IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 February 2024, accepted 25 February 2024, date of publication 29 February 2024, date of current version 6 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3371579

RESEARCH ARTICLE

Robust Traversability Prediction Using Multiple Costs for Quadruped Robot in Random Terrains

FIKIH MUHAMAD[1], JUNG-SU KIM[1], (Member, IEEE), AND JAE-HAN PARK[2], (Member, IEEE)
[1]Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
[2]AI Robot Research and Development Department, Korea Institute of Industrial Technology (KITECH), Ansan 15588, South Korea

Corresponding author: Jung-Su Kim (jungsu@seoultech.ac.kr)

This work was supported by Korea Research Institute for defense Technology Planning and Advancement (KRIT) funded by Korean Government through the Defense Acquisition Program Administration (DAPA) for Ground Perception Technologies for Quadruped Robots, in 2023, under Grant 20-107-C00-007-03.

ABSTRACT The quadruped robot has to assess the feasibility of upcoming terrains before making contact to safely traverse various environments. This assessment is called traversability in the literature on quadruped robots. Trasversability has recently posed challenges due to a high-dimensional system that leads to long computational times. Furthermore, exteroceptive observations often suffer from noise that potentially causes misinterpretations of terrains and results in an inaccurate assessment. This paper proposes a robust traversability predictor to tackle these issues by utilizing a Convolutional Neural Network (CNN) encoder, CNN decoder, and Multi-layer Perceptron (MLP) to predict multiple costs associated with traversability. The integration of the CNN encoder and decoder helps mitigate the effect of noise in exteroceptive observations, while the MLP network serves as a predictor for multiple costs. The proposed method utilizes the information collected from a physics simulator to avoid hand-crafted multiple-cost labeling. It can predict a comprehensive set of costs that overcomes the limitations of relying on a single cost metric. It also achieves faster computational time by utilizing neural networks, in contrast to the model-based approach in the literature. The robustness of the proposed method is validated by comparing it to a baseline noise-free prediction model and an existing method in the literature. The results indicate that the proposed method exhibits the lowest prediction errors. Therefore, despite the noise in exteroceptive observations, the proposed multiple cost-based traversability predictor has better accuracy and robustness than the baseline and existing methods.

INDEX TERMS Machine learning, quadruped robot, reinforcement learning.

I. INTRODUCTION

A. BACKGROUND

Quadruped robots have gained popularity for their capability to navigate various terrains without falling. However, when the robot encounters unknown environments, this capability declines due to limitations both in the locomotion controller and the kinematics of the robot itself. To ensure the robot successfully traverses such challenging scenarios, it becomes crucial to equip the robot with the ability to assess its surroundings and internal state. This assessment provides essential information to determine the most efficient path to traverse. It guarantees the robot grasps the potential obstacles and upcoming terrain conditions during traversal. It is also crucial for enabling the robot to evaluate the feasibility of reaching its desired position before making contact. This assessment is known as ''traversability'' in the literature on quadruped robots.

In nature, humans and animals possess the innate ability to determine safe and efficient routes during traversal. They can use their foresight to perceive the current state and the upcoming terrain to understand its surroundings. This

The associate editor coordinating the review of this manuscript and approving it for publication was Okyay Kaynak.

ability allows them to assess their surroundings, process the information from their sensors, and calculate the optimal path to ensure their safety and efficiency during traversal. In contrast, the quadruped robot relies on multiple sensors, including an Inertial Measurement Unit (IMU), to monitor its internal state and Lidar as exteroceptive observations to perceive the terrain. These sensors serve as counterparts to the sensory systems found in humans and animals. Therefore, the robot is expected to be able to assess its surroundings by utilizing these sensors and determine the safe path to traverse.

Recently, quadruped robots have faced challenges in determining the feasibility of transitioning from their current state to the desired state when provided with information concerning their current state and external observations. This issue arises from the computationally intensive nature of the calculation itself. That is because the calculation involves a high-dimensional state to assess its feasibility transition accurately. Furthermore, another problem arises when the robot traverses in environments where potential noise in exteroceptive observations frequently occurs, especially when navigating reflective and deformable terrains. These conditions distract the robot's ability to perceive the terrain appropriately, resulting in insufficient performance. Additionally, it is advisable to avoid relying solely on human intuition-based assessments when operators estimate the traversability based on their intuition. It is because of the uncertainties inherent in robot locomotion that cannot be calculated explicitly. These uncertainties can indirectly lead to suboptimal assessments.

B. PREVIOUS WORKS AND MOTIVATIONS

Previous studies in the literature have employed pre-scanned terrain maps, as explained in [1] and [2]. These maps have served as the foundation for calculating cost-affecting traversability by leveraging human intuition. Subsequently, this cost has been employed to optimize the robot's trajectory in a step-by-step manner. Hereafter, other approaches involve labeling traversability based on expert driver paths [3], [4]. This path is assumed to be a highly traversable path. Additionally, another method involves a geometric analysis of the failure costs of quadruped traversability [5]. In pursuing an optimal trajectory through complex motion planning, these hand-crafted labeling approaches presuppose prior knowledge of the complete terrain layout and the robot's states. Nevertheless, this methodology relied on human intuition to assign cost values in the intricate geometry of the terrain, resulting in imprecise assessments.

The model-based approach outlined in [6], focuses on planning footstep placements for walking robots. It considers obstacle awareness and non-convex constraints. However, this method relies heavily on the availability of several convex formulations and obstacle-free regions, potentially leading to local optimal trajectories. In contrast, Winkler et al., as described in [7], utilize trajectory optimization to determine gait sequences and foothold positions for navigating on non-flat terrains. They employ a nonlinear programming

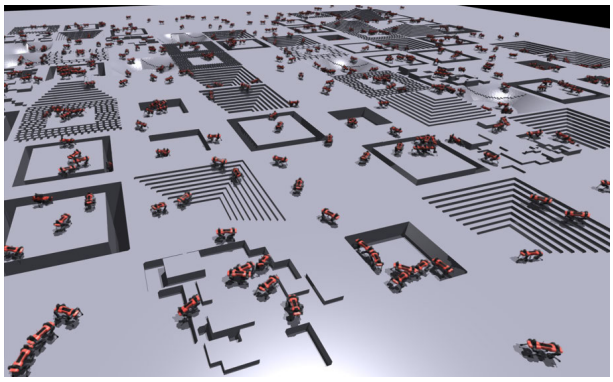


FIGURE 1. Multiple Anymal robots training in randomly generated terrain using Isaac Gym simulation.

solver for this purpose. Nonetheless, this approach simplifies the robot's model and limits its ability to fully exploit the potential of high-dimensional legged robots. An alternative approach presented in [8] utilizes violation constraints to assess the feasibility of transitioning to the next state for legged robots. This approach assumes legged robot motion as a nonlinear optimization problem. However, it suffers from significant computational challenges when transitioning between the current and next states. Several approaches use the Gaussian Process (GP) to predict the likelihood of falling cost in rover operations [9], [10]. However, it suffers from expensive computations during operation.

To address challenges related to high computational demands and simplify calculations, deep learning approaches have been employed to predict traversability costs, as discussed in [11] and [12]. Sevastopoulos et al. [13] compared the performance between model-based and deep-learning methods for the traversability of mobile robots. They explain that deep learning significantly improves traversability estimation. Deep learning overcomes challenges faced by model-based approaches, particularly in handling environmental complexity. Various studies have demonstrated the effectiveness of deep learning approaches employing CNN to predict traversability costs for mobile robots [14], [15], [16], [17], [18], [19]. These approaches are effective for mobile robot applications. However, these approaches have limitations in accurately predicting the feasibility of movement for legged robots due to their distinct and intricate body movements. In the work of Yang et al., [20], traversability is calculated for a given terrain in conjunction with the goal position. However, this approach overlooks the robot's internal state, which can significantly impact costs during traversal on the same terrain. CNN has proven effective in handling large image datasets, as outlined in [21], with the potential for continuous and real-time height map data feeding in robot perception.

Gan et al. [22] propose that incorporating inertial features can enhance model fidelity and provide a reward dependent on the legged robot's state during deployment. An alternative method involves using a CNN to extract information from 3D voxel maps and the robot's internal state to predict a

single falling cost, as proposed in [23]. However, the network in this approach is trained without considering the presence of noise in exteroceptive observations commonly occurring in light-based external sensors. Furthermore, relying on a single cost prediction may be insufficient for an accurate assessment, as various factors affecting traversability, such as stumbling, getting stuck, and collisions, should be taken into consideration.

The previous method focused solely on pre-scanned terrain and neglected considerations for terrain properties or common noise during perception. To address this limitation, DogTouch [24] utilizes a force sensor at the foot's tip, and Feng [25] deploys an extended wheel to inspect forthcoming terrain. However, these sensors can only perceive a small area near the robot's feet. Another approach involves a neural network to identify non-traversable and flexible plants for mobile robots [26], providing a robust perception of plant properties, which is often misinterpreted by exteroceptive sensors. Additionally, another method implicitly perceives terrain properties by integrating multiple networks [27]. Furthermore, CNNs prove effectively extract unimportant features, such as noise in height maps, generating a noise-free height map suitable for real-time locomotion control [28]. Based on these findings, CNNs have the capability to reconstruct noise-free height maps from continuously fed, noisy height map data. Subsequently, this reconstructed height map serves as input to the traversability predictor for determining the associated traversability cost. To the best knowledge of the author, such robust perception has not been previously utilized in traversability predictors to predict multiple costs associated with traversability.

C. CONTRIBUTIONS

This paper proposes a robust traversability predictor for the operation of the robot in random terrains. It can decide whether a robot can safely transition from its current state to the next state without fail. The proposed method is devised based on four essential costs related to traversability assessment, including fall, stumble, stuck, and collision costs, even in the presence of noise in exteroceptive observations. Each of these costs holds notable implications for the feasibility of the robot's movements during traversal. The detailed explanation of these multiple costs is described in section III-C. The proposed method takes into consideration various sources of information, including terrain perception, the robot's internal state, the orientation of the robot, and its target command. It is expected to make accurate multiple-cost predictions by comprehending its surroundings and orientations.

Moreover, the robust traversability predictor is integrated with a CNN encoder-decoder architecture. This integration has the purpose of handling noise in exteroceptive observations. It can efficiently compress noisy measurements, filter out nonessential data such as noise, and result in noise-free information.

In contrast to existing approaches, this paper presents a robust traversability predictor using multiple costs designed for quadruped robots navigating in random terrains with noisy exteroceptive observations. This work contributes in the following ways:

- The proposed method can predict multiple costs associated with traversability, such as fall, stumble, stuck, and collision costs. It exceeds the limitation of previous works that only consider a single cost.
- The proposed approach integrates CNN encoder and decoder networks to mitigate the effect of noise in exteroceptive observations. This integration can extract irrelevant features, such as noise, and reconstruct noise-free observations. As a result, it outperforms previous methods that rely solely on noise-free observations and enhances overall performance immensely.
- The cost label for training the proposed method is calculated from physical simulation in Isaac Gym. This avoids human labeling that can lead to poor prediction since the quadruped robot is a severe nonlinear system comprising of complicated hardware and software.

II. PRELIMINARIES

The quadruped robot perceives the internal state of its body using proprioceptive sensors, such as the IMU and joint encoder. Define the internal state of the robot $v_B, \omega_B \in \mathbb{R}^3$ as the linear and angular velocities of the robot base B with respect to the global frame. $g \in \mathbb{R}^3$ represents the gravity vector of the robot. $q, \dot{q} \in \mathbb{R}^{12}$ are joint angular positions and velocities, respectively. The robot orientation can be represented with Euler angles $\varepsilon = [\phi\ \varphi\ \psi]$ as the roll, pitch, and yaw of the robot respectively. The robot uses actions $a_t \in \mathbb{R}^{12}$ as a joint torque for each joint to move its leg. To simplify the internal state of the robot, it can be expressed as:

$$\Phi = \left[v_B\ \omega_B\ g\ q_t\ \dot{q}_t\ a_{t-1}\right] \in \mathbb{R}^{45} \tag{1}$$

The desired velocity are given with high-level commands $c = [v_x^*\ v_y^*\ \omega_t^*]$ as the body velocity targets in the x and y directions, and the yaw rate target, respectively. Additionally, the robot is assumed to be able to perceive the terrain information by extracting the robo-centric height map using the light-based sensor. $\mathcal{H} : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^{80\times 80}$ represents the height map with size $-4\,m$ to $4\,m$ both in the x and y axis of the robot with a $10\,cm$ resolution. The robot used in this paper is an Anymal robot [29] which has length $0.9\,m$, width $0.53\,m$, and height $0.89\,m$.

III. METHODOLOGY

A. OVERVIEW

This section outlines the learning process for the robust traversability predictor. Initially, the locomotion controller is designed by utilizing the internal robot state, commands, and height map information as observations to generate optimal actions for navigating challenging terrains. This optimal

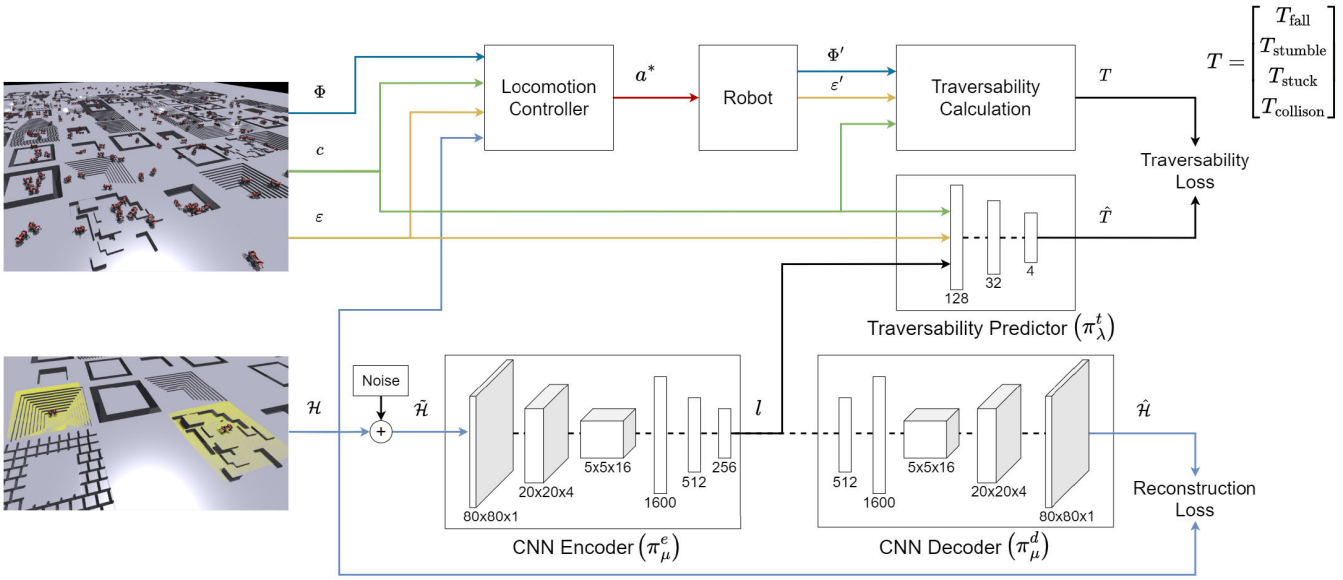


FIGURE 2. Illustration of the robust traversability predictor training process. The locomotion controller utilizes robot state Φ, commands c, robot's orientation ε, and height map $\mathcal{H}$ for optimal terrain navigation. Multiple costs (fall, stumble, stuck, and collision costs) are computed in real-time, while artificial noise is injected into the height map. The CNN encoder and decoder integration reconstructs a noise-free height map from the noisy height maps.

locomotion controller enables the robot to traverse various terrains successfully.

Subsequently, the optimal locomotion controller is assigned random commands to explore a different map than the map used during training. During this exploration, multiple costs are directly assessed from its current state. Various scenarios are labeled as costs in traversability, such as the robot falling on stairs, getting stuck in a pit, stumbling over stones, or colliding with obstacles. These labels are made exclusively from experiences in a physical simulator to mitigate potential biases and suboptimal cost labeling which are often associated with human-crafted labeling.

Additionally, artificial noise is intentionally introduced directly into the height map during exploration to reduce the reality gap in the learning process. This noise replicates common misinterpretations of terrains encountered in real-world scenarios. Consequently, the artificial noise effectively transforms the height map into a noisy height map. This noisy height map can make the traversability prediction become suboptimal. Therefore, the integration of CNN encoder and decoder is employed to mitigate the impact of noise. The integration of neural networks operates to reduce the effect of noise in the noisy height map and reconstruct a noise-free height map.

Moreover, this reconstructed noise-free height map, combined with the robot's internal state and commands, serves as an input for the traversability predictor network. This is the main network in this learning process. It is responsible for estimating multiple costs based on the current state. This prediction is crucial for the locomotion controller to make real-time adjustments while ensuring safe and efficient movement.

The robust traversability predictor undergoes training via classical supervised learning. It uses information from a noisy height map, the robot's internal state, the robot's orientation, and the target command as inputs. Furthermore, multiple cost labels and the original height map from exploration are used as outputs to train the network. An overview of the robust traversability predictor learning process is depicted in Figure 2.

Additionally, it is worth noting that while this paper employs specific robot models and controllers tailored to optimal locomotion, this framework is flexible. It allows for the use of alternative robots and locomotion controllers to adapt to different scenarios. This adaptability enhances the versatility and applicability of the system across various robotic applications.

B. LOCOMOTION CONTROLLER

The locomotion controller is designed to generate the optimal actions while simultaneously ensuring the robot's body balance. To accomplish this, the locomotion controller has access to important information, including the robot state Φ, orientation ε, and the ground truth terrain height map $\mathcal{H}$. Additionally, the controller also receives high-level navigation commands c, which are used to determine body linear and angular velocity targets.

The locomotion controller has to produce the optimal actions a^* to track the velocity targets. In this paper, the reinforcement learning approach in [30] is used to produce the optimal locomotion controller as an optimal policy. Reinforcement learning has the goal of maximizing the accumulative return over the given horizon. This can be assumed as Markov Decision Process (MDP), which can be

formulated by (s, a, r, s', a'). $s \in S$ is a state s in the set of states S, $a \in A$ is an action a in the set of actions A, and $r \in R$ is a reward r in the set of rewards R.

The policy can be represented using a multi-layer perceptron, denoted as $\pi^p(s_t|\theta)$, operating on the state s with parameter weights θ. The output of this policy yields a probability distribution over the next actions $a' \in A$ for state $s \in S$. In order to enable the policy to produce the optimal action a^*, the weights must be iteratively updated to maximize the expected return in the MDP until the optimal weights θ^* are achieved, and σ^p represents the variance of policy actions. This optimal policy can be expressed as follows:

$$a_t^* \sim \mathcal{N}\left(\pi^p\left(\Phi, c, \varepsilon, \mathcal{H}|\theta^*\right), \sigma^p I\right) \tag{2}$$

C. TRAVERSABILITY CALCULATION

A set of traversability costs are calculated directly from the next circumstance of the robot after implementing the optimal actions a^*. It considers the next internal state of the robot Φ' and the next orientation of the robot ε'. This calculation can be expressed as:

$$T = f(\Phi', \varepsilon', c|a^*) \in \mathbb{R}^4 \tag{3}$$

where T denotes multiple costs associated with traversability.

This paper introduces a comprehensive approach to robust traversability assessment, exceeding the existing works in the literature, which focused solely on a single cost factor. The proposed method considers four distinct costs, each representing a robot's capacity to navigate various properties of terrains. The considered multiple traversability costs include falling, stumbling, stuck, and collision costs.

Each of these traversability costs operates on a binary labeling. 1 represents the presence of a particular risk and 0 means it is safe. These binary labels facilitate a straightforward way of assessing the various challenges and potential hazards that the robot may encounter during its terrain traversal. The value of these costs can be expressed as:

$$T = \begin{bmatrix} T_{\text{fall}} & T_{\text{stumble}} & T_{\text{stuck}} & T_{\text{collison}} \end{bmatrix} \in \mathbb{R}^4 \tag{4}$$

$$T_{\text{fall}} = \begin{cases} 1 & \left|\phi'\right| > 45^\circ \quad \text{roll's constraint} \\ 1 & \left|\varphi'\right| > 45^\circ \quad \text{pitch's constraint} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$T_{\text{stumble}} = \begin{cases} 1 & \text{feet hitting vertical surfaces} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$T_{\text{stuck}} = \begin{cases} 1 & \left\|v'\right\|_2 - \|c\|_2 < 0.1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$T_{\text{collision}} = \begin{cases} 1 & \text{body collision with terrains} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

First, the falling cost is designed to evaluate the risk of the robot falling after it has violated the predetermined orientation's constraints on the terrain. This assessment relies on measurements of the robot's body orientation to ensure that the robot maintains a stable posture throughout its traversal. In this paper, the predetermined orientation threshold is set to $\pm 45^\circ$ in roll and pitch Euler angles. This value is considered the maximum angle where the robot cannot return to the original position due to its mechanical properties.

Second, the stumble cost operates as an alert detecting where the robot legs collide with vertical surfaces. These collisions can lead to a loss of stability and divert the robot's movement. This stumble cost preserves the robot's stability and anticipates forthcoming obstacles during its traversal.

Third, the stuck cost assesses the robot's mobility following the given target velocities. This evaluation serves to determine whether the robot is capable of moving or immobilizing within challenging terrain conditions. It guarantees the robot's ability to overcome terrain conditions and maintain the tracking movement.

Fourth, the collision cost indicates whether the robot's body has made contact with the terrain surface. This cost prevents unintended collisions of the robot with its surroundings, which could result in potential damage.

D. ARTIFICIAL NOISE

The distinction between the height map obtained from exteroceptive measurements and the original terrain can be ascribed to the presence of noise. This noise introduces blur to the original height map $\mathcal{H}$, making it become a noisy height map $\tilde{\mathcal{H}}$. In the context of exteroceptive perception, noise often arises due to the limitations of light-based sensors when perceiving reflective and deformable surfaces [28]. This noise potentially leads to inaccuracies in terrain perception. Moreover, Gaussian noise assumptions are employed to model this noise, as its characteristics cannot be precisely quantified with specific values [28]. This modeling approach utilizes $\sigma^{\mathcal{H}}$ as an artificial signal-to-noise ratio parameter to simulate the noisy height map. Furthermore, this assumption is applied in training a robust traversability model with a noisy height map. The noisy height map from sensor measurements is expressed as follows:

$$\tilde{\mathcal{H}} \sim \mathcal{N}(\mathcal{H}, \sigma^{\mathcal{H}} I) \tag{9}$$

E. CNN ENCODER

The noisy height map $\tilde{\mathcal{H}}$ is delivered into the CNN encoder to reduce the effect of noise. This is achieved through the CNN encoder's ability to compress the input data, effectively filtering out irrelevant information, such as noise in the height map. This CNN encoder network is denoted as π^e with weights μ and employs a Relu activation function. The network's size progressively decreases until it reaches the size of the latent output l, which serves as its output. This latent layer produces compressed information representing the estimated height map. The relationship between the CNN encoder and the noisy height map is represented as follows:

$$l = \pi^e\left(\tilde{\mathcal{H}}|\mu^*\right) \tag{10}$$

TABLE 1. Parameters of all maps.

	First Map		Second Map		Third Map	
	dist	level	dist	level	dist	level
upside slope	5%	30°	-	-	-	-
downside slope	5%	30°	7%	45°	14%	60°
rough slope	10%	30°	-	-	-	-
downstairs pyramid	35%	0.2 m	12%	0.23 m	14%	0.28 m
upstairs pyramid	25%	0.2 m	7%	0.23 m	14%	0.28 m
discrete obstacles	20%	0.15 m	12%	0.22 m	14%	0.28 m
stepping stones	-	-	12%	0.1 m	14%	0.1 m
flat terrain	-	-	26%	-	2%	-
gap terrain	-	-	12%	0.9 m	14%	0.9 m
pit terrain	-	-	12%	0.9 m	14%	0.9 m

F. CNN DECODER

The CNN decoder is assigned to reconstruct the predicted height map $\hat{\mathcal{H}}$ from the latent output l. This process is achieved by progressively increasing the network size until it matches the dimensions of the original height map. The decoder is denoted as π^d and shares weight μ parameters with the encoder network. It utilizes a ReLU as its activation function. The optimal construction of the height map is obtained by finding the optimal weights μ^*, which are updated iteratively by minimizing the error between the constructed height map $\hat{\mathcal{H}}$ and the ground-truth height map $\mathcal{H}$. The CNN decoder is expressed as follows:

$$\hat{\mathcal{H}} = \pi^d \left(l|\mu^*\right) \tag{11}$$

G. TRAVERSABILITY PREDICTOR

The role of the traversability predictor is to assess whether the upcoming terrain is feasible for traversal. It considers the command, robot orientation, and the latent output as its inputs. This latent output represents the noise-free height map extracted from the upcoming terrain. It also considers different command and robot orientation values, which can result in varying costs and accurate predictions. The traversability predictor is constructed using a multi-layer perceptron network denoted as π^t, with independently learned weights represented as λ, and it employs a Sigmoid activation function. The optimal weights λ^* for the predictor are determined through iterative updates to minimize the error between the ground truth multiple costs T and the predicted multiple costs $\hat{T}$. The optimal traversability predictor is represented as follows:

$$\hat{T} = \pi^t \left(c, \varepsilon, l|\lambda^*\right) \in \mathbb{R}^4 \tag{12}$$

IV. RESULTS

A. EXPERIMENTAL SETUP

In the experimental setup, three randomly generated maps are utilized in IsaacGym to validate the proposed method. These maps incorporate various challenging features to create diverse testing environments, including slopes, stairs, discrete stones, steps, gaps, and pits. Each map is characterized by a specific distribution ('dist' in the Table 1) and level for each terrain type. The distribution denotes how frequently a particular terrain appears on the map, while the level signifies the height of the vertical surfaces or the inclination of terrains.

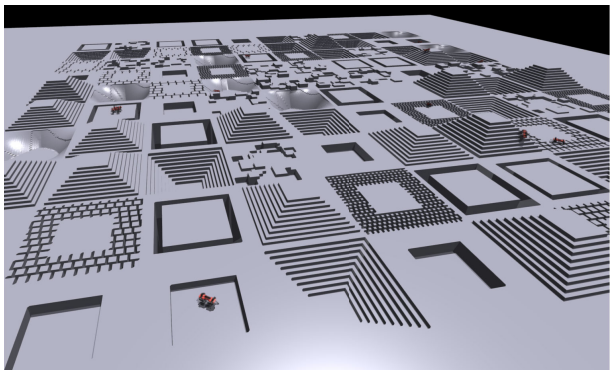


FIGURE 3. The third map has a higher distribution and level of terrain than the first and second maps to validate the robustness purpose.

The first map is identical to the one used in [30], featuring slopes, stairs, and discrete stone terrains. This first map serves as the training ground for the locomotion controller. In the second map, additional terrains such as stepping stones, gaps, and pits are introduced to train the traversability predictor and CNN encoder-decoder networks. The second map differs in the distribution and level of each terrain compared to the first map, facilitating learning in scenarios the robot never encounters during locomotion training. The second map is intentionally designed to be more challenging, with increasing distribution and level of each terrain. This is to make traversability costs more observable. The second map is illustrated in Figure 1.

Furthermore, a third map is introduced with the highest number of terrain distributions and levels among all maps, as illustrated in Figure 3. It means the third map has higher difficulty with more various terrain types, higher steps, and higher inclination degrees than other maps. This third map validates the robust comparison between the proposed method and an alternative method in the section IV-G. The distinctions between the maps are detailed in Table 1.

The training process is conducted on Ubuntu 20.04 with multiple Anymal robots [29] to enable parallel learning. This parallel learning can significantly reduce the training time. To efficiently distribute the training load, the entire program is implemented using PyTorch and executed on two NVIDIA®Titan V GPUs. Additionally, termination conditions, such as detecting the robot falling or reaching the maximum step limit, are incorporated. These conditions are considered to ensure the robot's safe operation and reset it to its initial state when necessary.

In this paper, the entire experiment is conducted solely in simulation to mitigate the uncertainties that may arise in the real world due to the complexity and non-linearities of the robot. These uncertainties can affect the dynamic characteristics of the robot and deviate from the main objective. The full potential of the robot can be realized by avoiding these uncertainties and focusing on the main objective of the proposed method. Features such as body inertia, sensor bias, and friction are also incorporated to minimize the reality gap. Several existing studies in the

literature also employ similar simulation-based approaches to mitigate the impact of such uncertainties [7], [31], [32].

B. TRAINING THE LOCOMOTION CONTROLLER

The locomotion controller's primary objective is to navigate across as much terrain as possible while maintaining the robot's balance. The locomotion controller is trained as a policy using the Proximal Policy Optimization (PPO) algorithm [33] and locomote in the first map. The PPO algorithm refines the policy by minimizing the actor loss, continually updating the network parameters until it converges to the optimal weights θ^*. Parallel learning regimes are implemented to address the issue of sample inefficiency often associated with reinforcement learning. This can be achieved by employing 1000 robots that significantly reduce the training time. This training is completed in one hour. The effectiveness of the optimized policy is demonstrated by its ability to successfully navigate diverse and randomly generated terrains without experiencing falls.

C. ADDITIONAL NOISE

Once the optimal locomotion controller is obtained, the subsequent phase of this approach involves training the traversability predictor using noisy exteroceptive observations. This replicates real-world conditions where sensors often encounter challenges such as reflective or deformable surfaces. This paper incorporates artificial noise directly into the robot's exteroceptive observations, as outlined in the equation (9). In this setup, the signal-to-noise ratio parameter $\sigma^{\mathcal{H}}$ is set to a value of $-10\% \leq \sigma^{\mathcal{H}} \leq 10\%$ specifically for height map measurements. The signal-to-ratio parameter value used in this paper is the same as the one employed in the existing results in [30]. This artificial noise is essential to ensure that the traversability predictor can make robust assessments even in the presence of noise caused by exteroceptive sensors.

D. TRAVERSABILITY TRAINING

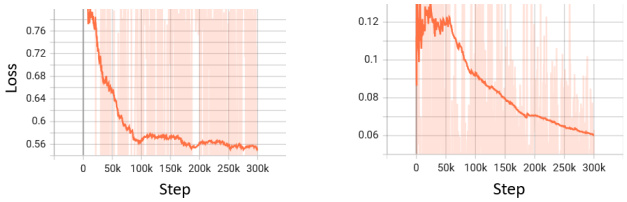
The robust traversability predictor has the objective of predicting multiple costs for upcoming terrain based on inputs, including the recent command, robot orientation, and a noisy height map.

The learning process of this method involves utilizing the second map, which is different from the first map used during the locomotion controller training, as illustrated in Figure 1. Initially, the optimal locomotion controller receives a random command to explore the map. The height map is then extracted and subjected to artificial noise to generate a noisy height map. Subsequently, the costs are labeled directly based on the robot's experience during the exploration.

To ensure accurate predictions, the network first addresses the need to eliminate noise from the height map. This task is achieved by utilizing a CNN encoder and decoder, working collaboratively to reduce the effect of noise within the noisy height map input and generate a noise-free representation as output. Subsequently, the traversability predictor processes

TABLE 2. Hyperparameters in the traversability training process.

Max step	300000
Number of robots	1000
Batch size	2000
Epoch	10
Learning rate	0.0003



(a) Reconstruction loss (b) Traverseability loss

FIGURE 4. Losses in the traversability training process.

the noise-free height map, the robot's internal state, and command, to calculate and predict multiple costs. This comprehensive approach enables the traversability predictor to provide precise assessments of the robot's ability to traverse forthcoming terrains.

In this paper, two distinct losses are introduced. The first is the reconstruction loss, which quantifies the error between the reconstructed and original height maps without noise. It employs the Mean Square Error (MSE) metric for error calculation. The second loss is the traversability loss, which measures the error between the predicted and ground truth costs. The Binary Cross-Entropy (BCE) metric is used for error computation. These loss functions can be expressed as follows:

$$L^{MSE}_{\text{reconstruction}} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{H}_i - \hat{\mathcal{H}}_i\right)^2 \tag{13}$$

$$L^{BCE}_{\text{traversability}} = -\frac{1}{N}\sum_{i=1}^{N}\left(T_i\, log\hat{T}_i + (1-T_i)\, log(1-\hat{T}_i)\right) \tag{14}$$

To get a noise-free height map and accurate multiple cost prediction, classical supervised learning is employed with the hyperparameter, as shown in Table 2. Its main objective is to minimize the reconstruction and traversability losses by updating the weight until the optimal CNN encoder-decoder and traversability predictor weights (μ^*, λ^*) are obtained. The optimization problems can be expressed as:

$$\begin{aligned}\mu^* &= \arg\min_{\mu} L^{MSE}_{\text{reconstruction}} \\ &= \arg\min_{\mu} \frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{H}_i - \hat{\mathcal{H}}_i\right)^2\end{aligned} \tag{15}$$

$$\begin{aligned}\lambda^* &= \arg\min_{\lambda} L^{BCE}_{\text{traversability}} \\ &= \arg\min_{\lambda} \frac{-1}{N}\sum_{i=1}^{N}\left(T_i\, log\hat{T}_i + (1-T_i)\, log(1-\hat{T}_i)\right)\end{aligned} \tag{16}$$

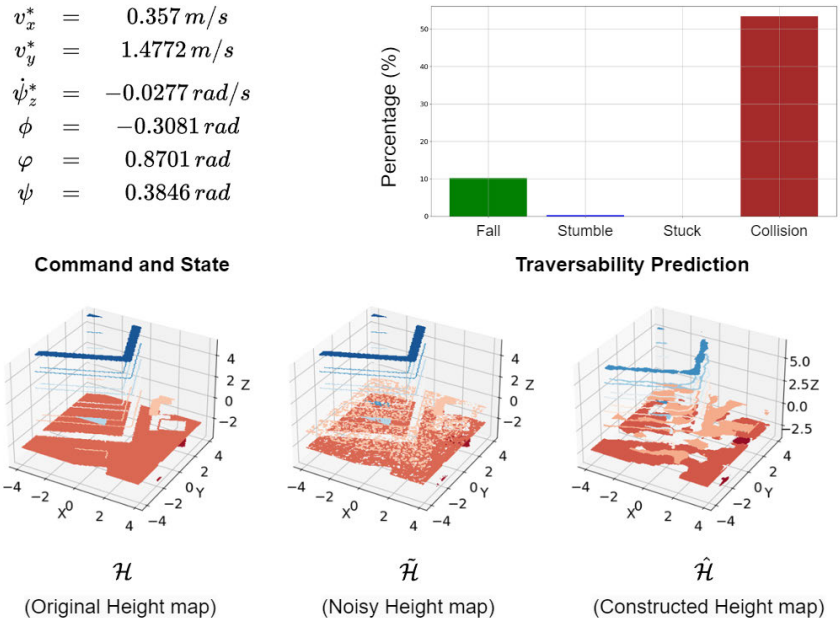


FIGURE 5. Traversability prediction based on the internal state and a noisy height map.

The losses from the training of the proposed method are illustrated in Figure 4. As depicted in the figure, both the reconstruction and traversability losses converge to zero. This indicates the success of the proposed method in learning to reconstruct a noise-free height map from a noisy height map and accurately predict multiple costs associated with traversability.

E. TRAVERSABILITY VALIDATION

Figure 5 illustrates the results derived from the proposed method. The process begins with the robot extracting the original height map $\mathcal{H}$ from the simulation using an exteroceptive sensor. Subsequently, artificial noise, equivalent to a signal-to-noise ratio of 10%, is introduced into the original height map. This results in the creation of a noisy height map denoted as $\tilde{\mathcal{H}}$.

The subsequent step involves the integrated operation of the CNN encoder and decoder networks. It aims to extract the noise from the noisy height map $\tilde{\mathcal{H}}$ and reconstruct a height map with reduced noise effects in perception, denoted as $\hat{\mathcal{H}}$. The visual representation in the figure highlights the effectiveness of the integration of CNN encoder and decoder networks in successfully reconstructing a noise-free height map from a noisy height map.

The performance of the proposed method is further validated through its capability to predict the percentage of costs incurred as the robot encounters an upcoming terrain. As illustrated in Figure 5, the traversability predictor adeptly predicts the likelihood of various costs by leveraging information from the internal state, command inputs, and the constructed height map.

Specifically, the figure indicates that the robot faces a 10% chance of falling, a 0.5% chance of stumbling, no probability of getting stuck (0%), and a significant 50% chance of colliding when confronted with an upcoming vertical object. This validation highlights the method's ability to reduce the noise effect and provide predictions regarding potential incidents in traversing upcoming terrains.

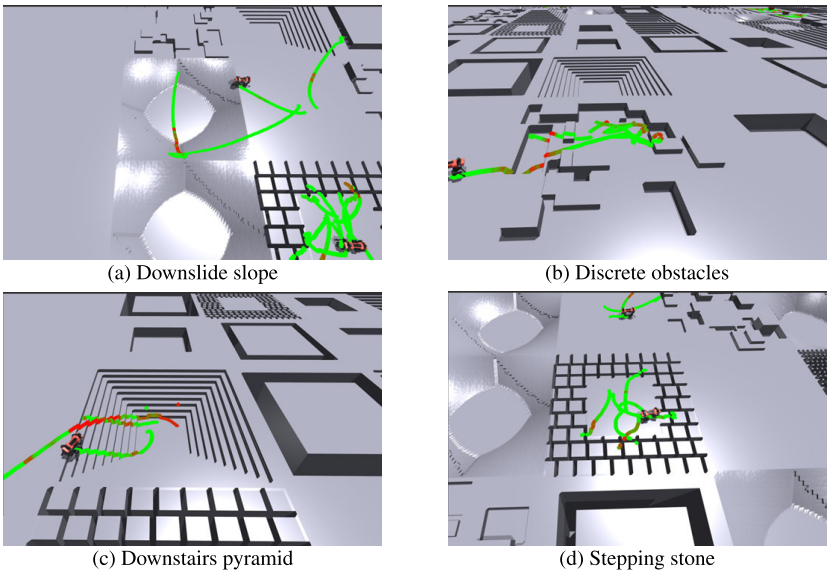
F. VALIDATE USING COLOR

The method's ability is further demonstrated through extended validation using colors across various challenging terrains, as depicted in Figure 6. This visual representation proves the robot's ability to assess its capacity to navigate various forthcoming challenging terrains safely.

A color spectrum that transitions from green to red, has been thoughtfully employed to indicate traversability percentages. The green color denotes that the robot can safely traverse the terrain without unnecessary risk ($< 50\%$ chance of falling), while the red color indicates the presence of a potential risk of the robot falling or encountering obstacles ($\geq 50\%$ chance of falling).

Furthermore, the robot demonstrates its proficiency in predicting traversability across terrains, including slopes, rugged stone surfaces, stairs, and stepping stones. This variety of terrains emphasizes the adaptability and versatility of the proposed method. It makes a worthwhile assessment for navigating complex terrains with precision.

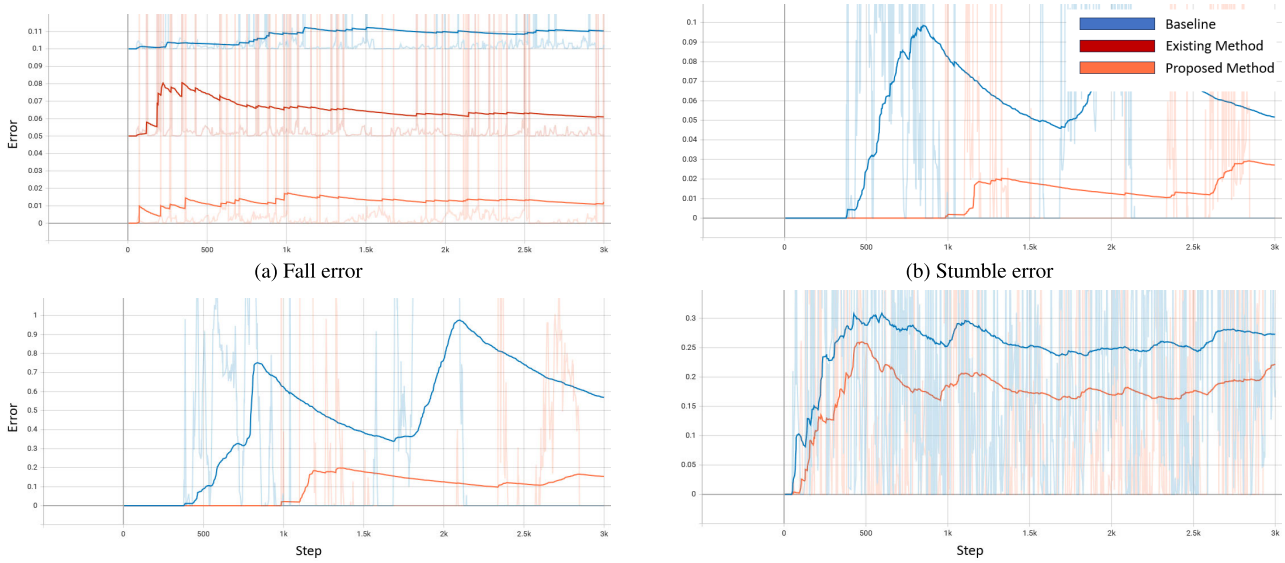
In Figure 6a, the robot assesses the traversability of a downslide slope terrain. In the flat area of the slope, the mask is dominated by green, indicating that the area is safe. However, on the inclined side of the slope, the color shifts to orange-red, signifying a higher risk of failure. Figure 6b illustrates the assessment of the side of discrete obstacles with a vertical surface, where the orange-red color denotes a potential risk of the robot failing.



(a) Downslide slope (b) Discrete obstacles

(c) Downstairs pyramid (d) Stepping stone

FIGURE 6. The fall cost prediction in the color spectrum on various terrains. The green color indicates safety (< 50%) and the red color indicates the robot has a high possibility of falling (≥ 50%).



(a) Fall error (b) Stumble error

(c) Stuck error (d) Collision error

FIGURE 7. A comparison of the robustness between the proposed method and the baseline approach.

In the downstairs pyramid with a four-sided surface (Figure 6c), different colors (i.e., green and red) are observed even when the robot is on the same side of the stairs. When the robot moves perpendicularly to the stair steps, the proposed method predicts that the robot can move safely, which is masked by the green color. On the other hand, when a different command is given, which asks the robot to move diagonally with the stair steps, a bigger risk of falling is predicted, which is meant by the red color. What can be observed from Figure 6c is that both the commands given to the robot and the orientation are important in determining the traversability of the robot in addition to the terrain properties. This is in sharp contrast with the existing results in which the focus is solely on the terrain without considering the command and orientation.

Furthermore, in Figure 6d, the robot can assess the risk of stepping stones even before making contact with the upcoming terrain, as indicated by the red color near the stepping stones.

The video demonstration of the proposed method's validation can be found at: https://youtu.be/emqNSuAdi7E

G. VALIDATE THE ROBUSTNESS

To assess the robustness and efficiency of the proposed method, this paper conducted a performance comparison with both a baseline approach and an existing method outlined

in [23]. The baseline approach represents a multiple-cost prediction model trained in the absence of noise in exteroceptive observations. In contrast, the existing method is a cost prediction model trained without noise and solely relies on a single fall cost. The primary objective of this comparative analysis is to validate the impact of noise and the significance of considering multiple costs in traversability prediction.

The third map is employed to examine the capabilities of the proposed method, as visually depicted in Figure 3. Unlike the first and second maps, this map poses a higher difficulty level and has never been used during the training phase. The map's composition aligns with the second map. However, it differs from the second map by featuring more uneven terrain, the absence of flat surfaces, and an elevated level of terrain complexity.

The graphical illustration in Figure 7 serves as a visual validation of the robustness of the proposed method. Comparative analysis reveals that the proposed method consistently produces the smallest errors compared with the baseline and existing approaches. This figure shows the predictive accuracy of the proposed method across various costs, including fall, stumble, stuck, and collision costs. It demonstrates the method's proficiency in handling multiple cost considerations.

The results of this validation demonstrate that incorporating multiple costs enables a more accurate assessment of traversability for forthcoming terrains. It overcomes the limitations of the existing approach, which relies solely on a single fall cost. Moreover, the validation shows the robustness of the proposed method in effectively mitigating the impact of noise in exteroceptive observations, emphasizing that the proposed method has advantages over the baseline approach. This evaluation demonstrates the method's capabilities as a robust method for traversability prediction in the presence of real-world complexities.

V. CONCLUSION

This paper presents a robust traversability predictor by employing both multiple costs and a combination of a CNN encoder-decoder and a multi-layer perceptron. This network demonstrates its capability to effectively handle noise that is commonly encountered in external sensors during exteroceptive observations. The collaborative operation of the CNN encoder and decoder is crucial in extracting important information from noisy height maps while filtering out unnecessary values, such as noise. Moreover, it possesses the ability to predict a set of costs that collectively affect the traversability. It contrasts to tackle the limitations associated with single-cost predictions. Additionally, the utilization of simulation-based labeling proves essential in obtaining precise cost estimations, as opposed to relying solely on human-crafted labels.

A limitation of the proposed method lies in its reliance on a height map as an exteroceptive observation. Height maps offer computational efficiency. However, they fail to fully leverage the three-dimensional (3D) terrain features. Notably, it is error-prone in the face of overhanging obstacles, multi-floor structures, or low-ceiling scenarios, resulting in the exclusion of certain poses in the 3D space from the path planning considerations [23]. Future work will involve integrating the proposed method with a voxel-based 3D map to address this limitation. It aims to enhance terrain perception and overcome the challenges associated with the limitations of height maps. Subsequently, it will improve performance and the applicability of the proposed approach in complex and various environments.

REFERENCES

[1] M. Zucker, J. A. Bagnell, C. G. Atkeson, and J. Kuffner, "An optimization approach to rough terrain locomotion," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 3589–3595.
[2] P. D. Neuhaus, J. E. Pratt, and M. J. Johnson, "Comprehensive summary of the institute for human and machine cognition's experience with LittleDog," *Int. J. Robot. Res.*, vol. 30, no. 2, pp. 216–235, Feb. 2011.
[3] S. Sharma, L. Dabbiru, T. Hannis, G. Mason, D. W. Carruth, M. Doude, C. Goodin, C. Hudson, S. Ozier, J. E. Ball, and B. Tang, "CaT: CAVS traversability dataset for off-road autonomous driving," *IEEE Access*, vol. 10, pp. 24759–24768, 2022.
[4] J. Seo, S. Sim, and I. Shim, "Learning off-road terrain traversability with self-supervisions only," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 4617–4624, Aug. 2023.
[5] D. D. Fan, A.-A. Agha-Mohammadi, and E. A. Theodorou, "Learning risk-aware costmaps for traversability in challenging environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 279–286, Jan. 2022.
[6] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Nov. 2014, pp. 279–286.
[7] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1560–1567, Jul. 2018.
[8] P. Fernbach, S. Tonneau, O. Stasse, J. Carpentier, and M. Taïx, "C-CROC: Continuous and convex resolution of centroidal dynamic trajectories for legged robots in multicontact scenarios," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 676–691, Jun. 2020.
[9] M. Endo and G. Ishigami, "Active traversability learning via risk-aware information gathering for planetary exploration rovers," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11855–11862, Oct. 2022.
[10] J. Li, Y. Cheng, J. Zhou, J. Chen, Z. Liu, S. Hu, and V. C. M. Leung, "Energy-efficient ground traversability mapping based on UAV-UGV collaborative system," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 69–78, Mar. 2022.
[11] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1373–1385, Apr. 2013.
[12] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, "Learning ground traversability from simulations," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1695–1702, Jul. 2018.
[13] C. Sevastopoulos and S. Konstantopoulos, "A survey of traversability estimation for mobile robots," *IEEE Access*, vol. 10, pp. 96331–96347, 2022.
[14] J. Bae, J. Seo, T. Kim, H.-G. Jeon, K. Kwak, and I. Shim, "Self-supervised 3D traversability estimation with proxy bank guidance," *IEEE Access*, vol. 11, pp. 51490–51501, 2023.
[15] J. Hyun, S. Woo, and E. Kim, "Street floor segmentation for a wheeled mobile robot," *IEEE Access*, vol. 10, pp. 127601–127609, 2022.
[16] M. V. Gasparino, A. N. Sivakumar, Y. Liu, A. E. B. Velasquez, V. A. H. Higuti, J. Rogers, H. Tran, and G. Chowdhary, "WayFAST: Navigation with predictive traversability in the field," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10651–10658, Oct. 2022.
[17] S. Xie, R. Song, Y. Zhao, X. Huang, Y. Li, and W. Zhang, "Circular accessible depth: A robust traversability representation for UGV navigation," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4875–4891, Dec. 2023.
[18] G. G. Waibel, T. Löw, M. Nass, D. Howard, T. Bandyopadhyay, and P. V. K. Borges, "How rough is the path? Terrain traversability estimation for local and global path planning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16462–16473, Sep. 2022.

[19] J. Seo, T. Kim, K. Kwak, J. Min, and I. Shim, "ScaTE: A scalable framework for Self- supervised traversability estimation in unstructured environments," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 888–895, Feb. 2023.
[20] B. Yang, L. Wellhausen, T. Miki, M. Liu, and M. Hutter, "Real-time optimal navigation planning using learned motion costs," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 9283–9289.
[21] O. Elharrouss, Y. Akbari, N. Almaadeed, and S. Al-Maadeed, "Backbones-review: Feature extraction networks for deep learning and deep reinforcement learning approaches," 2022, *arXiv:2206.08016*.
[22] L. Gan, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, "Energy-based legged robots terrain traversability modeling via deep inverse reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 8807–8814, Oct. 2022.
[23] J. Frey, D. Hoeller, S. Khattak, and M. Hutter, "Locomotion policy guided traversability learning using volumetric representations of complex environments," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Oct. 2022, pp. 5722–5729.
[24] N. D. Weerakkodi Mudalige, E. Nazarova, I. Babataev, P. Kopanev, A. Fedoseev, M. A. Cabrera, and D. Tsetserukou, "DogTouch: CNN-based recognition of surface textures by quadruped robot with high density tactile sensors," in *Proc. IEEE 95th Veh. Technol. Conf.*, Jun. 2022, pp. 1–5.
[25] L. Feng, T. Miao, X. Jiang, M. Cheng, Y. Hu, W. Zhang, and A. Song, "An instrumented wheel to measure the wheel–terrain interactions of planetary robotic wheel-on-limb system on sandy terrains," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.
[26] S. Matsuzaki, H. Masuzawa, and J. Miura, "Image-based scene recognition for robot navigation considering traversable plants and its manual annotation-free training," *IEEE Access*, vol. 10, pp. 5115–5128, 2022.
[27] F. Muhamad, J.-S. Kim, and J.-H. Park, "Learning robust perception-based controller for quadruped robot," *IEEE Access*, vol. 11, pp. 94497–94505, 2023.
[28] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Sci. Robot.*, vol. 7, no. 62, Jan. 2022, Art. no. eabk2822.
[29] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Bloesch, H. Kolvenbach, M. Bjelonic, L. Isler, and K. Meyer, "ANYmal—Toward legged robots for harsh environments," *Adv. Robot.*, vol. 31, no. 17, pp. 918–931, Sep. 2017.
[30] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proc. Conf. Robot Learn.*, 2021, pp. 91–100.
[31] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter, "DeepGait: Planning and control of quadrupedal gaits using deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3699–3706, Apr. 2020.
[32] M. Bloesch, M. Burri, H. Sommer, R. Siegwart, and M. Hutter, "The two-state implicit filter recursive estimation for mobile robots," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 573–580, Jan. 2018.
[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–12.
[34] L. Gan, Y. Kim, J. W. Grizzle, J. M. Walls, A. Kim, R. M. Eustice, and M. Ghaffari, "Multitask learning for scalable and dense multilayer Bayesian map inference," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 699–717, Feb. 2023.
[35] J. Lee, G. Park, I. Cho, K. Kang, D. Pyo, S. Cho, M. Cho, and W. Chung, "ODS-bot: Mobile robot navigation for outdoor delivery services," *IEEE Access*, vol. 10, pp. 107250–107258, 2022.
[36] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter, "Perceptive locomotion in rough terrain—Online foothold optimization," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5370–5376, Oct. 2020.
[37] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini, "Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 1096–1103.
[38] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3019–3026, Oct. 2018.
[39] A. Escande, A. Kheddar, and S. Miossec, "Planning support contact-points for humanoid robots and experiments on HRP-2," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 2974–2979.
[40] K. Hauser, T. Bretl, and J. C. Latombe, "Non-gaited humanoid locomotion planning," in *Proc. 5th IEEE-RAS Int. Conf. Humanoid Robots*, Dec. 2005, pp. 7–12.

FIKIH MUHAMAD received the B.S. degree in electrical engineering from Universitas Indonesia, in 2020. He is currently pursuing the master's degree with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology. His research interests include robotic and artificial intelligence.

JUNG-SU KIM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea University, in 1998, 2000, and 2005, respectively. From 2005 to 2008, he was a Postdoctoral Researcher with Seoul National University, the University of Stuttgart, Germany, and the University of Leicester, U.K. Since 2009, he has been with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology. His research interests include MPC, machine learning, and their applications.

JAE-HAN PARK (Member, IEEE) received the B.S. degree in electronic engineering from Dong-A University, in 1998, the M.S. degree in electronic engineering from Pusan National University, in 2000, and the Ph.D. degree in electrical engineering from Korea University, in 2017. He is currently a Principal Research Scientist with the AI Robot Research and Development Department, Korea Institute of Industrial Technology, Ansan, South Korea. His research interests include 3D perception for intelligent robots, robot motion planning, and AI-based robot control.