Training and Simulation of Quadrupedal Robot in Adaptive Stair Climbing and Descending for Indoor Firefighting: An End-to-End Reinforcement Learning Approach

Baixiao Huang[1⋆] Baiyu Huang[1⋆] and Yu Hou[2,3†]

[1]Independent Researcher
[2]Department of Construction Management, Western New England University
[3]Department of Engineering Technology and Construction Management, University of North Carolina at Charlotte
⋆These authors contribute equally to this work †Corresponding author

baixiaohuang1@gmail.com, baiyuhuang2@gmail.com, yu.hou@wne.edu, yu.hou@charlotte.edu

Abstract -

Quadruped robots are used for primary searches during the early stages of indoor fires. A typical primary search involves quickly and thoroughly looking for victims under hazardous conditions and monitoring flammable materials. However, situational awareness in complex indoor environments and rapid stair climbing and descending across different staircases remain the main challenges for robot-assisted primary searches. In this project, we designed a two-stage end-to-end deep reinforcement learning (RL) approach to optimize both navigation and locomotion. In the first stage, the quadrupeds, *Unitree Go2*, were trained to climb and descend stairs in *Isaac Lab*'s pyramid-stair terrain. In the second stage, the quadrupeds were trained to climb and descend various realistic indoor staircases in the *Isaac Lab* engine, with the learned policy transferred from the previous stage. These indoor staircases are straight, L-shaped, and spiral, to support climbing and descending tasks in complex environments. This project explores how to balance navigation and locomotion and how end-to-end RL methods can enable quadrupeds to adapt to different stair shapes. Our main contributions are: (1) A two-stage end-to-end RL framework that transfers climbing/descending skills from abstract pyramid terrain to realistic indoor stair topologies. (2) A centerline-based navigation formulation that enables unified learning of navigation and locomotion without hierarchical planning. (3) Demonstration of policy generalization across diverse staircases using only local height-map perception. (4) An empirical analysis of success, efficiency, and failure modes under increasing stair difficulty.

Keywords -

Robot Dogs; Quadrupedal Robots; Stair Climbing/Descending; Isaac Lab; Fire Primary Search

1 Introduction

During indoor fires, the primary search phase involves a rapid, meticulous survey of the environment to locate potential victims, often under hazardous conditions. This critical task also involves continuous monitoring for highly flammable and combustible materials that could pose an immediate threat to first responders. With the development of artificial intelligence (AI) and advanced robotics, quadrupedal robots, commonly referred to as robot dogs, have become a vital tool for primary search tasks. These quadrupeds possess several distinct advantages that make them superior to other robotic platforms in this specialized application. Their cost-effectiveness is a significant factor, making them more accessible for deployment by various emergency services. Moreover, their superior adaptability allows them to traverse uneven terrain and navigate cluttered indoor spaces that would immobilize wheeled or tracked robots. Crucially, their inherent stability, derived from their four-legged gait, provides a robust platform for sensor deployment and reliable movement across debris-strewn floors. Finally, researchers also use indoor drones for primary search, but they cannot handle obstacles as effectively as quadrupeds. For example, mounting manipulator arms on robot dogs allows them to open doors or clear barriers during primary search.

Despite these strengths, the use of robot dogs in primary searches is still constrained by two major technical hurdles. The first is achieving comprehensive situational awareness within the complex and dynamic indoor fire environment. This challenge involves overcoming visual obstructions caused by smoke, accurately mapping three-dimensional space, and interpreting sensor data (such as thermal readings) to build a clear, actionable picture of the environment for human commanders. Researchers are studying the application of building information modeling (BIM) to support robot dogs' situational awareness. The second challenge is developing robust, high-speed stair-climbing and descending capabilities. Given the architectural variety of buildings, the robots must be able to rapidly and reliably ascend and descend a multitude of staircase types. These staircase types include regular, spiral, and damaged steps to facilitate a rapid and compre-

hensive primary search across all floors. Overcoming this challenge is essential for maximizing the effectiveness of robot-assisted primary searches.

In this project, we explored an innovative approach to solving the quadruped's stair-climbing/descending challenges. This approach presents a two-stage end-to-end deep reinforcement learning (RL) framework to optimize both navigation and locomotion. We trained and tested our algorithm with the NVIDIA *Isaac Lab* platform and implemented our approach on a widely used quadruped robot prototype, *Unitree Go2*. In the first stage, the quadruped was trained to climb/descend stairs in *Isaac Lab*'s pyramid-stair terrain. In the second stage, dogs were trained to climb/descend various realistic indoor staircases in the *Isaac Lab* engine, using the policy learned in the previous stage. There are many types of indoor staircases in residential and commercial houses. In order to test our approach, we conducted experiments on three typical indoor staircases, including straight, L-shaped, and spiral stairs, to support climbing/descending tasks in complex environments.

This project explores how to balance navigation and locomotion and how end-to-end RL methods can enable the robot dogs to adapt to different stair shapes while maintaining high agility.

2 Related work

2.1 Robots for Indoor Primary Search

In Firefighting, primary search requires an initial and rapid search to locate and remove victims quickly and safely while the fire is still active and conditions remain dangerous. Researchers have implemented robots for primary search. Gelfert et al. [1] and Talavera et al. [2] developed a mobile robot concept to assist firefighters in locating victims in smoky indoor apartment fires using an Unmanned Ground Vehicle (UGV), but their UGVs are tracked-based and consider only one floor in a fire scenario. Pritzl et al. [3] and Han et al. [4] used an Unmanned Aerial Vehicle (UAV) to autonomously enter a target building through an open window in a firefighting scenario. However, UAVs cannot handle obstacles and can only access fire zones through openings, such as open doors and windows. Based on this discussion, quadruped robots offer a broader range of indoor firefighting use cases than existing UGVs and UAVs.

2.2 Quadruped Robot Prototypes and Training Platforms

There are numerous representative platforms and reference quadruped robot prototypes, including those from Boston Dynamics [5], Unitree Robotics [6], Ghost Robotics, and ANYbotics/ANYmal. They have been used for building-environment inspection, disaster response, search and rescue, and military reconnaissance. NVIDIA *Isaac Lab* is a widely used open-source robotics learning platform for developing RL algorithms and deploying Sim-to-Real workflows.

2.3 Quadruped Robot Climbing Tasks

Qi et al. [7] and Liang et al. [8] used a model-based framework to explore the stair-climbing algorithms with perception, planning, and control structures. However, the stair environments are simplified to straight shapes. Vogel et al. [9] explored ladder climbing (equivalent to a 90° stair) using a custom hook-shaped end effector, but their method cannot be generalized to other staircases due to its reliance on specialized hardware. Hoeller et al. [10] utilized ANYmal to perform agile navigation in a complex environment. They decomposed the problem into perception, locomotion, and navigation modules. In the locomotion module, the trained quadrupedal robot walks, jumps, and climbs. However, their testing environments are static. Kim et al. [11] proposed a hierarchical navigation framework with a low-level tracker controller and a high-level foothold planner. The framework enables quadrupedal robots to traverse discrete terrain at high speed. The hierarchical framework could potentially increase system complexity and exclude globally optimal behaviors.

In summary, future quadruped robot climbing tasks need to be investigated to develop a general end-to-end framework for various stair shapes to address the complex staircases. The end-to-end methods are simple to design, fast to execute, and naturally handle perception-control feedback. Additionally, training robots to be familiar with various stair shapes helps them conduct indoor fire primary search effectively.

3 Methodology

We trained a neural network policy to perform both the navigation and locomotion tasks on the stairs terrain. The policy takes proprioceptive states, environmental states represented as a height map, and user-commanded goals as inputs/observations. It then outputs joint position commands as actions, and robot dogs' Proportional-Derivative (PD) controllers convert joint position commands into torques applied to joint motors.

The neural network is trained by using deep RL with the on-policy algorithm Proximal Policy Optimization (PPO) [12] in the *Isaac Lab* environment. The training framework consists of two stages. The first stage trains the policy on stairs terrain to acquire basic stair-climbing and descending locomotion skills. As shown in Figure 1, the terrain, preprogrammed by *Isaac Lab*, is a pyramid-stair pattern that tapers to a flat platform at its center. The robot

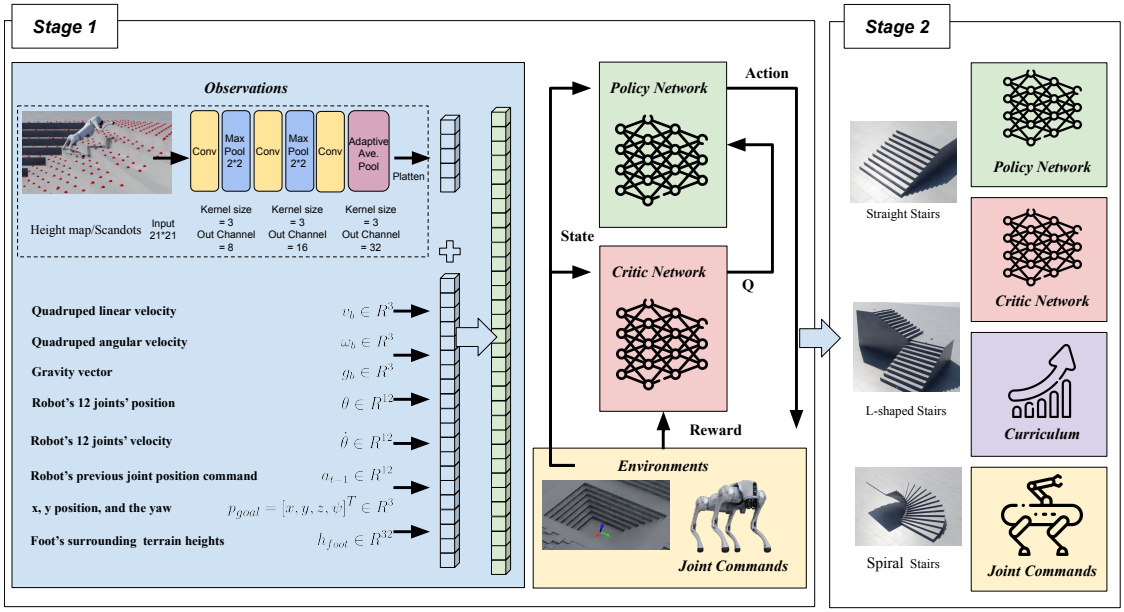


Figure 1. Two-Stage training

is placed at the center of the staircase and tasked with navigating to a specific goal position and orientation. The second stage trains the policy on our various customized stair terrains to develop terrain-specific navigation skills in addition to basic locomotion skills. In the second stage, the robot is tasked with tracking the stair centerline.

3.1 Observations

The inputs, or observations, to the neural network are defined as $o = \left[v_b, \omega_b, g_b, \theta, \dot{\theta}, a_{t-1}, p_{goal}, e\right]$, where $v_b \in \mathbb{R}^3$ and $\omega_b \in \mathbb{R}^3$ are the quadruped's linear and angular velocity in the body frame, and $g_b \in \mathbb{R}^3$ is the gravity vector in the body frame. In the observation equation, $\theta \in \mathbb{R}^{12}$ and $\dot{\theta} \in \mathbb{R}^{12}$ are the position and velocity of 12 joints, $a_{t-1} \in \mathbb{R}^{12}$ is the 12 joint position command in the previous timestep passed to the PD controller, and $p_{goal} = [x, y, z, \psi]^T \in \mathbb{R}^4$ is the goal pose consisting of x, y position, and the yaw of the goal pose. Lastly, $e = \left[h_{foot}, h_{grid}\right]^T$ is an environmental state containing the heights of the terrain around the quadruped's 4 feet with a space of $h_{foot} \in \mathbb{R}^{32}$ and the heights of the terrain centering around the quadruped in a grid-like pattern with a space of $h_{grid} \in \mathbb{R}^{441}$. The grid-like pattern has a 21x21 shape with a 0.2m resolution. These observations provide the policy with robot kinematic and proprioceptive states, as well as the terrain around the robot, which are necessary for the policy to produce navigation and locomotion strategies. Finally, the policy outputs joint action for the controller $a_t = f_{policy}(o)$, where f_{policy} is the deep neural network, and $a_t \in \mathbb{R}^3$ is the joint position setpoint.

3.2 Network architecture

The neural network f_{policy} consists of a shallow convolutional neural network (CNN) encoder and a 3-layer multilayer perceptron (MLP). The CNN takes a 21-by-21 height map/ scandots as input and outputs a 128-dimensional feature vector representing terrain information to the MLP layer, along with other observations as described previously. The MLP then outputs the action, or 12 joint position command. The CNN consists of convolutional and pooling layers, while the MLP consists of 3 hidden layers with sizes 128, 128, and 64.

The critic network has the same architecture as the policy network and shares the CNN encoder with it. The policy CNN encoder's output is fed into the critic MLP network. By sharing the CNN encoder, we remove the burden of training a separate encoder for terrain features. However, the critic's MLP has its own set of weights, which estimate long-term returns given the current robot and environmental states.

3.3 Rewards and Penalties

In this paper, we formulated the problem as a navigation task, rather than a velocity-tracking problem because the robot may need to move at different velocities depending on the terrain. We divided all rewards into two groups based on their functions: task/navigation rewards and regularization rewards. The total rewards are thereby $r_{total} = r_{task} + \sum_{i=1}^{n} \lambda_i r_{reg}^{(i)}$, where λ_i is the weight, and $r_{reg}^{(i)}$ represents the regularization reward and penalties.

The reward and penalty weights for two stages are summarized in Table 1.

3.4 Task rewards

In stage 1 training, we used a navigation reward to indicate how close the quadruped was to the goal. It is formulated as in Equation (1).

$$r_{nav} = 1 - tanh(\frac{d_{goal}}{\sigma_{nav}}) \tag{1}$$

where r_{nav} is the navigation reward, the d_{goal} is the Euclidean distance from the robot to the goal, and σ_{nav} is the range in which the reward takes effect. The larger the σ_{nav} is, the further the influence of the reward is. The purpose of the $tanh$ function is to reshape the reward in the range of 0 and 1, while keeping the gradient of the reward smooth. We used two navigation rewards, the coarse navigation reward, $r_{nav,coarse}$, and the fine navigation rewards, $r_{nav,fine}$, to handle both long-distance and close-to-goal tracking. The σ_{nav} values are 5.0 and 1.0, respectively, based on our training experience.

In stage 2 training, the robot was presented with a different task, navigating up the stairs. Therefore, the coarse navigation reward was replaced with two rewards, a centering reward and a path reward. A centerline is the path situated in the middle of the stairs and running along them. We required the quadruped to stay close to this centerline and traverse it to reach the goal. This naturally leads to the centering reward and path reward, represented as Equation (2) and Equation (3).

$$r_{center} = 1 - tanh(\frac{d_{center}}{\sigma_{center}}) \tag{2}$$

$$r_{path} = (1 - tanh(\frac{d_{path}}{\sigma_{path}})) \times (1 - tanh(\frac{d_{center}}{\sigma_{center}})) \tag{3}$$

In Equations (2) and (3), r_{center} and r_{path} are the centering and path rewards, d_{center} is the distance from the quadruped to the center line; d_{path} is the distance to the goal along the centerline; σ_{center} and σ_{path} determine the distance of influence similar to stage 1 navigation reward. Note that the path reward is conditioned on distance to the centerline, so that the reward can only be gained by traversing near the centerline. These two rewards encourage the robot to travel in the center of the stairs while navigating to the goal.

To track heading, we constructed a heading tracking penalty simply as, $r_{tracking} = -f_{angle}(\psi, \psi_{goal})$, where ψ is the current quadruped heading or yaw, and ψ_{goal} is the goal heading. f_{angle} is a function that returns the angular distance between two angles. The negative sign indicates that this term is a penalty.

3.5 Regularization and penalties

In addition to task rewards, there are regularization rewards that limit the quadruped from extreme behavior, such as moving too fast, creating jerky movement, or colliding with walls. Therefore, we presented noteworthy regularization rewards..

$$r_{power} = -\sum_{i=1}^{12} \tau_i \dot{\theta}_i \tag{4}$$

$$r_{stationary\ power} = -\sum_{i=1}^{12} \tau_i^2 \tag{5}$$

$$r_{action\ rate} = -\|a_t - a_{t-1}\|_2^2 \tag{6}$$

$$r_{joint\ limit} = -\sum_{i=1}^{12} (ReLU(\theta_i - \theta_{max}) + ReLU(\theta_{min} - \theta_i)) \tag{7}$$

$$r_{joint\ velocity\ limit} = -\sum_{i=1}^{12} ReLU(\dot{\theta} - \dot{\theta}_{max}) \tag{8}$$

$$r_{joint\ velocity} = -\left\|\dot{\theta}\right\|_2^2 \tag{9}$$

$$r_{joint\ acceleration} = -\left\|\ddot{\theta}\right\|_2^2 \tag{10}$$

The equations 4- 10 above are joint-related regularization. r_{power} and $r_{stationary\ power}$ penalize motor power output and excessive torque. $r_{action\ rate}$ penalizes fast action change. $r_{joint\ limit}$ and $r_{joint\ velocity\ limit}$ penalize reaching the physical limit of the motors, whereareas $r_{joint\ velocity}$ and $r_{joint\ acceleration}$ penalize excessive joint velocity and acceleration.

We penalized collisions between the quadruped and objects. Specifically, a penalty of -8.0 is added when the quadruped's body, head, hip, and thigh contact the ground or obstacles. An additional penalty of -0.2 is added when the calf is in contact, because mild contact with calf is acceptable.

To make the quadruped's gait more stable, we also penalized flying gait and encouraged larger steps through gait-related penalties and rewards. Denote $c_i(t) \in \{0, 1\}$ as the foot i contact state, which takes the value of 1 when in contact and 0 otherwise. The flying gait penalty and the step reward are $r_{flying} = \begin{cases} -1\ \forall i \in \{1,2,3,4\}\ c_i(t) = 0 \\ 0\ otherwise \end{cases}$ and $r_{step} = \sum_{i=1}^{4} 1 - c_i(t)$. Finally, we penalized excessive action when the robot is in the goal region using the same reward function as $r_{action\ rate}$.

3.6 Curriculum

It was shown that using a terrain-based curriculum can assist the learning [13]; therefore, we employed our customized curriculum for the training of stair climbing and descending. Each terrain type is divided into 10 levels, with each level progressively more difficult. In stage 1 training, pyramid stairs terrain increases its step height from 0 cm (flat terrain) to 12 cm. In stage 2 training, we increased the step height from 2cm to 12cm. In addition to step height, stair width also varies from 2.0 meters to

(a) Climb: STR. (b) Climb: L. (c) Climb: Spiral (d) Descend: STR. (e) Descend: L. (f) Descend: Spiral

Figure 2. Robot Dogs Climbing and Descending Various Staircases

Table 1. Reward and penalty weights for two stages.

Reward/Penalty	1st Stage	2nd Stage
$r_{nav,fine}$	1.0	1.0
$r_{nav,coarse}$	1.0	0
r_{center}	0	0.1
r_{path}	0	1.9
$r_{collision}$	-8.0	-8.0
$r_{tracking}$	-0.3	-0.3
r_{power}	-1×10^{-4}	-1×10^{-4}
$r_{stationary\ power}$	-1×10^{-5}	-1×10^{-5}
$r_{action\ rate}$	-1×10^{-4}	-1×10^{-4}
$r_{joint\ limit}$	-0.1	-0.1
$r_{joint\ velocity\ limit}$	-0.1	-0.1
$r_{joint\ velocity}$	-5×10^{-5}	-5×10^{-5}
$r_{joint\ acceleration}$	-2×10^{-8}	-2×10^{-8}
r_{flying}	-0.1	-0.1
r_{step}	0.2	0.2

1.4 meters. One important feature of the curriculum design is that the length of the stairs increases as difficulty increases, and we observed that this design significantly improves learning speed. Specifically, for L-shaped terrain, the length of stairs after the turn ranges from 0 to 3 meters, which allows the policy to gradually learn to turn around the corner. In spiral terrain, the spiral length grows from 0.2 to 0.5 revolution as difficulty increases. The rule of leveling up follows Rudin et al [14]. Given a quadruped at a specific level, it can level up to the more difficult terrain if it can reach the goal position with specified yaw. Otherwise, it gets demoted to a lower level if it fails to reach its goal. If it completes all levels, it is placed on a random level for continued training to prevent catastrophic forgetting of the policy.

4 Experiments and Evaluations

To test the performance of our resulting policy, we evaluated it on straight, L-shaped, and spiral stairs with parameters that differed from those in the training terrain. Given a single terrain type, we split the testing terrain into 6 levels, each with a different stair riser height ranging from 4cm to 14cm. The performance of the policy is measured by six metrics: **Goal-Reached Success Rate**, **Mean Linear Velocity**, **Average Climbing/Descending Rate**, **Position Error**, **Heading Error**, and **Mean Power Output**. For example, position error is the distance error between the quadruped position and the commanded position. This informs us how accurately the quadruped reaches its goal. Mean power output is produced by the 12 motors averaged over the whole testing episode. This indicates the policy's efficiency.

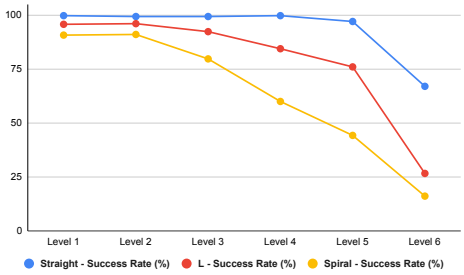
5 Results and Discussion

5.1 Model performance on different terrains

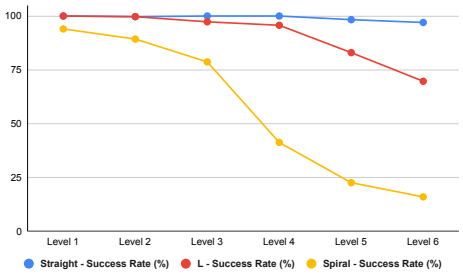
In this testing, we rolled out 300 episodes and computed the mean metric values. At the beginning of the episode, the quadrupeds were placed 0.5 meters in front of the stairs with uniform [-0.3, 0.3] meters of random lateral offsets. The quadrupeds also had [-45, 45] degrees of random heading offset. Then, they were tasked to reach the target position and orientation randomly sampled from the terrain. During this process, metrics are collected and averaged for analysis.

The testing results for straight, L-shaped, and Spiral staircases, are recorded in a YouTube video [15]. Figure 2 shows the YouTube screenshots of the climbing and descending results. Meanwhile, we comparatively demonstrated the performance of the quadrupeds in Figure 3.

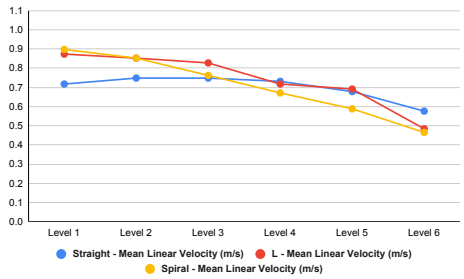
Figure 3 (a) and (b) quantify the reliability of climbing and descending task completion as the experimental difficulty increases. The figures show that higher levels are associated with a lower success rate, especially for spiral stairs. This indicates that the spiral terrain is the most difficult terrain. Many quadrupeds hesitating to climb to the top floor at the 6th difficulty level. We hypothesize that attempting to climb higher stairs can lead to a high penalty due to a fall or collision, so the policy chooses to stay put and not attempt the climb. Figures 3 (c) - (f) show the robot's mean linear velocity and climb/descend rate across different difficulty levels. Mean linear velocity generally decreases as level increases. Climb/descend rate shows an overall increasing trend with level. It suggests that the robot's velocity and climb/descend rate were less influenced by the complexity and difficulty of the stairs. As for the error handling, Figures 3 (g) - (j) show that position and heading errors both increase with level. Robots climbing/descending straight stairs have the smallest errors in both cases. Robots exhibit high position and heading errors in the L-shaped and Spiral staircases, par-



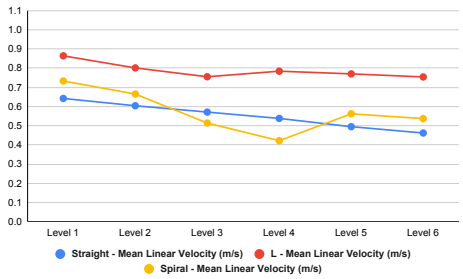
(a) Ascending - Levels vs. Success Rate (%)



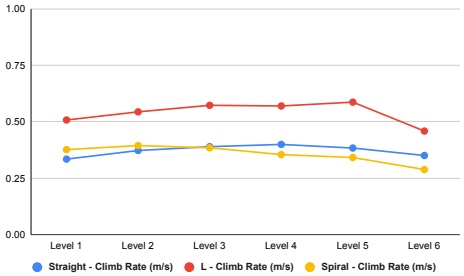
(b) Descending - Levels vs. Success Rate (%)



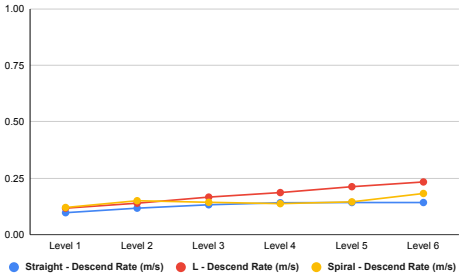
(c) Ascending - Levels vs. Linear Velocity (m/s)



(d) Descending - Levels vs. Linear Velocity (m/s)



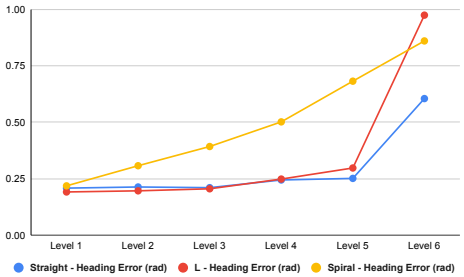
(e) Ascending - Levels vs. Climb Rate (m/s)



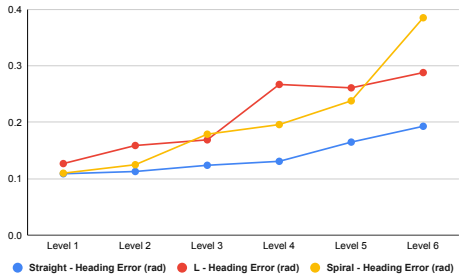
(f) Descending - Levels vs. Descend Rate (m/s)

(g) Ascending - Levels vs. Position Error (m)

(h) Descending - Levels vs. Position Error (m)

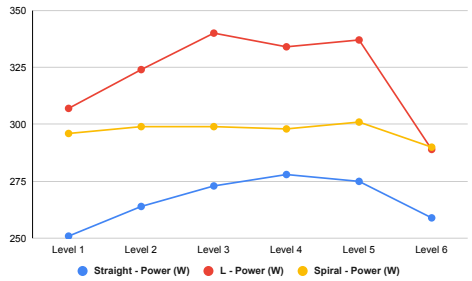


(i) Ascending - Levels vs. Heading Error (rad)

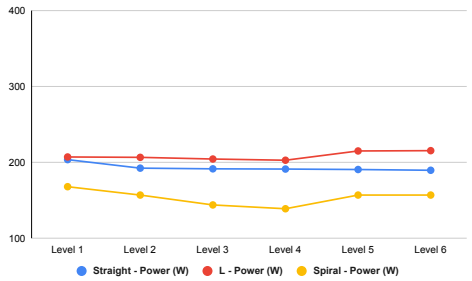


(j) Descending - Levels vs. Heading Error (rad)

Figure 3. Ascending vs. Descending Comparison for Straight, L-shaped, and Spiral Stairs



(k) Ascending - Power (W)



(l) Descending - Power (W)

Figure 3. Ascending vs. Descending Comparison for Straight, L-shaped, and Spiral Stairs (Continued)

ticularly, there is a surge from the 5th to 6th level. This is expected, as many target poses are set on the stairs, it becomes increasingly difficult to align the quadruped's body to target poses on steep stairs. As for power consumption, shown in Figures 3 (k) and (l), robots consume more power as the complexity and difficulty of the staircases increase. It should be noted that the decrease in power consumption at the 6th level could result from the policy adopting more conservative and slower motions to avoid failure, as aggressive behaviors will result in a penalty for failures.

5.2 Discussion and Limitation

We assessed the significance of two-stage training and compared the performance of stage 1 and terrain-specific models trained on stage 2. The models are evaluated in level 3 difficulty terrains. Figure 4 shows the stage 1 model's success rate, compared with other models specifically trained for those terrains. The second-stage training can significantly improve model performance. In addition, the stage 1 model performs best in straight terrain, as it is the least difficult. The Stage 1 model performs worst on L-shaped terrain due to its inability to turn around the corner. The quadruped 'short-cuts' the stairs and goes straight toward the goal position, resulting in a collision or fall. The second stage of training enables the quadrupeds to recognize the shape of the stairs and navigate properly around the corner.

Quadrupeds may have a lower success rate of reaching the top floor as difficulty increases, particularly when the staircases reach the 6th level; the success rate drops significantly. We replayed the climbing testing results and found that robot dogs usually hesitated to climb to the top floor at the 6th difficulty level. We speculated that the rewards of staying, as computed by the RL algorithm, are higher than those of taking risks to reach goals. Therefore, the robot chose conservative strategies. A potential strategy to mitigate this issue is to use exploration rewards, which encourage the robot to perform new behaviors.

We only use local height map information rather than

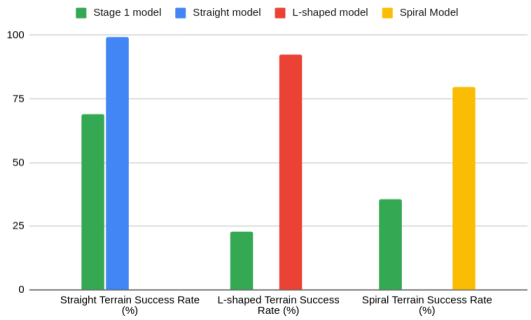


Figure 4. Terrains v.s. Success Rate. Terrains are level 3 difficulty.

a global map, but the results show that the robot can still navigate the stairs with this limited environmental information. It demonstrates the policy's robustness to partial observability, which is highly valued in a firefighting situation where a complete map of the building is unavailable.

Finally, the second-stage training is essential for acquiring navigation capability. The second-stage training, along with its CNN architecture, allows the model to recognize the stair patterns and the ascending direction. Without the stage 2 training, the model fails to recognize the direction of ascent from local terrain information and would choose a straight-line path toward the goal.

In this project, we assumed the stairs were intact, but they may be damaged or obstructed during a fire. In the future, we will test our robot in a harsh environment.

6 Conclusion

In this paper, we designed and tested an End-to-End two-stage RL approach that transfers stair-climbing-descending skills from pyramided terrain to optimize both navigation and locomotion for a robot dog climbing various stairs. It shows a high success rate, high agility, and low position and heading errors with certain diffi-

culty levels. Our RL algorithm design is with effective rewards, regularization, and penalties, and our curriculum strategies implement a cumulative learning process. Our centerline-based navigation formulation enables unified learning of navigation and locomotion without hierarchical planning. In the future, we will optimize our RL training to handle a wider range of scenarios and test them on a real robot by using *Isaac Lab*'s Sim-to-Real workflows.

References

[1] Sebastian Gelfert et al. Novel mobile robot concept for human detection in fire smoke indoor environments using deep learning. In *Proceedings of the 2022 International Conference on Information Technology for Social Good (GoodIT '22)*. ACM, 2022. doi:10.1145/3573910.3573913. URL https://dl.acm.org/doi/10.1145/3573910.3573913.

[2] Noelia Fernández Talavera, Juan Jesús Roldán Gómez, Francisco Martín, and M. C. Rodríguez-Sánchez. An autonomous ground robot to support firefighters' interventions in indoor emergencies. *Journal of Field Robotics*, 40(3): 614–625, 2023. doi:10.1002/rob.22150. URL https://onlinelibrary.wiley.com/doi/10.1002/rob.22150.

[3] Václav Pritzl, Petr Stepan, and Martin Saska. Autonomous flying into buildings in a firefighting scenario. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 239–245. IEEE, 2021. doi:10.1109/ICRA48506.2021.9560789. URL https://dl.acm.org/doi/10.1109/ICRA48506.2021.9560789.

[4] Weitao Han, Guangyu Wu, and Hairen Zhang. Efficient perception for indoor fire rescue uavs: A unified multimodal semantic fusion framework. In *Proceedings of the 3rd International Workshop on UAVs in Multimedia: Capturing the World from a New Perspective (UAVM 2025)*. ACM, 2025. doi:10.1145/3728482.3757382. URL https://dl.acm.org/doi/10.1145/3728482.3757382.

[5] Boston Dynamics. Changing your idea of what robots can do. On-line: https://bostondynamics.com/, Accessed: 01/20/2026.

[6] Unitree. Unitree robotics. On-line: https://www.unitree.com/, Accessed: 01/20/2026.

[7] Shuhao Qi, Wenchun Lin, Zejun Hong, Hua Chen, and Wei Zhang. Perceptive autonomous stair climbing for quadrupedal robots. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2313–2320, 2021. doi:10.1109/IROS51168.2021.9636302.

[8] Qixing Liang, Bin Li, Yiming Xu, Landong Hou, and Xuewen Rong. Vision-based dynamic gait stair climbing algorithm for quadruped robot. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1390–1395, 2022. doi:10.1109/ROBIO55434.2022.10011934.

[9] Dylan Vogel, Robert Baines, Joseph Church, Julian Lotzer, Karl Werner, and Marco Hutter. Robust ladder climbing with a quadrupedal robot. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7239–7244, 2025. doi:10.1109/IROS60139.2025.11247166.

[10] David Hoeller, Nikita Rudin, Dhionis Sako, and Marco Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024. doi:10.1126/scirobotics.adi7566. URL https://www.science.org/doi/abs/10.1126/scirobotics.adi7566.

[11] Hyeongjun Kim, Hyunsik Oh, Jeongsoo Park, Yunho Kim, Donghoon Youm, Moonkyu Jung, Minho Lee, and Jemin Hwangbo. High-speed control and navigation for quadrupedal robots on complex and discrete terrain. *Science Robotics*, 10(102): eads6192, 2025. doi:10.1126/scirobotics.ads6192. URL https://www.science.org/doi/abs/10.1126/scirobotics.ads6192.

[12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[13] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47), October 2020. ISSN 2470-9476. doi:10.1126/scirobotics.abc5986. URL http://dx.doi.org/10.1126/scirobotics.abc5986.

[14] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning, 2022. URL https://arxiv.org/abs/2109.11978.

[15] H. Baiyu. Stair climbing/descending isaaclab simulation. On-line: https://youtu.be/H53FuVl3M_o, Accessed: 03/29/2026.