.

$$\boldsymbol{e}_t^o = \boldsymbol{E}_{\boldsymbol{\theta}_2}(\widetilde{\boldsymbol{o}}_t^{\boldsymbol{H}}) = (\hat{\boldsymbol{v}}_t, \hat{\boldsymbol{c}}_t) \tag{3}$$

2) Asymmetric Actor-Critic: Certain privileged information, in addition to terrain maps, can significantly enhance the stability and robustness of the robot's locomotion ability under risky gaps. Thus, our controller adopts the asymmetric actor-critic structure for terrain-aware robot learning, enabling more effective exploration under high-dimensional spaces of legged robots. The actor and critic learn from distinct objectives: the former aims to maximize expected rewards, while the latter minimizes the difference between predicted and actual values.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

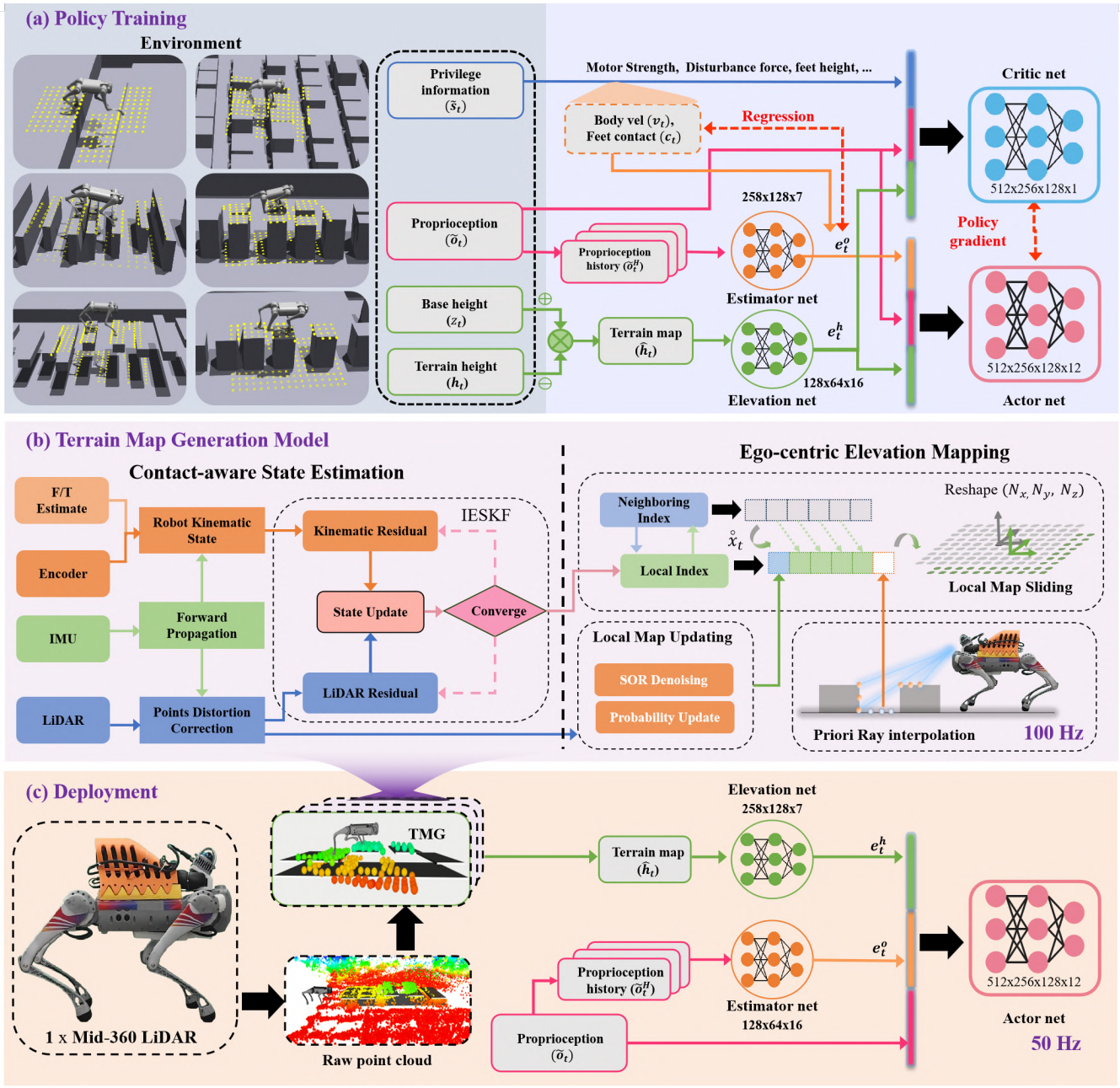


Fig. 2. The framework of the proposed MARG, showing (a) Policy training module including actor, critic, estimator, and elevation nets, along with their data flow; (b) Terrain map generation module (TMG) including the processes of state estimation, mapping, and interpolation; (c) Deployment module indicating how the MARG is implemented in the real-world with a Mid-360 LiDAR inputs and network operations.

The actor net is designed to derive the action $\boldsymbol{a}_t$ at each time step, which integrates several inputs to create a comprehensive representation of the current state. By leveraging multiple sources of information, the actor net can better adapt its actions to dynamic environmental conditions, allowing it to make informed decisions. The combined input vector $\boldsymbol{o}_t$ includes the proprioception observation $\widetilde{\boldsymbol{o}}_t$, the critical privilege features $\boldsymbol{e}_t^o$, and the elevation features $\boldsymbol{e}_t^h$, defined as follows:

$$\boldsymbol{o}_t = \left[\widetilde{\boldsymbol{o}}_t,\ \boldsymbol{e}_t^o,\ \boldsymbol{e}_t^h\right] \tag{4}$$

Except for the proprioception observation and the elevation features, the critic net also integrates more ground-truth physical knowledge $\widetilde{\boldsymbol{s}}_t$ about the environment, which enables this net to more accurately assess the value of the current state and provide useful feedback to the actor net. The input vector $\boldsymbol{s}_t$ of the value net can be organized as follows:

$$\boldsymbol{s}_t = \left[\widetilde{\boldsymbol{o}}_t,\ \widetilde{\boldsymbol{s}}_t,\ \boldsymbol{e}_t^h\right]. \tag{5}$$

3) Concurrent Training: These four networks in MARG are trained together by Proximal Policy Optimization [40] in simulation to achieve real-time adaptation in risky gap terrains. During training, the policy gradient $loss_{policy}$ can be backpropagated through the actor net to update its parameters of the estimator and elevation net, allowing them to learn from

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

the rewards received during interactions with the environment. Simultaneously, the value function $loss_{value}$ is computed to minimize the error between the predicted value and the actual returns, which is also backpropagated to update the critic net. Meanwhile, the training of the estimator net is not independent of the training of the actor net, the parameters of the estimator net are also updated via a regression loss $loss_{reg}$ to reduce the Mean Squared Error (MSE), as follows:

$$loss_{reg} = MSE(\hat{\boldsymbol{v}}_t, \boldsymbol{v}_t) + MSE(\hat{\boldsymbol{c}}_t, \boldsymbol{c}_t) \tag{6}$$

C. Rewards for Footholds

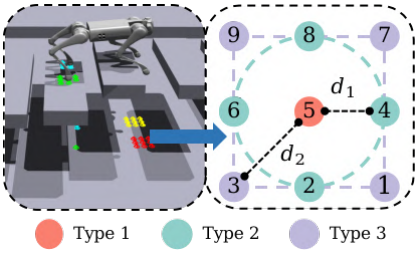


Fig. 3. The feet center reward calculation: points around the robotic foot are categorized as Type 1 (at contact), Type 2 (within d_1 radius), and Type 3 (within d_2 radius).

The reward functions are designed to track the command velocity, penalize the unsmoothness of robot locomotion, avoid collisions, constrain the joint motion of quadrupeds, and optimize footholds, as shown in Table I. To enable safe footholds on risky terrains, we design three foot-related rewards: feet air time, feet stumble, and feet center.

Feet air time encourages the leg lifting time during movement, helping the robot cross uneven terrain. We calculate it as $\sum_{f=0}^{4}(\boldsymbol{t}_{air,f} - 0.5)$, where $\boldsymbol{t}_{air,f}$ represents the duration each foot stays in the air (i.e., the time when it is not in contact with any terrain).

Feet stumble is a penalty for feet hitting vertical surfaces, which can prevent the robot from getting stuck or tripping during its movement. In the Isaac Gym simulator [41], we can easily obtain the contact force of each body at any time. By extracting the corresponding indices of each foot, we can accurately obtain the contact forces $\boldsymbol{f}_{x,y,z} = [\boldsymbol{f}_x, \boldsymbol{f}_y, \boldsymbol{f}_z]$ in the x, y, and z directions. Subsequently, we use these components to define the reward, as follows: any $\left(\|\boldsymbol{f}_{x,y}\| > 4 \cdot |\boldsymbol{f}_z|\right)$, where $\|\boldsymbol{f}_{x,y}\|$ represents the magnitude of the horizontal contact force and $|\boldsymbol{f}_z|$ denotes the absolute value of the vertical contact force.

The feet center imposes a penalty for each step on the edges, helping the robot to choose safer areas for locomotion. Specifically, we select 9 points around each foot end, as shown in Fig. 3. We classify the points around the foot into three types based on their distance from the foot. The contact position of the foot (i.e., $id = 5$) is classified as type 1. Points within a radius $d_1 = 5$ cm from the foot are type 2, while points within a circle of radius $d_2 = \sqrt{50}$ cm from the foot end position are type 3. We can determine whether the foot is on the edge based on the heights of the points in each type, as follows:

$$n_t^i = \begin{cases} 1, & \text{if } id \in \text{Type } i \text{ and } h_{id} < -0.2 \\ 0, & \text{else} \end{cases} \tag{7}$$

where id and h_{id} denote the index and height of this point.

TABLE I
REWARD TERMS.

Term	Reward	Equation	Weight
Task	Lin. velocity tracking	$e^{-4\|\|\boldsymbol{v}^*_{xy} - \boldsymbol{v}_{xy}\|\|^2}$	1.0
	Ang. velocity tracking	$e^{-4(\boldsymbol{w}^*_{yaw} - \boldsymbol{w}_{yaw})^2}$	0.5
Smoothness	Linear velocity (z)	$\boldsymbol{v}_z^2$	-2.0
	Angular velocity (xy)	$\|\|\boldsymbol{w}_{xy}\|\|^2$	-0.05
	Joint torque	$\|\|\boldsymbol{\tau}\|\|^2$	$-e^{-5}$
	Action rate	$\|\|\boldsymbol{a}_t - \boldsymbol{a}_{t-1}\|\|^2$	-0.01
	Joint accelerations	$\|\|\ddot{\boldsymbol{q}}\|\|^2$	$-2.5e^{-7}$
Safety	Collisions	$-n_{collision}$	1.0
Pose	Orientation	$\|\|\boldsymbol{q}_{xy}\|\|^2$	-0.2
	Joint motion limit	$\sum_{j=0}^{12}\|\|\boldsymbol{q}_{t,j} - \mathring{\boldsymbol{q}}_j\|\|$	-0.02
Footholds	Feet air time	$\sum_{f=0}^{4}(\boldsymbol{t}_{air,f} - 0.5)$	1.0
	Feet stumble	any $\left(\|\boldsymbol{f}_{x,y}\| > 4 \cdot \|\boldsymbol{f}_z\|\right)$	-1.0
	Feet center	$c_t \cdot (n_t^2 + 2 * n_t^3)$	-0.01

III. Terrain Map Generation Model

An accurate terrain map $\hat{\boldsymbol{h}}_t$ is critical for the successful deployment of our controller in real-world environments. Previous approaches have relied heavily on accurate global pose estimation [42] and clean point cloud inputs [15] to accumulate elevation information. Furthermore, current deployment solutions typically require multiple external sensors, such as arrays of cameras or LiDAR systems, which substantially increase the complexity and cost of practical implementation [14] [43].

In this section, we only use one LiDAR sensor for the perception of risky terrains, which significantly reduces the complexity of hardware deployment. However, there are several challenges with LiDAR-based localization and mapping. First, typical LiDAR-based localization methods often suffer from height drift, which can be detrimental for quadrupedal locomotion as the height estimate is used to construct a height map. Furthermore, LiDAR-based mapping maintains a local map by employing a global sliding window, which is computationally intensive.

To address these limitations, we propose a Terrain Map Generation (TMG) model that incorporates kinematic measurements and a hash-based local map to provide accurate and computationally efficient elevation information. In the following sections, we will elaborate on the key components of our model.

A. Contact-aware State Estimation

We first estimate the transformation between two neighboring frames using an Error State Kalman Filter (ESKF) approach, which will be used later for constructing an ego-centric elevation map. Existing LiDAR-based localization methods [39], [37] have drift issues and cannot be directly implemented.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

To address these, we incorporate the contact point positions $\boldsymbol{p}_{f_i} \in \mathbb{R}^3$ in the state vector.

The ground-truth state is defined as a tuple $\boldsymbol{x} \in SO(3)\times\mathbb{R}^{27}$ comprising the following states:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{R}_{wb} & \boldsymbol{p}_{wb} & \boldsymbol{v}_{wb} & \boldsymbol{b}_a & \boldsymbol{b}_\omega & \boldsymbol{p}_{f_1}...\boldsymbol{p}_{f_4} & \boldsymbol{g} \end{bmatrix} \tag{8}$$

where $\boldsymbol{R}_{wb} \in SO(3)$ denotes the rotation matrix from the body frame to the world frame, while $\boldsymbol{p}_{wb} \in \mathbb{R}^3$ and $\boldsymbol{v}_{wb} \in \mathbb{R}^3$ represent the position and velocity of the body frame in the world frame, respectively. The terms $\boldsymbol{b}_a$ and $\boldsymbol{b}_\omega$ correspond to accelerometer and gyroscope biases, while $\boldsymbol{g}$ represents the gravity vector in the world frame that requires initialization. To incorporate the kinematic properties of the robot into the estimation, we augment the state vector with the positions of contact points $\boldsymbol{p}_{f_i}$, enabling us to mitigate positional drift in the estimation process.

1) Forward Propagation: Considering the state transition from time step t to $t+1$ and neglecting the noise terms ($\boldsymbol{n}_a$, $\boldsymbol{n}_\omega$, and $\boldsymbol{n}_{p_{fi}}$), the forward propagation can be governed by:

$$\bar{\boldsymbol{x}}_{t+1} = \bar{\boldsymbol{x}}_t \boxplus (\Phi(\bar{\boldsymbol{x}}_t, \boldsymbol{u}_t, 0)\Delta t) \tag{9}$$

$$\Phi(\bar{\boldsymbol{x}}_t, \boldsymbol{u}_t, \boldsymbol{n}_t) = \begin{bmatrix} \boldsymbol{\omega}_{m_t} - \boldsymbol{b}_{\omega_t} - \boldsymbol{n}_{\omega_t} \\ \boldsymbol{v}_{wb_t} \\ \boldsymbol{R}_{wb_t}(\boldsymbol{a}_{m_t} - \boldsymbol{b}_{a_t} - \boldsymbol{n}_{a_t}) + \boldsymbol{g}_t \\ \boldsymbol{n}_{ba_t} \\ \boldsymbol{n}_{b\omega_t} \\ \boldsymbol{n}_{p_{f_{i,t}}} \\ 0_{3\times1} \end{bmatrix} \tag{10}$$

where $\bar{\boldsymbol{x}}_{\boldsymbol{t}}$ represents the propagated state and Φ denotes the forward propagation function. The IMU measurements $\boldsymbol{u}$ are influenced by Gaussian noise terms: $\boldsymbol{n}_a$ for acceleration and $\boldsymbol{n}_\omega$ for angular velocity. During the swing phase, the foot position is affected by white noise $\boldsymbol{n}_{p_{f_i}}$. Due to the lack of ground contact, the uncertainty in foot position increases significantly. This is typically modeled by assigning a large variance to this process noise.

Let the measurement input vector be $\boldsymbol{u} = [\boldsymbol{\omega}_m \ \boldsymbol{a}_m]$ and the noise vector be $\boldsymbol{n} = \begin{bmatrix} \boldsymbol{n}_a & \boldsymbol{n}_\omega & \boldsymbol{n}_{b\omega} & \boldsymbol{n}_{ba} & \boldsymbol{n}_{p_{f_i}} \end{bmatrix}$. The components $\boldsymbol{n}_{b\omega}$ and $\boldsymbol{n}_{ba}$ correspond to the random walk noise for the IMU biases $\boldsymbol{n}_{\boldsymbol{\omega}}$ and $\boldsymbol{n}_{\boldsymbol{a}}$, respectively. Furthermore, the error state $\widetilde{\boldsymbol{x}}$ is characterized by:

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} \boxminus \bar{\boldsymbol{x}} \approx \boldsymbol{F}_{\widetilde{x}}\widetilde{\boldsymbol{x}} + \boldsymbol{F}_n\boldsymbol{n} \tag{11}$$

where $\boldsymbol{F}_{\widetilde{x}}$ and $\boldsymbol{F}_n$ represent the Jacobian matrices of Φ with respect to $\widetilde{\boldsymbol{x}}$ and $\boldsymbol{n}$, respectively.

2) LiDAR Measurement: For every LiDAR points ${}^L\boldsymbol{p}_j$, we model the LiDAR measurement $\boldsymbol{h}_j({}^L\boldsymbol{p}_j + {}^L\boldsymbol{n}_j)$ as:

$$0 = \boldsymbol{u}_j^T({}^G\boldsymbol{T}_I \ {}^I\boldsymbol{T}_L({}^L\boldsymbol{p}_j + {}^L\boldsymbol{n}_j) - {}^G\boldsymbol{q}_j) \tag{12}$$

where $\boldsymbol{u}_j$ denotes the unit vector of the corresponding plane, ${}^G\boldsymbol{T}_I$ represents the transformation from the IMU frame to the world frame, where the rotation component ${}^G\boldsymbol{T}_I$ equals $\boldsymbol{R}_{wb}$ since we assume that the IMU frame coincides with the body frame, and ${}^I\boldsymbol{T}_L = ({}^I\boldsymbol{R}_L, {}^I\boldsymbol{t}_L)$ is the extrinsic transformation from LiDAR frame to IMU frame. ${}^G\boldsymbol{q}_j$ represents a point on the corresponding plane, and ${}^L\boldsymbol{n}_j$ denotes the LiDAR measurement noise. The linearized measurement model is given by:

$$0 \simeq \boldsymbol{h}_j(\bar{\boldsymbol{x}}_t, \boldsymbol{0}) + \boldsymbol{H}_j\widetilde{\boldsymbol{x}}_t + \boldsymbol{n}_j \tag{13}$$

$$\begin{aligned} \boldsymbol{H}_j &= \left.\frac{\partial \boldsymbol{h}_j(\bar{\boldsymbol{x}}_t \boxplus \widetilde{\boldsymbol{x}}_t, \boldsymbol{0})}{\partial \widetilde{\boldsymbol{x}}_t}\right|\widetilde{\boldsymbol{x}}_t = \boldsymbol{0} \\ &= \boldsymbol{u}_j^T \begin{bmatrix} -\bar{\boldsymbol{R}}_{wb}({}^I\boldsymbol{R}_L{}^L\boldsymbol{p}_j + {}^I\boldsymbol{t}_L)^\wedge & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times21} \end{bmatrix} \end{aligned} \tag{14}$$

where $\boldsymbol{H}_j$ is the Jacobian matrix of $\boldsymbol{h}_j$ with respect to $\widetilde{\boldsymbol{x}}$, and $\boldsymbol{n}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_j)$ represents the LiDAR measurement noise.

3) Kinematics Measurement: The kinematics measurement can be defined as:

$$\boldsymbol{p}_{f_i^{rel}} = \boldsymbol{R}_{wb}^T \cdot (\boldsymbol{p}_{wb} - \boldsymbol{p}_{f_i}) \tag{15}$$

where the term $\boldsymbol{p}_{f_i^{rel}}$ represents the measured relative position of the foot contact point $\boldsymbol{p}_{f_i}$ in the body frame. Assuming that there is no relative sliding between the contact point and the ground, the velocity of the foot can be assumed as zero, and the residuals of contacted foot velocity $\boldsymbol{h}_{cv}$ and position $\boldsymbol{h}_{cp}$ can be defined as:

$$\boldsymbol{h}_{cv} = \boldsymbol{v}_{wb} + \boldsymbol{R}_{wb} \cdot (\boldsymbol{v}_{f_i^{rel}} + (\boldsymbol{\omega}_m - \boldsymbol{b}_\omega) \times \boldsymbol{p}_{f_i^{rel}}) \tag{16}$$

$$\boldsymbol{h}_{cp} = \boldsymbol{p}_{f_i^{rel}} - \boldsymbol{R}_{wb}^T \cdot (\boldsymbol{p}_{wb} - \boldsymbol{p}_{f_i}) \tag{17}$$

$$\begin{aligned} \boldsymbol{H}_{cv} &= \left.\frac{\partial \boldsymbol{h}_{cv}(\bar{\boldsymbol{x}}_t \boxplus \widetilde{\boldsymbol{x}}_t, \boldsymbol{0})}{\partial \widetilde{\boldsymbol{x}}_t}\right|\widetilde{\boldsymbol{x}}_t = \boldsymbol{0} \\ &= \begin{bmatrix} -\bar{\boldsymbol{R}}_{wb} \cdot (\boldsymbol{v}_{f_i^{rel}} + (\boldsymbol{\omega}_m - \bar{\boldsymbol{b}}_\omega)^\wedge \boldsymbol{p}_{f_i^{rel}})^\wedge & \boldsymbol{0}_{3\times3} \\ \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix} \\ &\quad \begin{bmatrix} \bar{\boldsymbol{R}}_{wb} \cdot \boldsymbol{p}_{f_i^{rel}}^\wedge & \boldsymbol{0}_{3\times12} \end{bmatrix} \end{aligned} \tag{18}$$

$$\begin{aligned} \boldsymbol{H}_{cp} &= \left.\frac{\partial \boldsymbol{h}_{cp}(\bar{\boldsymbol{x}}_t \boxplus \widetilde{\boldsymbol{x}}_t, \boldsymbol{0})}{\partial \widetilde{\boldsymbol{x}}_t}\right|\widetilde{\boldsymbol{x}}_t = \boldsymbol{0} \\ &= \begin{bmatrix} \bar{\boldsymbol{R}}_{wb} \cdot (\boldsymbol{p}_{f_i^{rel}})^\wedge & -\boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times9} & \boldsymbol{I}_{3\times12} \end{bmatrix} \end{aligned} \tag{19}$$

where $\boldsymbol{H}_{cv}$ and $\boldsymbol{H}_{cp}$ are the Jacobian matrices. $\boldsymbol{v}_{f_i^{rel}}$ denotes the measured relative foot velocity. The terms $\bar{\boldsymbol{R}}_{wb}$ and $\bar{\boldsymbol{b}}_w$ represent the propagated state of rotation from the body frame to the world frame and IMU gyroscope bias.

4) State Update: Following the conventional IESKF process [39], we formulate the state estimation problem as a Maximum A Posteriori (MAP) estimation:

$$\begin{aligned} \text{minimize}_{\widetilde{\boldsymbol{x}}_t}(&\|\boldsymbol{x}_t \boxminus \bar{\boldsymbol{x}}_t\|^2_{\bar{\boldsymbol{P}}_t^{-1}} + \sum_{j=1}^{m} \|\boldsymbol{h}_j + \boldsymbol{H}_j\tilde{\boldsymbol{x}}_t\|^2_{\boldsymbol{R}_j^{-1}} \\ &+ \|\boldsymbol{h}_{cv}(\bar{\boldsymbol{x}}_t, \boldsymbol{u}_t, 0, 0) + \boldsymbol{H}_{cv}\tilde{\boldsymbol{x}}_t\|^2_{\boldsymbol{\Sigma}_{cv}^{-1}} \\ &+ \|\boldsymbol{h}_{cp}(\bar{\boldsymbol{x}}_t, \boldsymbol{u}_t, 0) + \boldsymbol{H}_{cp}\tilde{\boldsymbol{x}}_t\|^2_{\boldsymbol{\Sigma}_{cp}^{-1}}) \end{aligned} \tag{20}$$

The optimal state estimate is obtained through the Kalman gain $\boldsymbol{K}_t$ and the corresponding state and covariance updates [39]. Finally, the neighboring state $\mathring{\boldsymbol{x}}_t$ is updated through:

$$\mathring{\boldsymbol{x}}_t = \Phi(\bar{\boldsymbol{x}}_t, \boldsymbol{u}_t, 0)\Delta t \boxplus \boldsymbol{K}_t \begin{bmatrix} \boldsymbol{h}_1 & \dots & \boldsymbol{h}_m & \boldsymbol{h}_{cv} & \boldsymbol{h}_{cp} \end{bmatrix}^T \tag{21}$$

This updated state estimate can subsequently be utilized to refine the local map representation.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

B. Ego-centric Elevation Mapping

Conventional elevation mapping maintains a global sliding window tracking absolute poses relative to a fixed origin, updating the entire map as the robot moves. This global approach has two key limitations: high computational overhead and deteriorating reliability with global odometry errors [44].

Confronting these challenges, we propose an ego-centric mapping strategy focusing on the robot's immediate vicinity. Using the local state estimate $\mathring{\boldsymbol{x}}_t$, our approach enables precise local updates while reducing both computational cost and dependency on global pose estimation accuracy.

1) Local Map Sliding: We employ a hashmap-based approach for zero-copy local map sliding, maintaining a local elevation map of dimensions $\boldsymbol{L} \in \mathbb{R}^3$ with resolution $\boldsymbol{r}$, discretized into $\boldsymbol{N}$ cells. Each cell is referenced by its global index $\boldsymbol{g}_i = (g_{i_x}, g_{i_y}, g_{i_z})$, which is hashed based on its position relative to the body frame. We also maintain a local index $\boldsymbol{l}_i = (l_{i_x}, l_{i_y}, l_{i_z})$ within the map dimensions.

As the robot traverses the environment, we update the global indices according to:

$$\boldsymbol{g}_i = \mathring{\boldsymbol{R}}_{wb}\boldsymbol{N}(\boldsymbol{g}_i) + \mathring{\boldsymbol{p}}_{wb} \oslash \boldsymbol{r} \tag{22}$$

where $\mathring{\boldsymbol{R}}_{wb} \in SO(3)$ and $\mathring{\boldsymbol{p}}_{wb} \in \mathbb{R}^3$ denote the incremental rotation and translation from the body frame to the world frame, and $\oslash$ represents the element-wise division operator. The local indices are obtained by normalizing the global indices to ensure they remain within the elevation map:

$$\boldsymbol{l}_i = \text{normalize}(\boldsymbol{g}_i, \boldsymbol{L}) \tag{23}$$

At each time step, the grid cells are incrementally updated using LiDAR measurements to maintain an accurate representation of the local environment.

2) Local Map Updating: We implement a systematic approach for updating the local map grid to handle the beam divergence characteristics of LiDAR sensors. Initially, we employ Statistical Outlier Removal (SOR) to filter measurement noise from the point cloud data.

Then, we maintain a cached frame $\boldsymbol{C}$ to track grid occupancy states through ray casting [45] and store probabilistic occupancy information. For a point $\boldsymbol{p}$ in the LiDAR frame where $\boldsymbol{x}_p \in \mathbb{R}^3$, and its corresponding grid cell $\boldsymbol{g}_p$ in the local map, we update the occupancy probability using a logarithmic odds formulation:

$$\begin{aligned}\boldsymbol{C}pro|_t = \boldsymbol{C}pro|_{t-1} + n_{hit}\log\left(\frac{p_{hit}}{1-p_{hit}}\right) + \\ n_{miss}\log\left(\frac{p_{miss}}{1-p_{miss}}\right)\end{aligned} \tag{24}$$

where n_{hit} and n_{miss} represent the number of hit and miss points within the grid cell, respectively. p_{hit} and p_{miss} denote their corresponding probability parameters.

To effectively manage both dynamic and static objects in the environment, we implement probability bounds using lower T_{low} and upper T_{high} thresholds:

$$\boldsymbol{C}pro|_t = \max(\min(\boldsymbol{C}pro|_t, T_{high}), T_{low}) \tag{25}$$

This bounded update ensures robust occupancy estimation while maintaining adaptability to environmental changes.

3) Priori Ray Interpolation: To extract a relative terrain map $\hat{\boldsymbol{h}}_t$, we determine the highest occupied voxel height, denoted as $\hat{\boldsymbol{h}}(x_i, y_i)$, within each column of the occupancy grid representation's horizontal plane. Moreover, we implement a bidirectional ray-based interpolation strategy to replenish regions with missing data, where empty columns are assigned elevation values through both forward and reverse traversals. Specifically, during forward traversal, the elevation value $\hat{\boldsymbol{h}}(x_f, y_f)$ is obtained from the farthest detected occupied cell at distance d_{far}, while in reverse traversal, the elevation value $\hat{\boldsymbol{h}}(x_n, y_n)$ is taken from the nearest occupied cell at distance d_{near}. As depicted in Fig. 2, this dual-direction interpolation effectively maintains terrain continuity while preserving critical structural features. Finally, the relative terrain map $\hat{\boldsymbol{h}}_t$ is fed into the elevation net to generate the elevation feature $\boldsymbol{e}_t^h$.

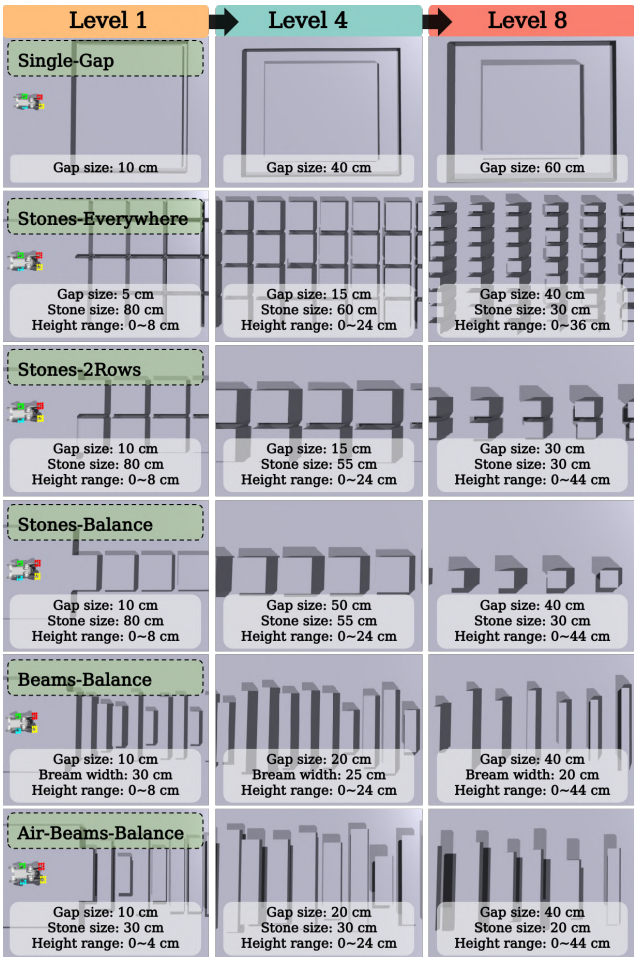


Fig. 4. The gap terrains of 8 different levels used in the experiments, such as Single-Gap, Stones-Everywhere, Stones-2Rows, Stones-Balance, Breams-Balance, and Air-Beams-Balance terrains. As the level increases, the gaps are enlarged, and the width and height of stones or beams are adjusted.

IV. Experimental Steup

A. Simulation

In this paper, we trained 4096 environments using Unitree Go1 with diverse domain randomization in parallel on the Isaac Gym simulator [41] for 20000 episodes, and each episode was terminated and reset under specific criteria, which

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

TABLE II
THE RANDOMIZATION RANGE OF PARAMETERS.

Parameters	Range	Unit
$\boldsymbol{K}_p$ factor	[0.9, 1.1]	Nms/rad
$\boldsymbol{K}_d$ factor	[0.9, 11]	Nms/rad
Payload	[-1.0, 2.0)	kg
System delay	[0.0, 12]	ms
Frictions coefficient	[0.2, 1.25)	-
Center of mass shift	[-0.05, 0.05)	m
Motor strength factor	[0.9, 1.1]	Nm
Noise of the terrain map h_t	[-5, 5)	cm

included collisions of the robot's trunk or hips with the terrain, or the height of each foot falling below -0.2 m, or being trapped for 20 seconds. The randomized parameters of the environment are randomized at the initialization stage, as shown in Table II. We utilized a game-inspired curriculum [46] to ensure progressive locomotion policy learning over risky terrains, as shown in Fig. 4. The terrains consisted of Single-Gap, Stones-Everywhere, Stones-2Rows, Stones-Balance, Breams-Balance, and Air-Beams-Balance terrains with 8 different levels. As the level increased, we progressively enlarged the gaps while reducing the width and elevating the height of the stones or beams.

The actor and critic networks for all controllers were trained together using PPO [40], and the architectures and the key parameter settings (such as clipping range, learning rate, etc.) were listed in [47]. In contrast, the architecture of the Estimator and Elevation networks of MARG were $[258, 128, 7]$ and $[128, 64, 16]$, respectively, utilizing Rectified Linear Units (ReLU) as the activation function. The entire training was performed on a desktop PC with Intel (R) Xeon(R) Platinum 8370C CPU @ 2.80GHz, 80 GB RAM, and an NVIDIA A100 GPU. Training of the MARG algorithm cost approximately twelve hours.

B. Hardware

To validate our approach, we conducted extensive real-world trials using Unitree Go1 and Go2 quadruped robots. Both platforms were equipped with a Livox Mid360 LiDAR operating at 10 Hz, while maintaining a consistent 50 Hz synchronization frequency across the control policy, state estimator, and elevation mapping network. All joint commands were transmitted through an Ethernet interface for reliable communication. The Go1 utilized an onboard Intel NUC for state estimation and control computations, adding nearly 2 kg of payload. We configured the controller with proportional and derivative gains of $\boldsymbol{k}_p = 30.0$ and $\boldsymbol{k}_d = 0.8$, respectively. In contrast, the Go2's computational tasks were handled by an externally Ethernet-connected laptop featuring an Intel i7-12700H CPU, and we used slightly higher control gains with $\boldsymbol{k}_p = 40.0$ and $\boldsymbol{k}_d = 1.2$.

V. RESULTS AND DISCUSSION

A. Algorithm Comparison

Fig. 5 (a) illustrates the learning performance of six different DRL algorithms (including MARG, MorAL [2], Vanilla PPO [48], Concurrent [4], RMA [49], and DreamWaQ [10]) in the risky gap terrain. MARG demonstrates the most robust overall performance due to its capacity for environmental perception, resulting in reward values that significantly exceed those of the other algorithms. MorAL and DreamWaQ exhibit notable performance in the later stages, attributed to their effective exploitation mechanisms, which incorporate terrain maps into the critic network and indirectly guide policy updates. However, these algorithms are prone to converging to local optima without active terrain perception. Due to the inclusion of some privileged information, such as mass or foot contact, the reward values of Concurrent and RMA are moderate but fluctuate greatly and clearly cannot adapt to the gap terrain. In contrast, Vanilla PPO consistently exhibits the lowest reward values, demonstrating slow growth and poor convergence. MARG shows minimal fluctuations in the later stages, indicating the best stability on gap terrain.

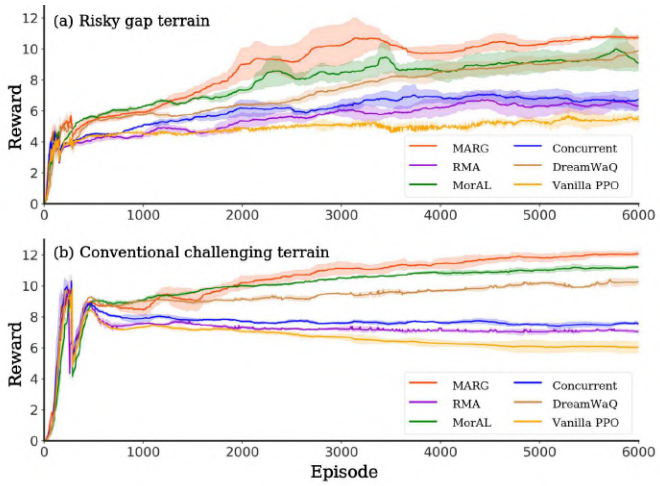


Fig. 5. The average rewards of six controllers on 4096 Go1 over 6000 episodes for two different types of terrain: (a) Risky gap terrain and (b) Conventional challenging terrain. Each curve's shaded region represents the standard deviation of reward values across three different random seeds, indicating the uncertainty in the results.

To further validate the generalizability of MARG, we extend the comparison to a conventional challenging terrain, consistent with the terrain settings employed in MorAL [2] and DreamWaQ [10]. As shown in Figure 5 (b), the performance trend of the six controllers is highly consistent with that in the risky gap terrain. Specifically, MARG still exhibits the best performance, with MorAL and DreamWaQ delivering better average rewards than Concurrent, Vanilla PPO, and RMA. Overall, these experiments demonstrate that MARG has significant advantages, whether for risky or conventional challenging terrain. This is attributed to MARG not only acquiring the explicit estimation but also the terrain map $\boldsymbol{h}_t$ surrounding the robot, which enables the policy to reason about the robot's states.

While several existing learning-based parkour controllers (including Anymal Parkour [35], PIE [50], Extreme Parkour [5], and Parkour [28]) have demonstrated capabilities in gap terrains, our comprehensive evaluation reveals MARG's distinct advantages across multiple performance dimensions, as

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

TABLE III
COMPARISON WITH DIFFERENT QUADRUPEDAL PARKOUR LOCOMOTION CONTROLLERS.

Controllers	**MARG**	**Anymal Parkour**	**PIE**	**Parkour**	**Extreme Parkour**
Agent numbers	4096	4096	4096	256	192
GPU memory	$\approx$ 12 GB	$>$ 45 GB	$>$20 GB	$>$ 15 GB	$>$ 12 GB
Training on single phase	Yes	No	Yes	No	No
Deploy with single policy	Yes	No	Yes	Yes	Yes
Extra sensors during simulation	$\times$	✓	✓	✓	✓
Extra sensors during deployment	1 LiDAR	6 Depth Cameras and 1 LiDAR	1 Depth Camera		

shown in Table III. The TMG model enables MARG to achieve remarkable training efficiency by eliminating the dependency on exteroceptive sensors (e.g., LiDARs or cameras) during simulation training. It reduces GPU memory consumption, allowing efficient large-scale training with 4096 parallel agents with minimal resource requirements. Moreover, the asymmetric actor-critic framework enables both MARG and PIE to achieve single-stage end-to-end training, which facilitates direct sim-to-real transfer without intermediate adaptation phases. This reduces the complexity of the training process and simplifies parameter adjustment and model optimization. Furthermore, unlike Anymal Parkour [35], MARG and others only rely on a single LiDAR or depth camera for environmental perception during deployment. This simplified approach not only significantly reduces system complexity but also remarkably cuts down on hardware costs. Overall, MARG demonstrates superior training efficiency and more practical deployment features among these controllers.

B. Explicit Estimation Comparison

The accurate foot contact state and linear velocity are crucial for the locomotion of legged robots, yet the IMU and joint encoders of these robots cannot directly acquire such key data [51]. Model-based estimators [52], [53], [54] often rely on precise dynamics models for estimation and also fail to handle foot slip. While model-free methods [55], [56], [57] allow robots to search for optimal strategies through continuous trial-and-error learning from data, thus avoiding the challenges of precise modeling. For example, NMN [58] uses supervised learning to train a neural network to estimate contact probability and body linear velocity. However, this estimator is highly dependent on pre-collected labeled data. In complex and unpredictable environments such as risky gaps, data collection is extremely difficult. This leads to poor generalization performance of the model, making it hard to adapt to new or unexpected terrain features. To address these issues, algorithms like MorAL [2] and DreamWaQ [10] train estimators and policies simultaneously on multiple terrains. Based on this, our MARG integrates terrain map information and privileged information during training, enabling more accurate estimation of foot contact states and linear velocities.

We further analyze the accuracy of the Estimator net on estimated linear velocity and foot contact states during Go1 across the continuous gap terrains, as shown in Fig. 6. The top panel illustrates the squared estimation error across a sequence of steps for three algorithms: MARG, NMN, MorAL, and

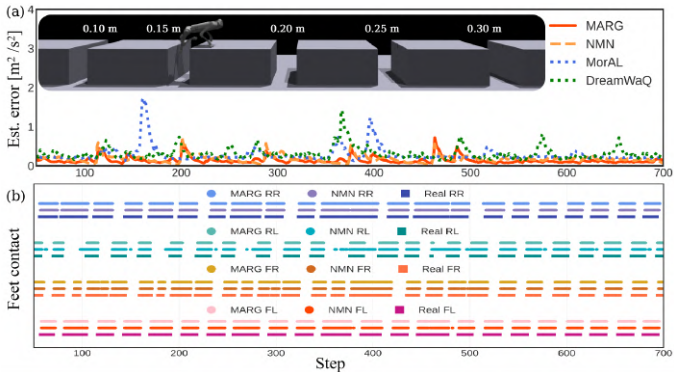


Fig. 6. Accuracy of Estimator Net on various gaps ranging from 0.10 to 0.30 m. (a) The squared velocity estimation error of MARG, NMN, MorAL, and DreamWaQ. (b) Comparison of real and estimated foot contact states.

DreamWaQ. MARG consistently maintains lower estimation errors across varying gap sizes, highlighting its superior environmental perception capabilities. NMN exhibits an estimation error pattern relatively close to MARG, indicating that there are relatively minor discrepancies between them in terms of velocity estimation error. In contrast, MorAL and DreamWaQ exhibit higher peaks in estimation error, particularly during specific gap transitions, indicating challenges in adapting to abrupt environmental changes.

The bottom panel focuses on comparing real and estimated contact states in teams of the front left (FL) and front right (FR) feet, as well as the rear left (RL) and rear right (RR) feet. The estimated contact states of MARG closely match the real states, accurately predicting foot contact across all feet. Meanwhile, each foot displays similar contact patterns, with the contact states maintaining a consistent pattern and slight periodic differences between the front and rear feet. FL and RR have similar contact durations longer than FR and RL, suggesting that the robot uses a trot gait to traverse the gaps. NMN also showcases a relatively good performance in predicting contact states. Although the RL foot shows a slightly noticeable prediction error, this minor deviation does not overshadow NMN's overall capacity to approximate the real contact states.

Overall, MARG proves to be a highly effective algorithm in both linear velocity estimation and contact state prediction. Its superior environmental perception and accurate state estimation contribute to stable and efficient locomotion in risky gap terrains. While NMN shows potential with a performance close to MARG in some aspects, it relies on a learned policy

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

to collect data, which is both time-consuming and labor-intensive, especially for complex and risky tasks. In contrast, our MARG controller can be trained simultaneously with the policy. This concurrent training approach allows MARG to encounter a broader range of scenarios compared to offline data collection methods.

TABLE IV
COMPARISON OF AVERAGE SQUARED ESTIMATED LINEAR VELOCITY ERROR ACROSS DIFFERENT ALGORITHMS IN GAP TERRAINS (3 INDEPENDENT SEEDS, 700 TIME STEPS).

Steps	Algorithms		
	MorAL	MARG	DreamWaQ
100	0.326 ± 0.043	$\mathbf{0.219 \pm 0.021}$	0.330 ± 0.078
300	0.311 ± 0.026	$\mathbf{0.225 \pm 0.029}$	0.294 ± 0.027
500	0.301 ± 0.068	$\mathbf{0.228 \pm 0.031}$	0.308 ± 0.017
700	0.270 ± 0.074	$\mathbf{0.229 \pm 0.029}$	0.304 ± 0.012

For a fair comparison, we also compare average squared linear velocity errors during gap traversal, as shown in Table IV. Notably, the MARG algorithm consistently demonstrates the smallest absolute error and the lowest standard deviation across all tested steps. This performance highlights that MARG's estimator net achieves exceptional precision and robustness in gap terrain scenarios.

C. Ablation Studies of Observation Space

Fig. 7 provides a comprehensive analysis of the performance of six different observation spaces on a risky terrain. We categorize observation spaces into six distinct types, each incorporating different sensor inputs, and evaluate these policies through four metrics: average reward, success rate, average velocity, and travel distance.

Fig. 7 (a) illustrates the reward distribution after the convergence of all controllers, specifically between episodes 4000 and 6000. The reward values decrease progressively from observation spaces ① to ⑥, with ① and ② achieving the highest rewards and minimal variability.

The robot is commanded to traverse the discontinuous terrains (including gaps, stones, and balance beams) at a speed of 0.9 m/s to further evaluate the different observation space combinations. After running the trials 50 times, we calculate the success rate, average velocity, and average travel distance, as shown in Figs. 7 (b)-(d). Notably, ①, ②, and ③ exhibit higher success rates, average velocity, and long travel distance, suggesting that these combinations are effective for this risky task. The nearly zero success rates and significantly shorter travel distances observed for cases for ④, ⑤, and ⑥ demonstrated that without the body velocity or terrain map, completing the task is impossible. Due to the lack of body velocity, the average speed of ⑤ differs significantly from the command, resulting in stagnation because the robot perceives the gap terrain as a dangerous area. In addition, when the terrain map is lacking, ⑥ exhibits a high average velocity, often resulting in direct collisions with the terrain or falling into gaps.

Overall, we can conclude that different combinations of observation spaces significantly impact the performance of the learned policy, with both privileged information and exteroception terrain contributing to the performance. The terrain map is the most crucial factor, followed by the body velocity and the contact state. Additionally, using Elevation Net to extract elevation features $\boldsymbol{e}_t^h$ proves more efficient and stable than inputting the relative height map $\hat{\boldsymbol{h}}_t$.

D. Ablation Studies of Footholds Rewards

We also carry out detailed ablation experiments and comprehensively analyze the performance of five different MARG controllers with various foothold reward configurations in risky terrains. These controllers are classified according to five distinct footholds rewards setups: (1) MARG with all Footholds rewards; (2) MARG with Feet stumble reward; (3) MARG with Feet air time reward; (4) MARG with Feet center reward; (5) MARG without Footholds rewards. Fig. 8 shows the three evaluation metrics: (a) average foot contact loss, (b) success rate, and (c) travel distance.

Fig. 8 (a) represents the variation of the average foot contact loss, measured as $MSE(\hat{\boldsymbol{c}}_t, \boldsymbol{c}_t)$, of the five different MARG controllers during 6000 training episodes on risky terrains. Each curve corresponds to a distinct foothold reward configuration for the MARG controllers. As the number of training episodes increases, all curves exhibit a downward trend, signifying a gradual enhancement in the model's performance. Among them, (1), which is equipped with all foothold rewards, showcases optimal performance. Its loss value consistently stays at the lowest level and drops most swiftly. Conversely, (5), lacking foothold rewards, demonstrates the poorest performance, characterized by the highest loss value and a notably slower rate of decline. The performances of the remaining controllers fall between these two extremes. In conclusion, controllers with all foothold rewards can significantly enhance the accuracy of the Estimate network. Although single-item rewards (2) to (4) can improve prediction accuracy to some extent, none of them can match the performance of controllers equipped with all foothold rewards. Figs. 8 (b) and (c) present the success rates and travel distance of five different MARG controllers when traversing through six distinct risky gap terrains. Each controller is evaluated via 20 repeated traversal tests on level-6 risky terrains. (1) shows significantly superior performance, outperforming the other controllers. In contrast, (5) exhibits the lowest success rate and travel distance. The success rates and traversal distances of (2) to (4) are intermediate, with values falling between those of (1) and (5). Among them, (2) shows the most outstanding performance, followed by (3), and (4).

Overall, the ablation experiments demonstrate the crucial role of foothold rewards in the performance of MARG. The controller, equipped with all foothold rewards, demonstrates a superior performance, outshining its counterparts across various evaluation metrics. In contrast, controllers lacking foothold rewards demonstrate significantly degraded performance, highlighting the indispensable role of these rewards in traversing risky gap terrains.

Different reward functions have varying degrees of influence on the performance of the controller. Among them, the foot

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

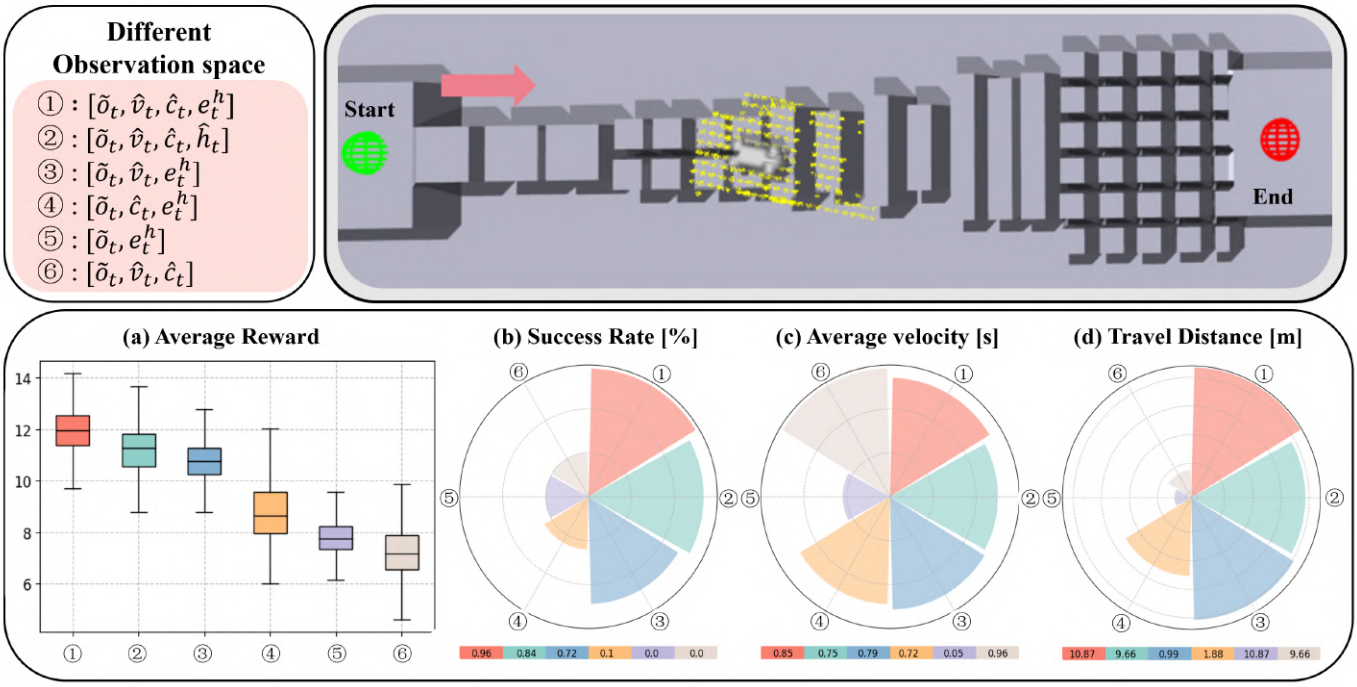


Fig. 7. Evaluating the impact of 6 different observation spaces on the performance of the learned policy in risky terrains. The evaluation is based on four metrics: (a) average reward, (b) success rate, (c) average velocity, and (d) travel distance. Different combinations of sensor inputs in these observation spaces show different effects on the robot's performance in traversing discontinuous terrains.

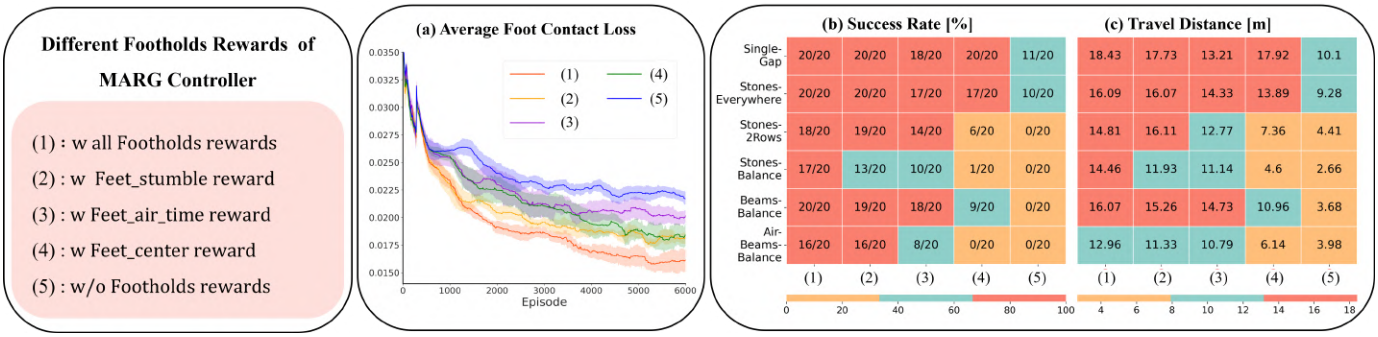


Fig. 8. Evaluating the impact of different foothold rewards on the performance of the MARG in risky terrains. (a) represents the average foot contact loss of 5 different MARG controllers during training 4096 Unitree Go1 robots for 6000 episodes. Each curve's shaded region represents the standard deviation of loss values across three different random seeds, indicating the uncertainty in the results. (b-c) represent the success rate and average traversal distance of five different MARG controllers when traversing six risky terrains. Each controller is evaluated through 20 repeat traversals of the level-6 risky terrains.

stumble reward has proven to be the most effective, followed by the foot air time reward, with the foot center reward having the least impact. The feet stumble reward plays a crucial role, as it punishes the robot for tripping and encourages the controller to maintain a more stable gait, which is essential for successfully crossing risky terrain. The feet air time reward may contribute to optimizing the timing and coordination of leg movements, thereby ensuring efficient and safe traversal of uneven gap terrains. Although the impact of the foot center reward is comparatively modest, it can still play a role in preventing the robot from slipping off by influencing the foothold of the quadruped. By guiding the foothold of the feet, this reward mechanism helps avoid precarious positioning on the edges of gaps, thereby reducing the likelihood of missteps and subsequent falls.

E. Dynamics analysis over risky terrains

Fig. 9 provides a comprehensive analysis of a quadruped robot's locomotion dynamics during the deployment of its learned policy in the Gazebo simulation. Fig. 9 (a) offers a side view of the robot's trajectory and gait sequence, with colored lines representing the paths of individual legs and body, highlighting the spatial adjustments for balance and progression. (b) comprises three subplots: the first displays the contact sequence of each leg with the ground and the specific gait (i.e. trot), both of which are crucial for stability and propulsion; the second shows the angular positions of the robot's joints over time, indicating adaptive responses; and the third depicts the torque applied to each joint, reflecting active adjustments for balance and control. Our controller demonstrates the intricate coordination and control mechanisms required for the robot to

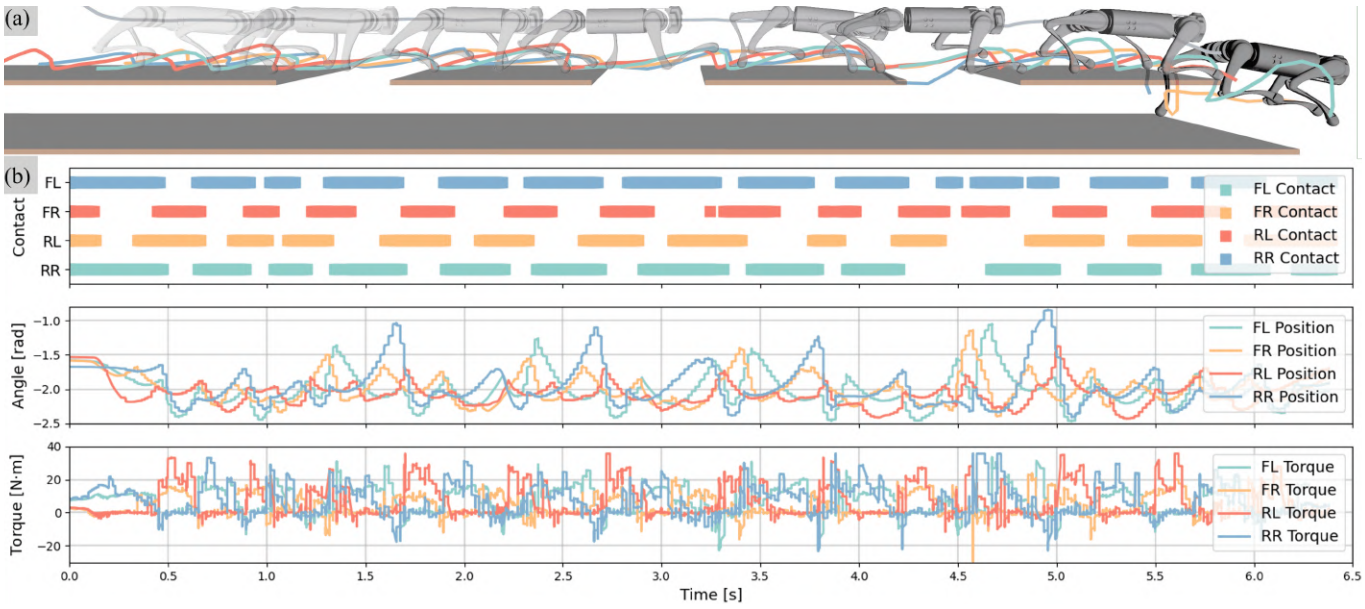


Fig. 9. Dynamics analysis of quadruped robot traversing discontinuous gaps in gazebo simulation. (a) Snapshots of the simulated consecutive traversals across 40 cm gaps with body and foot positions in the world frame. (b) Variation curves of contact, joint angles, and torques during gap crossings.

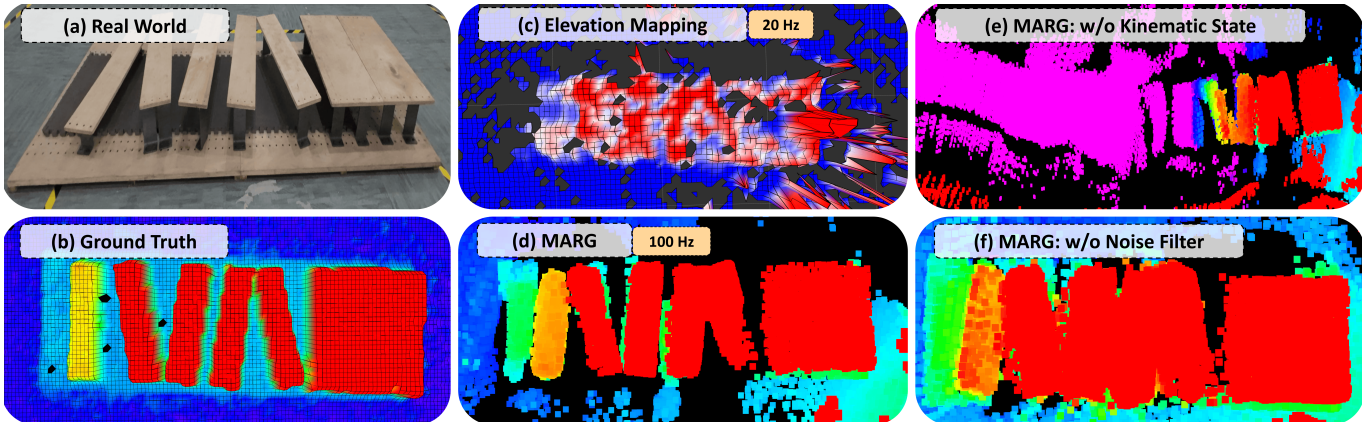


Fig. 10. Comparison of two mapping methods with the ground truth. (a) shows the real-world snapshots, (b) is the ground truth for the mapping accuracy benchmark. The ground-truth terrain map is generated through handheld LiDAR scanning (Mid360) with FAST-LIO2 SLAM reconstruction (0.01 m resolution) and CloudCompare post-processing. (c) is an elevation mapping method with a 20 Hz update rate, showing less detail and more noise. (d) is the MARG method with a 100 Hz update rate, providing a clearer and more detailed representation. (e) shows the MARG method without kinematic state, and (f) shows the MARG method without noise filter, which highlights the importance of these components in mapping.

achieve efficient and stable locomotion across risky gaps.

TABLE V
ACCURACY EVALUATION ON FOUR DIFFERENT DATASETS

Method	Park (m)		Running (m)		Indoor (m)		Corridor (m)	
	APE	RPE	APE	RPE	APE	RPE	APE	RPE
EKF	4.44	0.21	0.64	0.19	0.15	0.10	2.10	0.17
FAST-LIO2	0.30	0.15	0.30	0.18	0.19	0.25	0.49	0.19
LIO-SAM	0.21	0.15	0.10	0.17	1.24	0.71	1.04[1]	0.23[1]
A-LOAM	2.86	0.19	0.09	0.15	0.06	0.08	0.51	**0.14**
MARG (Ours)	**0.19**	**0.15**	**0.04**	**0.04**	**0.05**	**0.06**	**0.40**	0.21

[1] This method fails, and we cut off the drift part.

F. Analysis of the TMG model

Table. V presents an accuracy evaluation of different localization methods, including MARG, EKF [36], LIO-SAM [37], A-LOAM [38], FAST-LIO2 [39] and Ground Truth, based on Absolute Pose Error (APE) and Relative Pose Error (RPE) across a dataset [59] comprising Park, Running, Indoor, and Corridor scenarios. MARG consistently achieves the lowest APE and RPE across most environments, highlighting its precision and reliability.

Meanwhile, we also compare the performance of these five localization algorithms in long-distance trajectory drift in these datasets, as shown in Fig. 11 (a). MARG and FAST-LIO2 closely follow this path, indicating high accuracy, while EKF shows significant deviation, particularly in turns. LIO-SAM deviates after the path turns, and A-LOAM shows some inconsistencies. For the 3D elevation tracking shown in Fig. 11 (b), the ground truth line remains stable, with MARG aligning closely, demonstrating robust performance. In contrast, LIO-SAM exhibits significant fluctuations, indicating poor eleva-

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

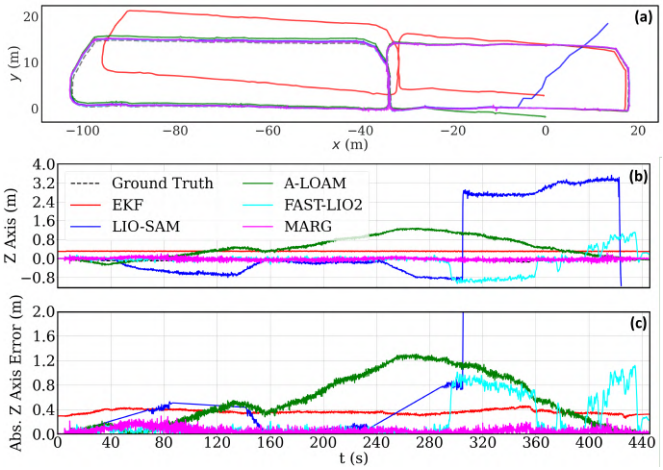


Fig. 11. Compare the five localization algorithms on the corridor dataset in terms of the long-distance trajectory (a), z-axis position (b), and absolute z-axis error (c).

tion accuracy. A-LOAM and FAST-LIO2 have deviations, as can be seen in Fig. 11 (c). These deviations cause errors in the elevation map, which can harm quadrupedal locomotion. The fact that only our MARG maintains an accurate height estimation highlights the effectiveness of our proposed method. Overall, MARG excels in both path and elevation tracking, highlighting the importance of choosing a reliable localization algorithm in complex environments.

G. Real-world transfer on indoor risky gap terrains

Fig. 10 compares two mapping methods for accurately representing a real-world structure. The top quadrants (a-b) depict the actual physical setup and the ground truth, serving as the benchmark for mapping accuracy. The elevation mapping method [60] operates at a slower update rate of 20 Hz, as illustrated in Fig. 10 (c). As a result, it captures less detail and exhibits more noise, highlighting its limitations in accurately rendering fine structural features. In contrast, the proposed MARG method (d), updating at 100 Hz, provides a significantly clearer and more detailed representation that closely aligns with the ground truth. In the absence of the kinematic state, as shown in (e), the elevation map exhibits significant deviations due to the IMU becoming oversaturated when the robot experiences impacts during movement. Compared to (f), it is evident that the noise filter effectively eliminates uncertainties caused by the beam angle and manages the beam divergence characteristics of the LiDAR sensor. These comparisons underscore MARG's superior capability in achieving high-resolution and accurate mapping at higher frequencies, thus demonstrating its effectiveness over lower-frequency methods.

Extensive tests have been carried out on various risky terrains, as shown in Fig. 12 and the supplementary video.

1) Large gaps: Our controller can cross large gaps of 65 cm (1.6 $\times$ robot length) with a trot gait, as shown in Fig. 12 (a). The MARG guides the robot to adjust the leg extension angle and the magnitude of force to maintain a stable posture and a safe landing point when crossing the gap.

2) Single-plank bridge: In addition, the controller can cross a 20 cm single-plank bridge scene (0.7 $\times$ robot width), demonstrating the robot's balance ability on narrow surfaces, as shown in Fig. 12 (b). The MARG controller can quickly respond to any tiny shaking or imbalance and dynamically adjust the support position and strength of the legs to ensure that the robot always maintains a stable center of gravity and avoids slipping or overturning.

3) Beams: Furthermore, we have conducted experiments on balance beams with different heights, widths, and inclined angles. The gaps between beams constantly change, as shown in Fig. 12 (c-f). These complex and variable experiments pose a severe challenge to the locomotion controller. For example, beams of different heights require the robot to precisely adjust the extension degree of its legs to adapt to different ascending and descending slopes. Various widths of breams and the constantly changing gap further increase the difficulty of the robot locomotion. The controller needs to accurately plan the footholds and leg movements at each step to ensure the robot can pass through these risky terrains stably and safely. Beams with different inclined angles (approximately 10 to 15 degrees) also demonstrate the controller's ability to perceive terrain and adjust the robot. These experiments illustrate the effectiveness and adaptability of the MARG controller in practical applications and highlight its significant advantages in solving the problem of quadruped robots walking on dangerous terrains.

4) Mixed risky terrain: Fig. 13 illustrates the quadruped robot locomotion across a series of uneven gaps, demonstrating its ability to maintain balance and stability. The top sequence shows the robot's progression over time, while the bottom graphs depict its position and velocity dynamics. The data highlights the robot's effective control mechanisms, enabling it to adapt its movement and maintain consistent velocity, even when encountering complex terrain features.

H. Real-world transfer on outdoor terrains

Fig. 14 (a) shows the comprehensive outdoor experimental scenarios conducted on the campus of the University of Hong Kong. We conduct systematic tests on Unitree Go2 to comprehensively evaluate the performance of MARG on diverse outdoor risky terrains, including 9 cm beams, 18 cm single-plank bridges, and various sizes of gaps. Beyond these risky terrains, the robot is also commanded to traverse various other conventional terrains, such as gardens, slopes, gaps, and stairs. Additionally, the environment tests are conducted on a wide range of lighting conditions, from bright daylight to dark. The results clearly indicate that the TMG model empowers robots to execute tasks efficiently across diverse campus outdoor environments, thereby underscoring its remarkable adaptability and stability.

Fig. 14 (b) highlights the robustness test of the hiking route in Lung Fu Shan Country Park. The yellow dashed line marks the experimental route, and seven key terrain nodes from A to G are selected. These terrains include various complex obstacles such as steps, platforms, and guardrails, comprehensively simulating the typical challenging scenarios in the natural outdoor environment. By presenting the real-scene photos and the submitted video, the entire movement

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

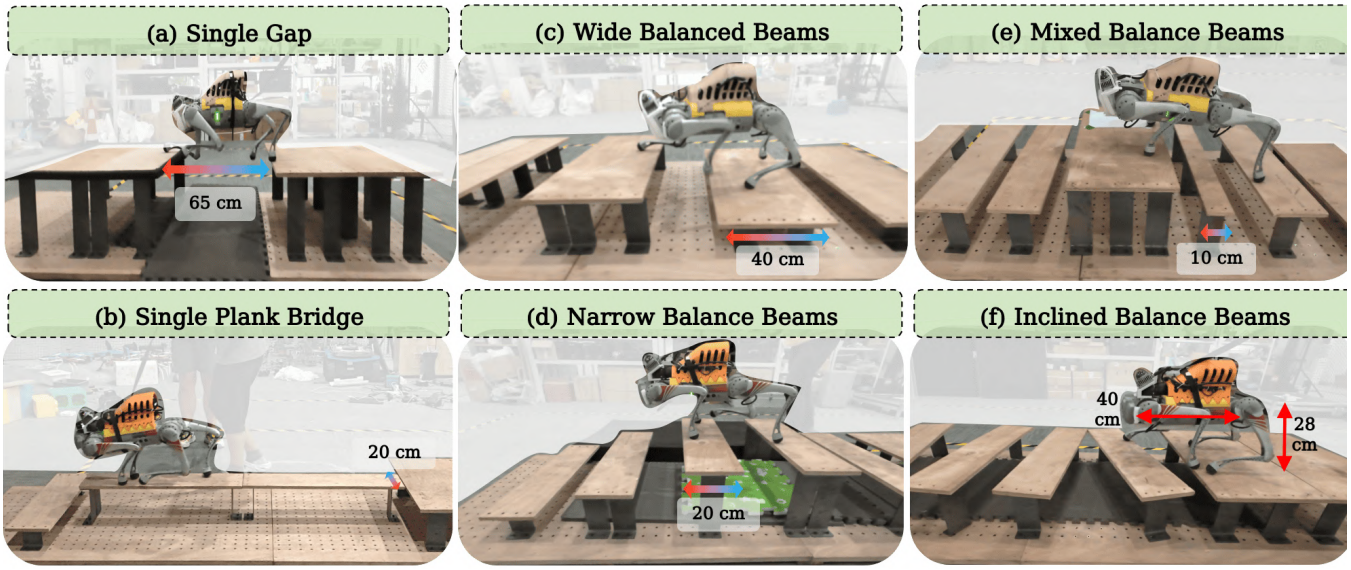


Fig. 12. Deployment scenarios on the quadrupedal robot Go1 under various risky gaps: (a) traversing a 65 cm wide gap using a trot gait. (b) walking on a 20 cm single-plank bridge to demonstrate balance ability. (c-f) crossing balance beams with varying heights, widths, and inclination angles.

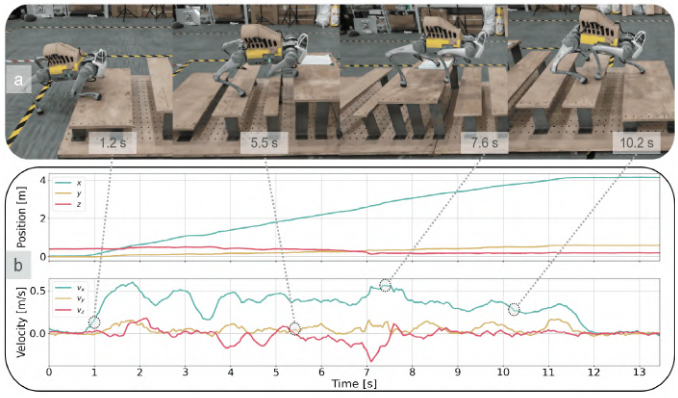


Fig. 13. Learned motions on balance beams in real-world experiments. (a) presents the motion snapshots of the robot during traversal in real-world scenarios, offering a visual record of its movement process. (b) exhibits the position and velocity of the robot, which shows the kinematic characteristics during the motion.

process of the robot outdoors is intuitively demonstrated. Moreover, the generated terrain map clearly demonstrates its dynamic changes during the robot's movement, enabling an intuitive grasp of how the robot senses and adapts to diverse terrain elevations and contours. The experimental results show that the TMG model can operate stably and effectively deal with various challenges of the outdoor terrain.

I. Limitations

1) Dependency on sensor accuracy: The performance of our system depends critically on the accuracy of the point cloud data captured by the LiDAR sensor. Although we have proposed a method to enhance point cloud fidelity, factors such as sensor noise, dust accumulation, calibration errors and the vertical field of view still influence the quality of the elevation maps which will shift or distort the point cloud, impair obstacle detection, and pose significant safety risks in high-precision robotic applications.

2) Limited Gait Diversity: MARG can only achieve the trot gait. Previous studies [61], [62] have shown that quadruped animals can adapt to different terrains through various gaits. In contrast, our current approach has not fully explored the locomotion potential of the robot.

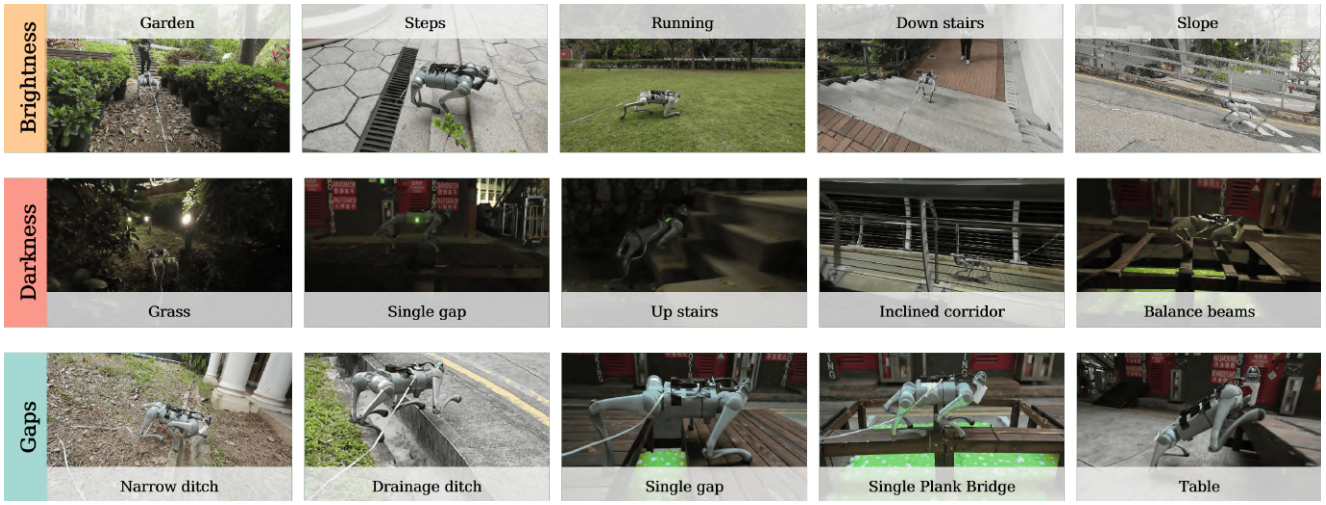
VI. CONCLUSIONS

This study successfully demonstrated the excellent locomotion performance of the MARG controller for quadruped robots. The MARG controller accurately predicted the body velocity and the contact state of each foot, ensuring that the robot adjusted its posture timely manner under imbalances. Meanwhile, the three foot-related rewards proved to be extremely effective in guiding the robot to explore safe footholds. In addition, the TMG model relied solely on a single LiDAR to generate accurate terrain maps, simplifying the hardware deployment. Thus, the policy trained in the simulation could be directly transferred to the real world, significantly enhancing the adaptability and practicality of the robots. The experiments demonstrated that the MARG controller was stable and effective across risky tasks, successfully balancing safety, stability, and efficiency during locomotion. In the future, we will further explore ways to optimize the controller's performance and expand its application to a broader range of scenarios, including soft, slippery, and unstable risky terrains.

REFERENCES

[1] S. Fahmi, V. Barasuol, D. Esteban, O. Villarreal, and C. Semini, "Vital: Vision-based terrain-aware locomotion for legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 885–904, 2023.

[2] 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.



(a) These outdoor experiments are conducted on diverse terrains, including gardens, slopes, gaps, stairs, and so on. The tests are performed under varying illumination conditions, ranging from bright to dark environments. These experiments systematically evaluate the performance of MARG in different real-world terrains.

(b) The experimental route covers seven key terrain nodes from A to G, including various obstacles such as steps, platforms, and guardrails. Meanwhile, through real-scene photos and photos of elevation points restored by TMG, the motion control performance is verified from multiple dimensions, comprehensively evaluating the stability and reliability of the controller in complex outdoor terrains.

Fig. 14. Robustness testing of the Unitree Go2 robot in diverse challenging terrains: (a) Outdoor experiments are conducted on the HKU campus, (b) Robustness tests are carried out on the hiking route in Lung Fu Shan Country Park.

[3] A. Abdalla, M. Focchi, R. Orsolino, and C. Semini, "An efficient paradigm for feasibility guarantees in legged locomotion," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3499–3515, 2023.

[4] G. Ji, J. Mun, H. Kim, and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4630–4637, 2022.

[5] 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.

[6] E. Xiao, Y. Dong, J. Lam, and P. Lu, "Learning stable bipedal locomotion skills for quadrupedal robots on challenging terrains with automatic fall recovery," *npj Robotics*, vol. 3, no. 22, pp. 1–13, 2025.

[7] D. Kim, H. Kwon, J. Kim, G. Lee, and S. Oh, "Stage-wise reward shaping for acrobatic robots: A constrained multi-objective reinforcement learning approach," *arXiv preprint arXiv:2409.15755*, 2024.

[8] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[9] 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.

[10] 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.

[11] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning*. PMLR, 2023, pp. 22–31.

[12] Z. Luo, E. Xiao, and P. Lu, "Ft-net: Learning failure recovery and

fault-tolerant locomotion for quadruped robots," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8414–8421, 2023.

[13] 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.

[14] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "Dtc: Deep tracking control," *Science Robotics*, vol. 9, no. 86, p. eadh5401, 2024.

[15] 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.

[16] S. Singh, R. P. Russell, and P. M. Wensing, "Analytical second-order derivatives of rigid-body contact dynamics: Application to multi-shooting ddp," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–8.

[17] H. Li and P. M. Wensing, "Cafe-mpc: A cascaded-fidelity model predictive control framework with tuning-free whole-body control," *IEEE Transactions on Robotics*, 2024.

[18] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti, "Biconmp: A nonlinear model predictive control framework for whole body motion planning," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 905–922, 2023.

[19] P. Holmes, R. J. Full, D. Koditschek, and J. Guckenheimer, "The dynamics of legged locomotion: Models, analyses, and challenges," *SIAM review*, vol. 48, no. 2, pp. 207–304, 2006.

[20] 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.

[21] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti, "Biconmp: A nonlinear model predictive control framework for whole body motion planning," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 905–922, 2023.

[22] Y. Yin, Y. Zhao, Y. Xiao, and F. Gao, "Footholds optimization for legged robots walking on complex terrain," *Frontiers of Mechanical Engineering*, vol. 18, no. 2, p. 26, 2023.

[23] M. Sombolestan and Q. Nguyen, "Adaptive force-based control of dynamic legged locomotion over uneven terrain," *IEEE Transactions on Robotics*, 2024.

[24] S. Fahmi, V. Barasuol, D. Esteban, O. Villarreal, and C. Semini, "Vital: Vision-based terrain-aware locomotion for legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 885–904, 2022.

[25] C. D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Günther, M. Tranzatto, P. Fankhauser, and M. Hutter, "Advances in real-world applications for legged robots," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1311–1326, 2018.

[26] M. Bjelonic, R. Grandia, O. Harley, C. Galliard, S. Zimmermann, and M. Hutter, "Whole-body mpc and online gait sequence generation for wheeled-legged robots," in *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2021.

[27] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[28] 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.

[29] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," in *8th Annual Conference on Robot Learning*.

[30] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (gru) neural networks," in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*. IEEE, 2017, pp. 1597–1600.

[31] A. Graves and A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

[32] X. Dong, M. A. Garratt, S. G. Anavatti, and H. A. Abbass, "Towards real-time monocular depth estimation for robotics: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16 940–16 961, 2022.

[33] 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.

[34] H. Shi, Q. Zhu, L. Han, W. Chi, T. Li, and M. Q.-H. Meng, "Terrain-aware quadrupedal locomotion via reinforcement learning," *arXiv preprint arXiv:2310.04675*, 2023.

[35] 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.

[36] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.

[37] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.

[38] J. Zhang, S. Singh *et al.*, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[39] 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.

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

[41] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu based physics simulation for robot learning," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[42] Z. Wang, Y. Li, L. Xu, H. Shi, Z. Ma, Z. Chu, C. Li, F. Gao, K. Yang, and K. Wang, "Sf-tim: A simple framework for enhancing quadrupedal robot jumping agility by combining terrain imagination and measurement," *arXiv e-prints*, pp. arXiv–2408, 2024.

[43] P. Fankhauser and M. Hutter, "A universal grid map library: Implementation and use case for rough terrain navigation," *Robot Operating System (ROS) The Complete Reference (Volume 1)*, pp. 99–120, 2016.

[44] 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, Oct. 2024, p. 8119–8125.

[45] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4423–4430.

[46] X. Wang, Y. Chen, and W. Zhu, "A survey on curriculum learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4555–4576, 2021.

[47] 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.

[48] L. Smith, I. Kostrikov, and S. Levine, "Demonstrating a walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning," *Robotics: Science and Systems (RSS) Demo*, vol. 2, no. 3, p. 4, 2023.

[49] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems XVII*, ser. RSS2021. Robotics: Science and Systems Foundation, Jul. 2021.

[50] S. Luo, S. Li, R. Yu, Z. Wang, J. Wu, and Q. Zhu, "Pie: Parkour with implicit-explicit learning framework for legged robots," *IEEE Robotics and Automation Letters*, 2024.

[51] Z. Yoon, J.-H. Kim, and H.-W. Park, "Invariant smoother for legged robot state estimation with dynamic contact event information," *IEEE Transactions on Robotics*, vol. 40, pp. 193–212, 2023.

[52] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.

[53] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State estimation for legged robots on unstable and slippery terrain," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 6058–6064.

[54] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 402–430, 2020.

[55] T.-Y. Lin, R. Zhang, J. Yu, and M. Ghaffari, "Legged robot state estimation using invariant kalman filtering and learned contact events," in *Conference on Robot Learning*. PMLR, 2022, pp. 1057–1066.

[56] R. Buchanan, M. Camurri, F. Dellaert, and M. Fallon, "Learning inertial odometry for dynamic legged robot state estimation," in *Conference on robot learning*. PMLR, 2022, pp. 1575–1584.

[57] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, "Tlio: Tight learned inertial odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5653–5660, 2020.

[58] D. Youm, H. Oh, S. Choi, H. Kim, and J. Hwangbo, "Legged robot state estimation with invariant extended kalman filter using neural measurement network," *arXiv preprint arXiv:2402.00366*, 2024.

[59] G. Ou, D. Li, and H. Li, "Leg-kilo: Robust kinematic-inertial-lidar odometry for dynamic legged robots," *IEEE Robotics and Automation Letters*, 2024.
[60] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3019–3026, 2018.
[61] R. M. Alexander, *Principles of animal locomotion*. Princeton university press, 2003.
[62] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning*. PMLR, 2023, pp. 22–31.

Yinzhao Dong received the B.Sc. degree from Jilin University, Changchun, China, and the M.Sc. degree from Dalian University of Technology, Dalian, China, in 2018 and 2021, respectively. He is currently working toward the Ph.D. degree in mechanical engineering with the Adaptive Robotic Controls Lab(ArcLab), the University of Hong Kong, Hong Kong.

His research interests include legged robotics, motion planning, and reinforcement learning.

Ji Ma received the B.Sc. degree in Information Engineering from Jilin University, Changchun, China, in 2023. He is currently working toward the Ph.D. degree in mechanical engineering with the Adaptive Robotic Controls Lab(ArcLab), the University of Hong Kong, Hong Kong.

His research interests include legged robotics, motion planning, and reinforcement learning.

Liu Zhao received the B.Sc. degree in Automation and M.Sc. degree in Control Science and Engineering both from Harbin Institute of Technology (HIT), in 2021 and 2023. She is currently working toward the Ph.D. degree in mechanical engineering with the Adaptive Robotic Controls Lab(ArcLab), the University of Hong Kong, Hong Kong.

Her research interests include vision, navigation, and control of legged robotics.

Wanyue Li (Graduate Student Member, IEEE) received the B.Sc. degree in Computer Science and Technology from South China Agricultural University in 2019, and the M.Sc. degree in Artificial Intelligence from Sun Yat-sen University in 2023. He is now a Ph.D. candidate in mechanical engineering with the Adaptive Robotic Controls Lab (ArcLab), the University of Hong Kong, Hong Kong.

His research interests include humanoid robot locomotion control, trajectory optimization, and reinforcement learning.

Peng Lu obtained his BSc degree in automatic control and MSc degree in nonlinear flight control both from Northwestern Polytechnical University (NPU). He continued his journey on flight control at Delft University of Technology (TU Delft) where he received his PhD degree in 2016. After that, he shifted a bit from flight control and started to explore control for ground/construction robotics at ETH Zurich (ADRL lab) as a Postdoc researcher in 2016. He also had a short but nice journey at University of Zurich & ETH Zurich (RPG group) where he was working on vision-based control for UAVs as a Postdoc researcher. He was an assistant professor in autonomous UAVs and robotics at Hong Kong Polytechnic University prior to joining the University of Hong Kong in 2020.

Prof. Lu has received several awards such as 3rd place in 2019 IROS autonomous drone racing competition and best graduate student paper finalist in AIAA GNC. He serves as an associate editor for IROS and session chair/co-chair for conferences like IROS and AIAA GNC for several times. He also gave a number of invited/keynote speeches at multiple conferences, universities and research institutes.