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

# VOCALoco: Viability-Optimized Cost-aware Adaptive Locomotion

Stanley Wu[1], Mohamad H. Danesh[1], Simon Li[2], Hanna Yurchyk[1], Amin Abyaneh[2], Anas El Houssaini[1], David Meger[1], and Hsiu-Chin Lin[1,2]

***Abstract*—Recent advancements in legged robot locomotion have facilitated traversal over increasingly complex terrains. Despite this progress, many existing approaches rely on end-to-end deep reinforcement learning (DRL), which poses limitations in terms of safety and interpretability, especially when generalizing to novel terrains. To overcome these challenges, we introduce VOCALoco, a modular skill-selection framework that dynamically adapts locomotion strategies based on perceptual input. Given a set of pre-trained locomotion policies, VOCALoco evaluates their viability and energy-consumption by predicting both the safety of execution and the anticipated cost of transport over a fixed planning horizon. This joint assessment enables the selection of policies that are both safe and energy-efficient, given the observed local terrain. We evaluate our approach on staircase locomotion tasks, demonstrating its performance in both simulated and real-world scenarios using a quadrupedal robot. Empirical results show that VOCALoco achieves improved robustness and safety during stair ascent and descent compared to a conventional end-to-end DRL policy.**

***Index Terms*—Robot Safety, Legged Robots, Reinforcement Learning, Deep Learning Methods**

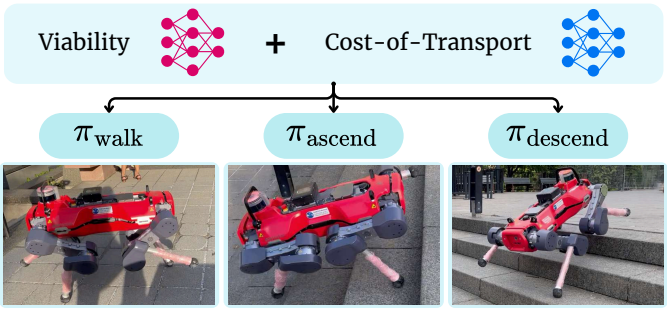


Fig. 1: Overview of VOCALoco. Given a heightmap of the local terrain, two high-level modules predict: (i) the viability and (ii) the Cost-of-Transport (CoT) of executing each skill of executing each skill. With both predictions at hand, we first filter unsafe skills. Then, among the safe skills, we select the skill with the lowest predicted energy expenditure as the final policy to execute on the robot. The three example images show the ANYmal-D robot switching between different policies depending on the terrain type.

## I. INTRODUCTION

LEGGED robots are capable of traversing complex and unstructured terrains that could be inaccessible to wheeled robots [1], [2]. As a result, the applications of legged robots have been of great interest in problems such as search-and-rescue, remote inspection, and autonomous exploration. While legged robot remains an inherently difficult problem, current research methods have enabled quadruped robots to traverse complex terrains by addressing the challenges of controlling underactuated systems and accurately perceiving an environment that can be highly variable and uncertain.

Deep reinforcement learning (DRL) has enabled many of the advances in quadruped control over complex terrains [2], [3], [4], [5], [6], [7], [8]. While most DRL methods employ an end-to-end approach, such policies perform poorly outside of their training environment and offer little transparency into their decision-making process. As an alternative, we propose a hierarchical and modular structure that selects multiple specialized policies, each tailored to a specific training distribution. Our method has more control over the behaviour of the robot, as we can tailor locomotion policies to a specific terrain. Our method is also safer and interpretable, as we evaluate the safety and energy-efficiency of each policy, allowing us to assess the behaviour of the robot at runtime. However, deploying multiple low-level policies requires a robust high-level strategy to switch between them.

Inspired by animal locomotion, where gait transitions occur naturally in response to environmental demands, prior work examined adaptive gait selection driven by base velocity. [9], [10], [11], [12], [13], [14]. While these methods are effective on flat and slightly uneven terrain, we focus on switching between gaits or skills on unstructured terrain. Few works have studied *non* end-to-end skill switching on unstructured terrain, namely [14], [7]. We improve on [7] by simplifying the high-level skill selection with a supervised learning approach inspired by [14].

We propose **V**iability-**O**ptimized **C**ost-aware **A**daptive **Loco**motion (VOCALoco), a hierarchical skill selection framework that uses the robot's perception of its local terrain to determine the most suitable locomotion policy. Assuming access to a set of low-level locomotion skills, our approach employs a high-level decision-maker to predict the viability and energy cost of each skill on the local terrain, enabling adaptive skill selection, conceptually illustrated in Figure 1. Our key contributions are as follows:

- We introduce a method to predict the viability and energy cost of executing different locomotion policies on perceived terrain structures, enabling safe and energy-aware skill switching.

Manuscript received: June, 15, 2025; Revised October, 9, 2025; Accepted October, 24, 2025.

This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Google DeepMind Scholarship, FRQNT doctoral training scholarships, and the NSERC Discovery Grant.

[1]Stanley Wu, Mohamad H. Danesh, Hanna Yurchyk, Anas El Houssaini, David Meger, and Hsiu-Chin Lin are with the School of Computer Science, McGill University, Montreal, Canada stanley.wu@mail.mcgill.ca

[2]Simon Li, Amin Abyaneh, and Hsiu-Chin Lin are with the Department of Electrical and Computer Engineering, McGill University, Montreal, Canada

Digital Object Identifier (DOI): see top of this page.

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

- Our modular framework provides greater interpretability than standard end-to-end policies, allowing experts to fine-tune or extend the set of low-level skills.
- All components are trained entirely in simulation using automatically generated labels, requiring no manual annotation or human intervention.
- We validate our approach on the ANYmal-D [15] robot in both simulation and the real-world with a zero-shot deployment.

## II. Related Work

### A. DRL for Legged Robots

In recent years, DRL has shown impressive results for legged robots. Part of the success can be attributed to closing the sim-2-real gap with domain and dynamics randomization [16], [17], learning to estimate terrain and environment properties [18], [19], [20], learning with GPU-acceleration and a game-style curriculum [21], and simulating realistic actuator dynamics [22].

Several works have demonstrated that incorporating environmental perception for DRL locomotion is essential for avoiding obstacles, traversing challenging terrain and generalizing to novel terrain [1], [23], [24], [25], [26], [27], [28], [29], [30]. The benefits of perception to generalize to diverse terrains has also been demonstrated in long-distance locomotion [2], [3], in parkour terrains [4], [5], [6], [7], [8], and in other highly unstructured terrains [26], [31], [32]. However, these approaches typically rely on a single end-to-end policy, which limits their flexibility: they are difficult to interpret and hard to adapt to new tasks.

One work on quadruped locomotion for parkour [7] employed a similar hierarchical architecture, where the high-level policy selects low-level locomotion policies. However, their high-level policy is a deep reinforcement learning policy, lacking interpretability. In contrast, our high-level policy is composed of two modules predicting the viability and energy consumption of each low-level policy, resulting in a more explainable and safety-aware approach. Furthermore, we used supervised learning to train our high-level modules, which avoids the complexity of using DRL.

### B. Geometric Traversability

One popular approach to estimate traversability on rigid terrain structures is by rolling out robot trajectories in simulation and automatically collecting synthetic traversability data [33], [34], [35], [36]. The advantage of this approach is the ability to generate infinite data automatically without any human labelling. We use this approach to estimate geometric traversability for each of our policies. Specifically, we collect simulation data similarly to [35] and extend this approach with skill selection.

### C. Skill and Gait Switching for Legged Robots

Recent studies explored quadruped skill and gait switching. [37] and [38] both trained DRL policies that are capable of adapting their behaviours to traverse different terrain types. However, both frameworks required a human operator to vary some parameters of their methods at runtime.

Other prior works have explicitly embedded traversability estimation into gait switching [39], [40]. However, these studies primarily focused on determining whether a terrain is traversable for navigation purposes. Further works have implicitly adapted locomotion behaviour based on terrain structure and difficulty, by adapting the footsteps [41] or adapting the whole-body movement [42].

Drawing parallel analogies to how and why animals switch between different gaits, studies have demonstrated the influence of robot energy consumption, or CoT, on gait patterns [9], [10], [11]. The link between CoT and gait switching has been further highlighted by a series of work on central-pattern-generators (CPG) for quadruped robots [12], [13], [14]. However, these works primarily focus on gait switching based on base velocities on flat or slightly uneven terrain, instead of more complex terrains. [14] address this issue by investigating biomechanical factors inducing gait switching on terrains with consecutive gaps. In contrast, VOCALoco learns to predict these biomechanical factors explicitly, using viability and CoT metrics estimated from perception, to perform skill selection on more diverse terrain structures.

## III. Problem Statement and Definitions

### A. Problem Formulation

We consider a legged robot whose state is denoted by $\mathbf{x} \in \mathbb{R}^d$. This state comprises the robot's joint positions $\mathbf{q}$, joint velocities $\dot{\mathbf{q}}$, joint torques $\tau$, base linear velocity $\mathbf{v}$, base angular velocity $\boldsymbol{\omega}$, and a local terrain representation $\mathbf{H} \in \mathbb{R}^{h \times w}$.

Our local terrain representation $\mathbf{H}$ is a heightfield. To ensure it covers the local terrain around and under the base of the robot to prevent it from getting stuck on certain terrains, we choose a rectangular region that expands $2m$ forward and $1m$ backward from the robot's base, and a width of $1m$. The points in $\mathbf{H}$ are spaced by $10cm$, yielding a $31 \times 11$ matrix.

The robot is equipped with a repertoire of $K$ distinct locomotion skills, represented as a set of parameterized policies $\Pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$. These policies may be obtained via diverse learning or optimization strategies, including DRL, imitation learning, or trajectory optimization. We impose no restrictions on the source or training method of these policies.

Given the current state $\mathbf{x}$ and observation $\mathbf{H}$, our objective is to design a skill selection mechanism that chooses the most energy-efficient policy $\pi_k \in \Pi$ from among those deemed viable i.e., capable of safely and stably executing locomotion in the current terrain context.

### B. Viability

The term viability refers to the ability of the robot to achieve its designated task. In VOCALoco, we define the notion of viability as the likelihood that a locomotion skill achieves our desired task of moving forward over a fixed horizon safely without any base collisions.

Several related works discussed in Section II-B adopt the term traversability to describe terrain assessment in order to avoid dangerous regions. In this work, we deliberately use

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

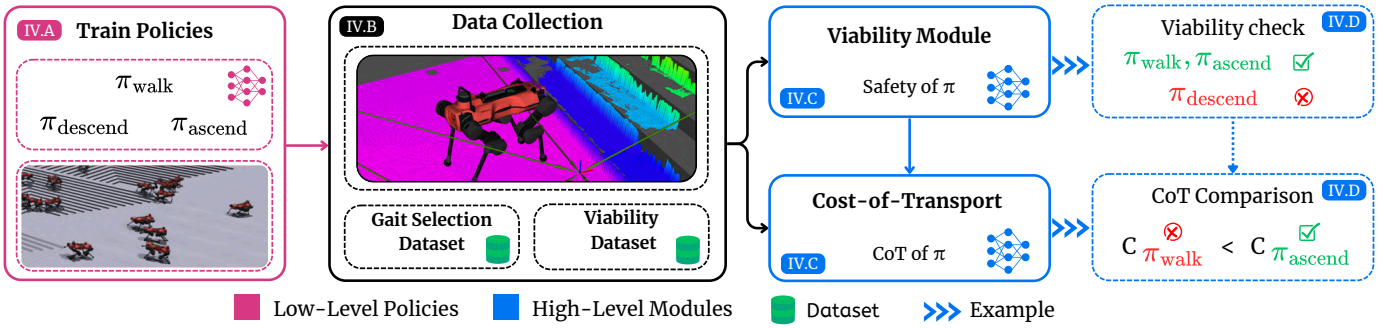


Fig. 2: In VOCALoco, we start by training low-level locomotion policies: a walking policy, an ascending policy, and a descending policy. Then, we perform rollouts with these policies, collecting data that will train the high-level policies: the viability and the CoT modules.

the term viability instead. This choice reflects the objective of VOCALoco to focus on estimating whether the robot can move forward over a fixed horizon without a fatal collision. Furthermore, this terminology is consistent with the usage in [14], which similarly emphasizes progress-oriented viability rather than a comprehensive traversability analysis.

### C. Cost of Transport

CoT measures the energy required to move a unit weight over a unit distance. In the context of gait switching, CoT plays a central role in determining when and why a robot should switch gaits. Formally, the CoT is defined as

$$c = \frac{\sum_{t=0}^{T} \boldsymbol{\tau}_t^\top \dot{\mathbf{q}}_t \Delta t}{mgd} \tag{1}$$

where $T$ is the total timesteps required to complete the task, $\boldsymbol{\tau}_t$ is total joint torques computed at time $t$, $\dot{\mathbf{q}}_t$ is the joint velocities at time $t$, and $\Delta t$ is the simulation timestep, $m$ is the mass of the robot, $g$ is the gravity constant, and $d$ is the distance travelled during the period $T$.

Prior work has demonstrated that different gaits (e.g., walking, trotting, galloping) have different CoT profiles depending on the terrain, speed, and payload (section II-C). An underlying assumption is that animals can reduce their overall energy consumption by switching to the gait with the lowest CoT for the current conditions [43], [44], [45]. In our work, we assume that locomotion policies, whether obtained through reinforcement learning or optimal control, are energy efficient on the terrain they were trained. We will learn a model that predicts the CoT over a short fixed horizon.

## IV. METHODOLOGY

VOCALoco follows a hierarchical structure to safe and robust quadruped locomotion, as outlined in Figure 2. Assuming the robot is equipped with different low-level locomotion skills, we proposed a high-level decision-making component that reads the robot's perception of the local terrain to evaluate each available locomotion skill. Over a short horizon, the viability module estimates the likelihood of successful traversal (i.e., viability), while the CoT module predicts the expected energy cost. Based on these predictions, we select the skill that is both safe and energy-efficient. We first train a set of low-level locomotion policies specialized for different terrain types in Section IV-A. These policies are then rolled out in simulation to generate synthetic data in Section IV-B, and to ultimately train the high-level viability and CoT modules in Section IV-C.

### A. Low-Level Policies

As discussed in Sec III-A, we assume that the robot is equipped with a repertoire of $K$ locomotion skills without making any assumption about how these low-level policies are gathered, whether through DRL, imitation learning, or optimal control approaches. In this work, we demonstrated our work with multiple policies $\pi_k$ trained with DRL in the Legged Gym simulator [21], which enables efficient training via multi-environment parallelization. For policy optimization, we adopt Proximal Policy Optimization (PPO) [46], a widely used DRL algorithm. We focus on learning policies for three distinct locomotion skills: (1) walking on flat terrain, (2) ascending stairs, and (3) descending stairs. Further details of VOCALoco's training setup are delineated in Section V-A.

### B. Data Collection

The data is collected by running various scenarios in simulation using Isaac Gym [47], to build a dataset of heightfields paired with their corresponding geometric viability and CoT for each low-level policy. We generate environments across three terrain types: (1) flat and uneven terrain, (2) ascending staircases, and (3) descending staircases. Figure 3.2, 3.3, and 3.4 show an example of the terrains we used to collect data. For each terrain type, we vary the difficulty level to generate extra environments (e.g., obstacle size, step height, etc).

We randomly spawn the robot in simulation and align the robot's roll and pitch with the terrain underneath it. Since the terrain and the initial pose of the robot are random, some combinations of initial positions may be infeasible to stand on (e.g., starting the robot at the edge of the staircase). We verify that the spawn position is valid by waiting for 1 second and checking if the robot's base is close to the desired spawn position, the robot's yaw is aligned with the desired yaw, and no base collisions are detected. Otherwise, another initial pose is generated.

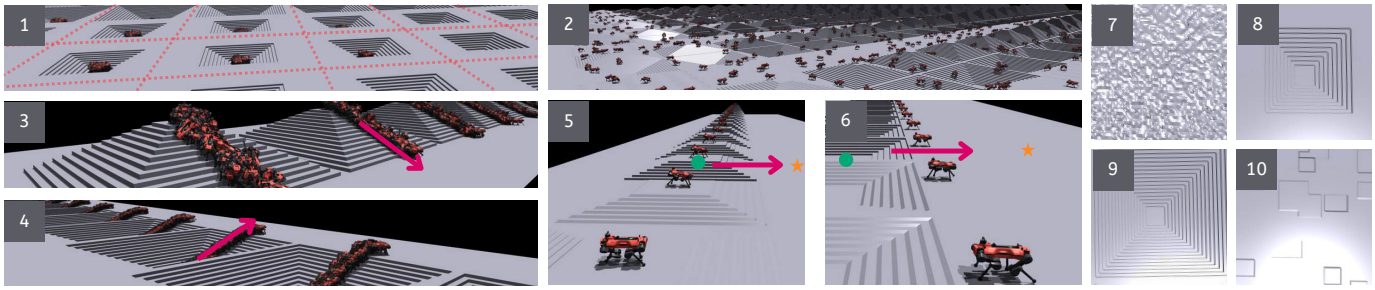


Fig. 3: Simulation terrains. (1) An example of a low-level locomotion policy training environment. The red dots denote the boundaries of the terrain cell that each robot cannot cross. (2) Policy rollouts on different terrain types to gather data for the high-level modules. (3) Descending staircase environment. (4) Ascending staircase environment. (5, 6) Evaluation environments. The terrains with transitions from flat to stairs and vice versa. The green dot represents the spawn position of the robot, and the star represents the target. (7) Rough terrain. (8) Stairs up. (9) Stairs down. (10) Discrete terrain.

*1) Collecting Viability Data:* If the robot has a valid initial pose, we take the local heightfield of the environment. For each valid initial pose, we rollout each policy on the same initial pose by moving the robot over a fixed horizon distance using a fixed velocity of $0.6m/s$. We choose $0.6m/s$ because on the real ANYmal-D, using a faster velocity could potentially lead to more noisy heightfields [21]. If the robot successfully reaches the target and no base collisions are detected, we label that data point as a success. If the robot did not reach the target with a time limit of 4 seconds, which could happen due to a collision or a terrain structure that steered the robot off its path, we label the data point as a failure. The same procedure is repeated for $N$ rollouts to obtain the success rate $(\frac{\text{number of successes}}{N})$.

For each sample, each value in the heightfield in simulation is injected with noise coming from a uniform distribution from -10cm to 10cm, which are the default noise values from [21]. Furthermore, we normalize the heightfield by subtracting the height value at its center from all other values. Specifically, if the center value is $x$, and a given heightfield value is $h$, we compute the normalized value as $h' = h - x$. This centers the heightfield around the robot's local elevation.

*2) Horizon distance:* While shorter rollout distances could better capture proximal terrain features to the robot, we set the rollout distance to $1.5m$ to avoid excessively frequent policy switching that can destabilize skill selection. Empirically, $1.5m$ also produces viability and CoT measurements with acceptable noise levels.

*3) Collecting CoT Data:* We follow a similar procedure to collect CoT data, where we rollout the policy and calculate its CoT using Equation 1. However, unlike in the viability data collection case, we set $N = 1$. Specifically, for a sampled heightfield, we do 1 rollout only since we are not measuring the success rate. In addition, the measured CoT values can be noisy. When a robot switches from a spawn state to a rollout state, it takes a maximum of $0.5s$ for the robot to accelerate to $0.6m/s$, depending on the low-level policy running on the robot, the spawn pose, and the spawn location. Therefore, to reduce noise in the data, we start measuring the CoT after the first $0.5s$ of the rollout. The rollout distance remains at $1.5m$. Any rollout that results in a base crash is marked invalid and excluded from the dataset.

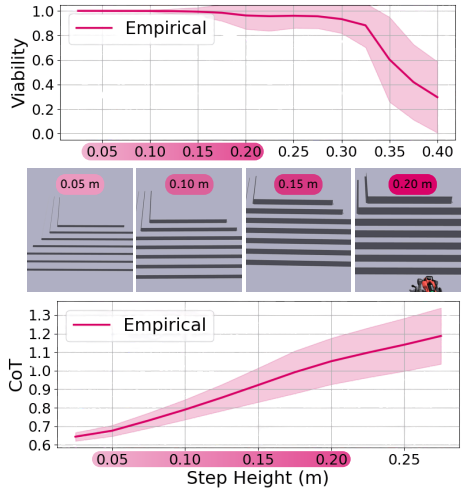


Fig. 4: Visualization of (middle) simulated environments with various difficulty levels and the collected (top) viability and (bottom) CoT data from rolling out the descent policy across different difficulty levels. The shaded region represents the standard deviation.

Figure 4 illustrates a representative example of the collected data. Terrains with different difficulty levels (step height) are generated (middle). The viability (top) and CoT (bottom) are collected by rolling out a policy in these terrains. As terrain difficulty increases, viability decreases while the cost of transport increases, which are both expected.

The cutoff of the CoT (bottom plot) was determined empirically by excluding step heights whose CoT datasets contained over 10% outliers (CoT < 0.25 or > 2.5), which resulted in a threshold of $0.275m$. The outlier cutoffs, 0.25 and 2.5, were determined empirically: for a rollout of $1.5m$, a poor spawn leading to an early crash would yield a CoT value below 0.25, and getting stuck or stumbling would yield a CoT above 2.5.

### C. Learning Viability and Cost-of-Transport

Given the data collected from Section IV-B, we aim to learn a mapping that predicts viability and CoT given a heightfield.

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

We take the policy viability as a regression problem on heightfield data. For each policy $\pi_k$ in $\Pi$, we aim to learn a predictive function that estimates the viability $\mathcal{V}^k(\mathbf{H})$.

Each $\mathcal{V}^k(\mathbf{H})$ takes a heightfield $\mathbf{H} \in \mathbb{R}^{w \times h}$ and regresses to a value between 0 and 1, where the higher the estimate, the more likely that traversing the local terrain of the robot is viable under a policy $\pi_k$: $\mathcal{V}^k(\mathbf{H}) : \mathbb{R}^{w \times h} \to [0, 1]$. For each policy, we also learn a predictive function that estimates the CoT: $\mathcal{C}^k(\mathbf{H}) : \mathbb{R}^{w \times h} \to \mathbb{R}^+$.

### D. Skill Selection

Our high-level component selects the optimal low-level policy given the robot's local terrain. It takes the heightfield $\mathbf{H}$ as input and predicts, for each policy $\pi_k$, its viability $\mathcal{V}^k(\mathbf{H})$ and CoT $\mathcal{C}^k(\mathbf{H})$. Policies with a viability lower than a predefined threshold $\mathcal{V}^k(\mathbf{H}) < \epsilon_v$ are discarded as unsafe. Among the remaining candidates, the module would select the policy with the lowest predicted CoT. Unlike prior works using CoT implicitly as a low-level reward term during training [21] [29] [30] [10] [11], VOCALoco predicts CoT explicitly at runtime to compare multiple pretrained skills and select the most energy-efficient one.

## V. EXPERIMENTS

### A. Experiment Setup

We begin with three locomotion tasks, walking, ascending stairs, and descending stairs. For this, 3 low-level policies are trained, together with their corresponding viability and CoT prediction model, resulting in 6 CNNs in total.

*1) Training Terrain-Specialized Locomotion Policies:* As discussed in Section IV, each policy is trained primarily on its specialized terrain type (e.g., stairs for climbing policies) with a smaller portion of uneven and discrete terrains included to promote more natural gaits. The rest of the environment settings are inherited from [21], including the curriculum learning, observation space noise, and the action space. Reward functions are tuned to obtain terrain-specific policies.[1]

*2) Heightfield:* The robot processes depth camera images into elevation maps [48], from which we query to produce heightfields. One issue is that occlusions in the heightfield could potentially cause incorrect predictions. This issue becomes especially problematic when the robot is ascending stairs and has no knowledge of the terrain structure beyond the top of the stairs, potentially causing the viability CNNs to return a conservative estimate. To address this, we perform a forward fill along each column of the heightfield, replacing occluded cells (NaN) with the last valid value observed behind them. This operation proceeds from the back of the robot toward the front, allowing us to propagate known height values forward into occluded regions.

*3) Implementation Details:* Each CNN in both the viability and CoT modules comprises 3 convolutional layers with 4, 8, and 8 channels, respectively. All convolutional layers use a 3×3 kernel, stride 1, and padding 1. A MaxPool layer (kernel 2) downsamples between the second and third convolutional layers. After these 3 convolutions, the feature map is passed through 2 fully connected layers of 128 neurons each before reaching a single output head. ReLU activations are applied to all hidden layers, but only the Viability CNN applies a sigmoid at the last logit. During training, we use mean-squared error loss and stochastic gradient descent with a learning rate of 0.005 and momentum of 0.9. The dataset for each CNN has 100k samples.

[1]Details are available at: https://sites.google.com/view/vocaloco.

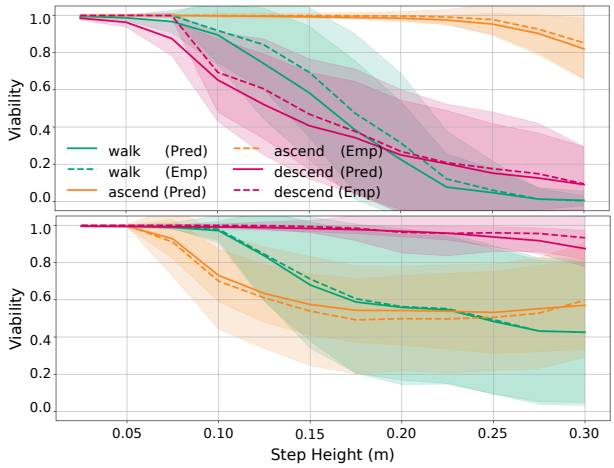


Fig. 5: The collected and the predicted viability across terrains with different step heights for (Top) ascending staircase environment (Figure 3.3) and (Bottom) descending staircase environment (Figure 3.4).

Our experiments were carried out on an NVIDIA 4090 GPU with VRAM of 24GB. In our case, collecting 100K viability and CoT samples takes around 4 hours and 12 minutes, respectively. Training the CNNs with 100K samples also takes around 12 minutes. We can parallelize the tasks, taking a total of 8-10 hours for data collection and training.

We set the viability thresholds to $\epsilon_{v,\,\text{walk}} = 0.95$, $\epsilon_{v,\,\text{ascend}} = 0.925$, and $\epsilon_{v,\,\text{descend}} = 0.925$, both in simulation and on the real robot. Selecting these values trades off risk-averse versus risk-seeking behavior in our viability CNN: higher thresholds yield more conservative behavior, leading to more false negatives (safe terrain flagged as unviable), whereas lower thresholds produce more aggressive behavior, resulting in more false positives (unsafe terrain flagged as viable). The chosen values strike a balance between avoiding unnecessary stops and preventing collisions or stalls.

We run the high-level modules at 50 Hz and we pass each predicted skill into a sliding window of length 10. When all skills in the window agree, we execute that skill on the robot. This consensus provides greater skill-switching stability. Moreover, the filtering smooths transitions: shorter windows lead to redundant switching, while longer windows introduce lag. Empirically, we found that using a window size of 10 enables timely policy transitions while avoiding redundant toggling.

### B. High-Level Module Performances

We start our analysis by validating the performance of the learned viability modules. Figure 5 shows the collected data

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

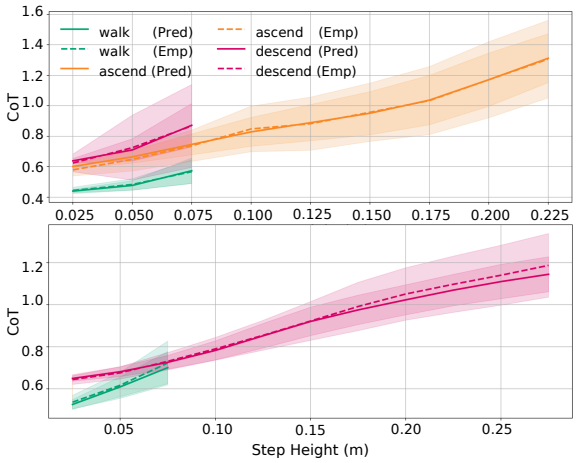


Fig. 6: The collected and predicted CoT across terrains with different step heights for (Top) ascending staircase environment (Figure 3.3) and (Bottom) descending staircase environment (Figure 3.4). We only plot data points if the policy is viable during rollouts (with threshold = 90%).

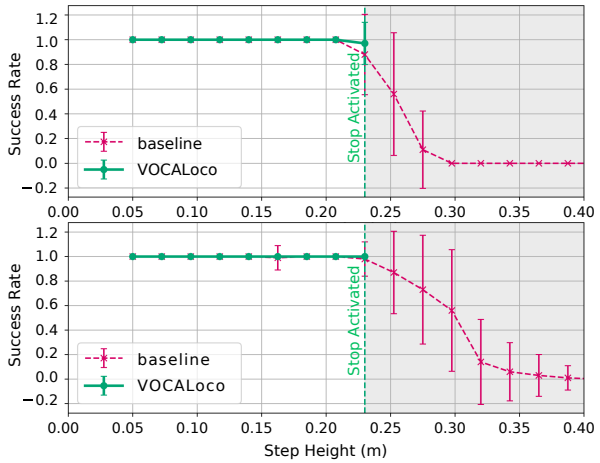


Fig. 7: Success rate of executing policies on terrains from (top) Figure 3.5 and (bottom) Figure 3.6. We perform 100 rollouts per step height and plot the success rate across increasing difficulty against the baseline [21]. The error bars represent the standard deviation. *Stop Activated* indicates that our viability module has determined the terrain is unviable and prevented the robot from moving.

and the predicted viability across terrains of increasing difficulty on ascending (Figure 3.3) and descending staircase environments (Figure 3.4). For each step height (incremented by $2.5cm$), we collect a test set of 500 samples.

For all policies, we observe that as the step heights increase, the viability decreases. While this result is expected, it is crucial to verify that our viability CNNs are well-calibrated on the terrain structures of interest. Accurate calibration allows us to set custom thresholds $\epsilon_v$ for each policy, thereby reliably determining whether a terrain segment is safe or unsafe to traverse.

Similarly, Figure 6 shows the predicted CoT and the variance over 500 testing data. First, we observe the expected general trend of increasing CoT as the step heights increase. We further note that the walking policy incurs a lower CoT than the ascend and descend policies on very low step heights. This observation is consistent with expectations, as the walking policy generally generates lower swing trajectories, resulting in reduced energy expenditure relative to the other policies. Note that, we only plot data points if the policy is (up to 90%) viable during rollouts. The outcome of the ascend policy on the descending staircase environment is omitted, because every test set at every height contained more than 10% of crashes.

Note that, despite fixing the step height for each test set, the data will still be diverse since the initial position and orientation of the base are randomly generated, and each test may involve terrain transitions from flat to stairs and/or from stairs to flat terrain. Therefore, there is a variance across most subplots in Figures 5 and 6. The variance of the test set and the predictions also increases as the step heights increase. This is also expected, since more difficult terrain leads to greater low-level policy instability and overall unpredictability in the robot's behaviour, especially when operating in an environment outside its training distribution.

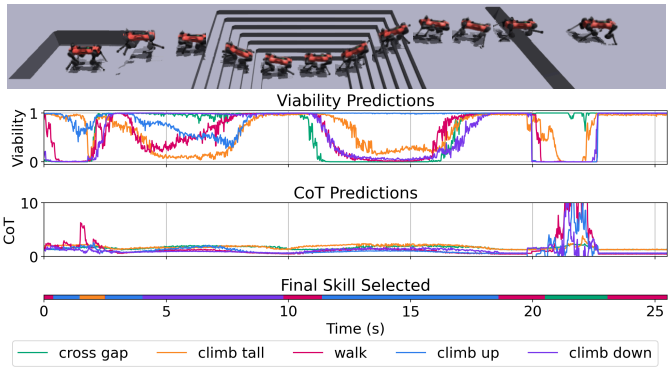


Fig. 8: Predicted viability, CoT, and final skill to execute on the ANYmal robot on an obstacle course in simulation.

### C. Performance in Simulation

We want to evaluate the performance of VOCALoco against the baseline, a rough terrain ANYmal-D locomotion policy using [21] in simulation. Our testing environments are shown in Figures 3.5 where the robot spawns on a flat terrain, then ascends stairs, then resumes on flat terrain and Figures 3.6 where the robot spawns on a flat terrain, descends stairs, then resumes walking on flat terrain. Success corresponds to reaching the target, depicted by a star in both figures.

Figure 7 illustrates success rates for scenarios using [21] as a baseline. *Stop Activated* indicates that our viability module has determined the terrain to be unviable, thereby preventing the robot from proceeding. We observe that VOCALoco identifies unviable terrain and halts, whereas the baseline, lacking any untraversability or risk detection, continues tracking the desired velocity and ultimately becomes stuck or crashes on higher-step stairs during both ascent and descent.

We note that our low-level policies are more robust on

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

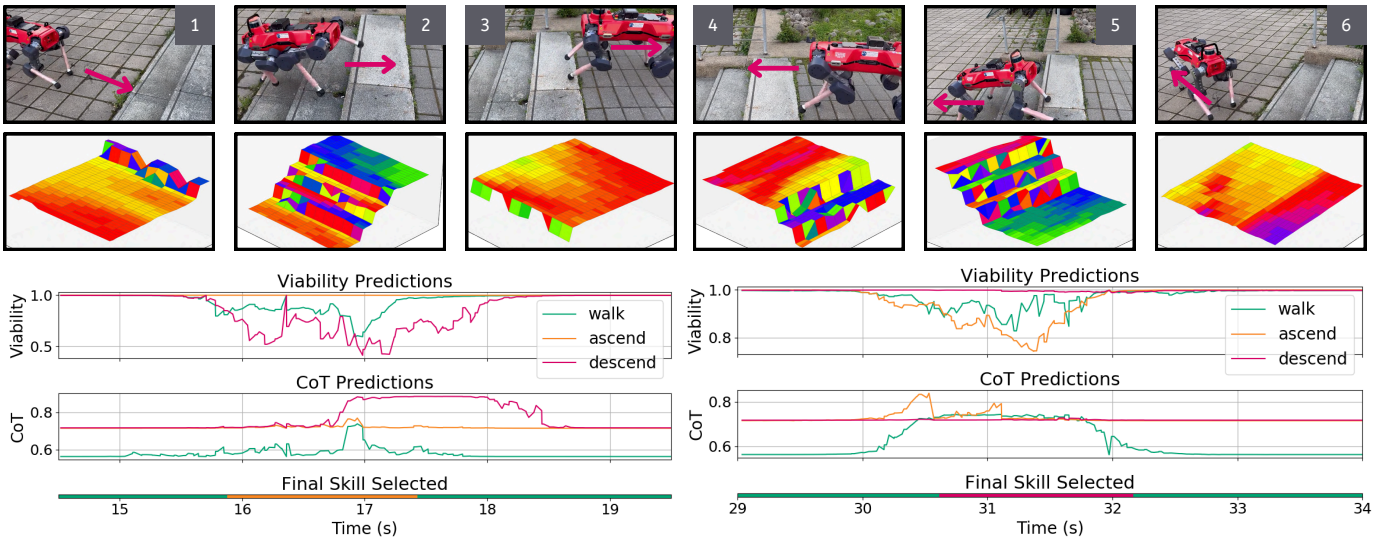


Fig. 9: Real-world deployment of ANYmal-D on a staircase with three $12cm$ steps. Snapshots 1-3 show ascent and 4-6 show descent. The rows represent (top) raw frames, (middle) extracted heightfields, and (bottom) viability and CoT predictions. Viability remains above threshold at each policy switch, and CoT curves correctly identify the most efficient policy, confirming smooth, reliable transitions.

terrains with higher steps than the baseline policy. Looking at the step height of $22.5cm$ for both subplots of Figure 7, we either observe fewer crashes or achieve a 100% success rate at reaching the target.

### D. Modular and Scalable Skill Integration

To demonstrate that our proposed framework is modular and scalable, we extend our framework to two new skills: climbing tall obstacles and crossing gaps. We start off with 3 existing skills, along with their associated viability and CoT networks. Then, as the first step in the extension, we train two new policies, climbing tall obstacles and gap-crossing, with the difference that they take position-based target commands [6] [7]. Once these skills are trained, we proceed to collect viability and CoT data and train their CNN modules. *Without retraining any of the existing CNNs*, we show that direct integration of the new skills and CNNs with the existing framework allows the robot to traverse an obstacle course with a wall and a gap of 0.5 meters (see Figure 8).

### E. Real World Deployment

We validate VOCALoco on real hardware using ANYbotics ANYmal-D [15]. At all times, a joystick provides linear velocity commands of between $0.5$ and $0.6m/s$. As in simulation, both the high-level and low-level policies operate at $50$ Hz, and no parameters are altered between the simulated and real-robot setups.

Figure 9 shows the output viability and CoT module outputs during deployment on a staircase with 3 steps all around $12cm$. Initially, all policies are equally viable, so the policy with a lower CoT is selected, corresponding to the walk policy. As the robot moves toward the staircase, it switches to the ascend policy because the viability module estimates that it is the only viable policy. Once the base of the robot is back on flat terrain, it resumes to the walking policy, as it has the lowest predicted CoT. We note that this switch happens while the hind legs are still on the last step of the stairs. In any of VOCALoco runs, however, this does not cause the robot to get stuck, as the walking policy is able to lift its hind legs to clear the last step. Similarly, we can see the descent policy is selected while the robot is descending the stairs, and is using the walking policy on flat terrain. We also do not observe any issues during the transitions between locomotion policies. For more hardware experiment outcomes on other staircases with different heights, please refer to the supplemental video available at: https://sites.google.com/view/vocaloco.

## VI. CONCLUSION

We present a novel approach to switch between terrain-specialized locomotion policies to enhance overall safety and efficiency. Our system combines high-level decision modules, a viability module and a CoT module, with low-level control policies. Each module leverages CNNs to assess the terrain for a fixed horizon. The viability module identifies which policies can safely traverse it, and then, the CoT module selects the most energy-efficient policy. We validated our approach in simulation and successfully deployed it zero-shot on ANYmal-D quadruped. As future work, we plan to extend our framework from discrete skill switching to policy mixing, enabling smoother and more continuous transitions between skills, for example, using a multi-expert composition approach.

## REFERENCES

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

[2] 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.

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

[4] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," in Conference on Robot Learning, 2023.

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

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

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

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

[9] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in Conference on Robot Learning, 2022, pp. 773–783.

[10] Z. Fu, A. Kumar, J. Malik, and D. Pathak, "Minimizing energy consumption leads to the emergence of gaits in legged robots," in Conference on Robot Learning, 2021.

[11] B. Liang, L. Sun, X. Zhu, B. Zhang, Z. Xiong, Y. Wang, C. Li, K. Sreenath, and M. Tomizuka, "Adaptive energy regularization for autonomous gait transition and energy-efficient quadruped locomotion," in 2025 IEEE International Conference on Robotics and Automation, 2025, pp. 5350–5356.

[12] G. Bellegarda and A. Ijspeert, "Cpg-rl: Learning central pattern generators for quadruped locomotion," IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 12 547–12 554, 2022.

[13] G. Bellegarda, M. Shafiee, and A. Ijspeert, "Allgaits: Learning all quadruped gaits and transitions," IEEE International Conference on Robotics and Automation, pp. 15 929–15 935, 2025.

[14] M. Shafiee, G. Bellegarda, and A. Ijspeert, "Viability leads to the emergence of gait transitions in learning agile quadrupedal locomotion on challenging terrains," Nature Communications, vol. 15, p. 3073, 2024.

[15] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2016, pp. 38–44.

[16] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in IEEE international conference on robotics and automation, 2018, pp. 3803–3810.

[17] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," in Robotics: Science and Systems, 2021.

[18] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in Conference on robot learning, 2020, pp. 66–75.

[19] 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.

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

[21] 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, 2021.

[22] 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.

[23] R. Yang, G. Yang, and X. Wang, "Neural volumetric memory for visual locomotion control," in Conference on Computer Vision and Pattern Recognition, 2023.

[24] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," in International Conference on Learning Representations, 2022.

[25] G. Bellegarda, M. Shafiee, and A. Ijspeert, "Visual cpg-rl: Learning central pattern generators for visually-guided quadruped locomotion," in IEEE International Conference on Robotics and Automation, 2024, pp. 1420–1427.

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

[27] 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.

[28] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control," IEEE Transactions on Robotics, vol. 38, no. 5, pp. 2908–2927, 2022.

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

[30] I. Nahrendra, B. Yu, M. Oh, D. Lee, S. Lee, H. Lee, H. Lim, and H. Myung, "Obstacle-aware quadrupedal locomotion with resilient multi-modal reinforcement learning," arXiv preprint arXiv:2409.19709, 2024.

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

[32] S. Kareer, N. Yokoyama, D. Batra, S. Ha, and J. Truong, "ViNL: Visual Navigation and Locomotion Over Obstacles," in IEEE International Conference on Robotics and Automation, 2023.

[33] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, "Learning ground traversability from simulations," in IEEE International Conference on Robotics and Automation, 2018.

[34] B. Yang, L. Wellhausen, T. Miki, M. Liu, and M. Hutter, "Real-time optimal navigation planning using learned motion costs," in IEEE International Conference on Robotics and Automation, 2021, pp. 9283–9289.

[35] J. Frey, D. Hoeller, S. Khattak, and M. Hutter, "Locomotion policy guided traversability learning using volumetric representations of complex environments," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2022, pp. 5722–5729.

[36] F. Muhamad, J.-S. Kim, and J.-H. Park, "Robust traversability prediction using multiple costs for quadruped robot in random terrains," IEEE Access, 2024.

[37] S. Chamorro, V. Klemm, M. d. L. I. Valls, C. Pal, and R. Siegwart, "Reinforcement learning for blind stair climbing with legged and wheeled-legged robots," in IEEE International Conference on Robotics and Automation, 2024, pp. 8081–8087.

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

[39] S. Zenker, E. E. Aksoy, D. Goldschmidt, F. Wörgötter, and P. Manoonpong, "Visual terrain classification for selecting energy efficient gaits of a hexapod robot," in IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2013, pp. 577–584.

[40] M. Elnoor, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "Pronav: Proprioceptive traversability estimation for legged robot navigation in outdoor environments," IEEE Robotics and Automation Letters, 2024.

[41] D. Belter, J. Bednarek, H.-C. Lin, G. Xin, and M. Mistry, "Single-shot foothold selection and constraint evaluation for quadruped locomotion," in IEEE International Conference on Robotics and Automation, 2019, pp. 7441–7447.

[42] F. Jenelten, R. Grandia, F. Farshidian, and M. Hutter, "Tamols: Terrain-aware motion optimization for legged systems," IEEE Transactions on Robotics, vol. 38, no. 6, pp. 3395–3413, 2022.

[43] D. J. Farris and G. S. Sawicki, "The mechanics and energetics of human walking and running: a joint level perspective," Journal of The Royal Society Interface, vol. 9, no. 66, pp. 110–118, 2012.

[44] E. M. Summerside, R. Kram, and A. A. Ahmed, "Contributions of metabolic and temporal costs to human gait selection," Journal of The Royal Society Interface, vol. 15, no. 143, pp. 180–197, 2018.

[45] M. Srinivasan and A. Ruina, "Computer optimization of a minimal biped model discovers walking and running," Nature, vol. 439, no. 7072, pp. 72–75, 2006.

[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[47] 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 Neural Information Processing Systems, 2021.

[48] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in Mobile Service Robotics, 2014, pp. 433–440.