Vision Based Cross-Embodied Navigation For Quadrupeds

By

Sourabh Prasad

Supervised by
Dr. Goldie Nejat

M.Eng. PROJECT REPORT

DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING
UNIVERSITY OF TORONTO

2025

1 Introduction

Quadrupedal robots, characterized by their four-legged locomotion, possess superior agility for navigating complex terrains where their wheeled counterparts often fail [1]. This capability has established them as promising platforms for diverse applications, including inspection [28], delivery [8], and farming [11].

Since quadrupeds are high-dimensional, non-linear, time-varying systems [1], classical approaches to navigation and control [1, 3, 27, 21], are fundamentally embodiment-specific. These methods rely on explicit, hand-crafted models of the robot's kinematics, dynamics, and sensor footprint. Consequently, a navigation stack engineered for one robot is not readily transferable to another with a different morphology, requiring significant re-tuning or complete redesign. This limitation has been a major catalyst for research into learning-based methods, which aim to create more generalizable and adaptive navigation policies that can accommodate complex embodiments [6] or even operate across embodiments [16].

This report outlines a framework that addresses embodiment-specificity by decoupling navigation into a two-part, hierarchical architecture of general-purpose modules. First, a high-level embodiment-agnostic planner, driven by a foundation model, processes visual sensory data (i.e., RGB images) to generate a sequence of navigational waypoints. Its planning is based solely on visual observations, making it independent of the robot's physical form. Second, a cross-embodied locomotion controller then executes these waypoints. This low-level policy achieves broad adaptability by inferring a robot's unique physical properties from its proprioceptive sensory data, allowing it to translate the abstract waypoints into appropriate joint torques for different morphologies. The primary contribution of this work is this synergistic, end-to-end framework that bridges high-level, abstract visual planning with low-level, embodiment-aware locomotion controller.

To evaluate the system, the locomotion controller's zero-shot generalization was tested in a simulated environment using commercially available robots that were not seen during training. Additionally, the complete navigation stack was evaluated in simulation. This stack was created by coupling the embodiment-agnostic planner with the cross-embodied locomotion controller via a Proportional Derivative (PD) controller. The full system successfully navigated an unseen environment by inferring embodiment properties from sensory observations.

2 Related Works

Prior literature on cross-embodiment navigation can be categorized into two: Deep Reinforcement Learning (DRL) [2, 13, 16, 5] and Imitation Learning (IL) [19, 20, 22].

Deep Reinforcement Learning (DRL) enabling agents to learn policies by optimizing a reward function through environmental interaction [24]. In the context of embodiment-specific locomotion, Lee et al. [10] demonstrated a successful quadruped control policy using model-free RL. They introduced a student-teacher framework where a "teacher" policy, with access to privileged information like terrain data, guides a "student" policy that relies solely on proprioceptive inputs.

Building on these principles, subsequent research has tackled the more complex challenge of cross-embodied locomotion. A prominent strategy involves morphology and domain randomization. For instance, Feng et al. [2] utilized this approach, en-

abling their model to infer robot dynamics and morphology from a history of sensor observations and actions. Similarly, Luo et al. [13] also apply morphology randomization to train a cross-embodiment controller. However, instead of inferring the robot morphology, their method employs a supplementary Multi-Layer Perceptron (MLP) to explicitly estimate the robot's morphology and linear velocity. This randomization paradigm has also been extended to high-level planning. Putta et al. [16] trained a cross-embodiment model for image-goal navigation [30] by sampling the robot's morphology parameters from a continuous range at the start of each training episode. Huang et al. [5] propose a decentralized DRL approach known as Shared Modular Policies (SMP). In contrast to centralized policies that control all of an agent's joints, the SMP architecture instantiates a modular policy at each actuator. Each module operates using only local sensory information from its corresponding actuator. The framework incorporates a messaging system that facilitates communication between the different modules to achieve complex coordination.

In contrast to DRL's trial-and-error approach, Imitation Learning (IL) trains policies by learning directly from expert demonstrations [29]. For example, [17] trained a quadruped gait controller using an expert policy generated by a Model Predictive Controller (MPC).

This paradigm has been utilized to train cross-embodiment, image-goal navigation models [30] by leveraging large, heterogeneous datasets from different robotic platforms. For instance, in [19] Shah et al. proposed the embodiment-agnostic General Navigation Model (GNM), a policy trained on a dataset aggregated from eight distinct robotic platforms. GNM uses visual observations to output relative waypoints that are normalized by the robot's top speed [19]. Building on this work, the Visual Navigation Transformer (ViNT) was introduced as a foundation model that, for exploring new environments, uses a separate image-to-image diffusion model to generate subgoal images conditioned on the robot's current observation [20]. Like GNM, these subgoals are used to predict relative waypoints normalized by the robot's top speed [20]. To improve upon the efficiency of ViNT, NoMaD was proposed. Rather than using a diffusion model to generate high-dimensional images, NoMaD uses a diffusion policy to directly model a distribution of future actions conditioned on the robot's observations [22].

A few papers have also explored the integration of DRL and IL methods to achieve better generalization. In [25], Wang et al. proposes X-Nav, which uses DRL to train expert policies for individual embodiments, which are then distilled into a single, general policy using IL and a transformer architecture. In [12], Liu et al. propose COMPASS, where IL is used to train a foundation model. Residual Reinforcement Learning (RL) [7] is then used to fine-tune this model for specific embodiments before, in a final step, distilling these specialized policies back into a single cross-embodiment policy.

3 Problem Formulation

In the cross-embodiment image-goal navigation problem, a mobile robot is tasked with navigating from a starting position l_s to a goal position l_g, specified using an RGB image. The objective is to learn a control policy, π, that maps a history of robot observations o_t to robot actions a_t. At each time step t, the policy takes as

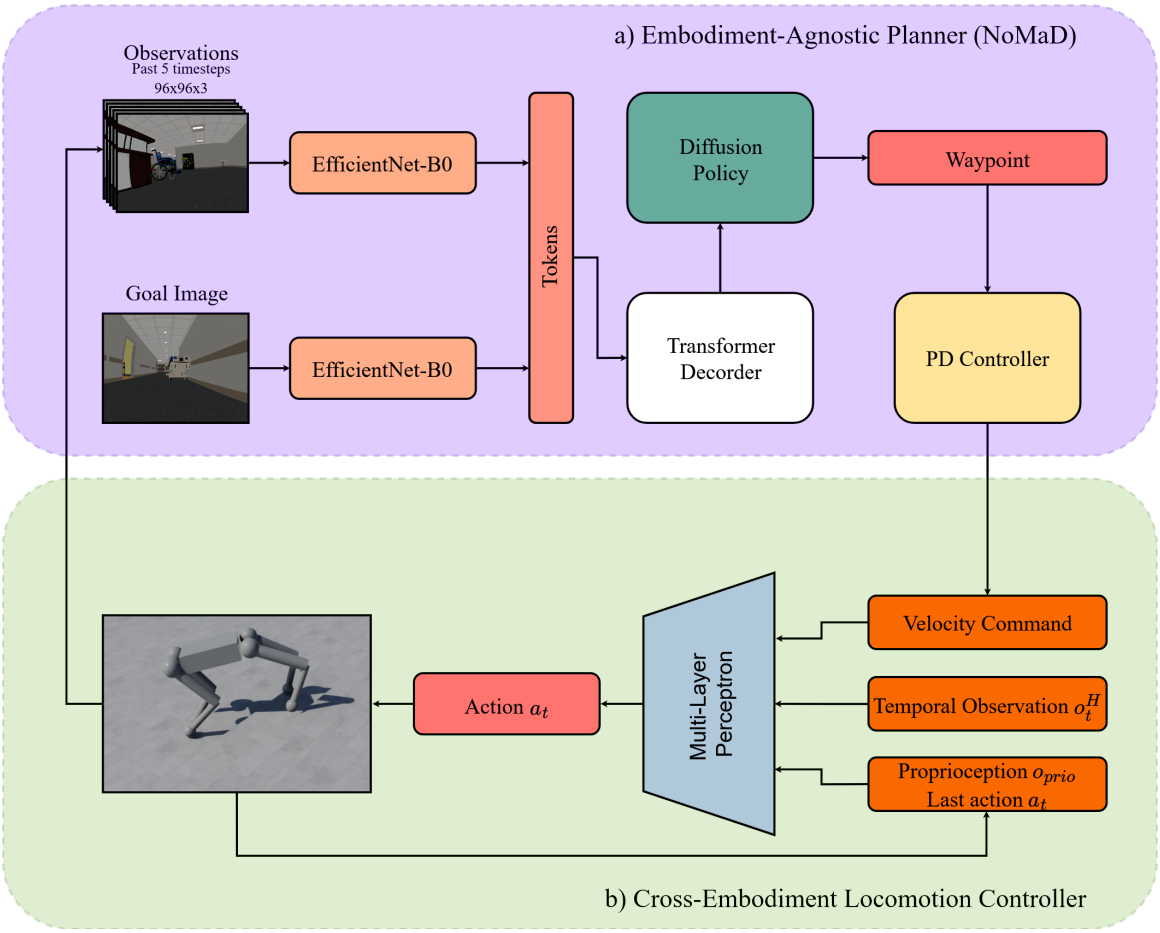


Figure 1: Architecture of proposed navigation stack. a) embodiment-agnostic planner and b) cross-embodiment locomotion controller.

input the observations of the robot, $o_t := o_{t-H:t}$ over the past H timesteps, composed of multimodal sensory data. This observation vector, detailed Equation 1, includes a visual observation (RGB image) $o_{visual} \in \mathbb{R}^{96\times96\times3}$, a goal image $o_{goal} \in \mathbb{R}^{96\times96\times3}$, LiDAR-based depth measurements $o_{depth} \in \mathbb{R}^{16\times10}$, and proprioceptive readings $o_{prio} \in \mathbb{R}^{30}$.

$$o_t = [o_{visual}, o_{goal}, o_{depth}, o_{prio}] \tag{1}$$

The proprioceptive observations, o_{prio}, are sourced from the robot's Inertial Measurement Unit (IMU) and motor encoders. As shown in Equation 2, this vector comprises the linear $v_t \in \mathbb{R}^3$, and angular velocities $\omega_t \in \mathbb{R}^3$ of the robot's base, a projected gravity vector in the robot's base frame $g_t \in \mathbb{R}^3$, and the positions $q_t \in \mathbb{R}^{12}$ and velocities $\dot{q}_t \in \mathbb{R}^{12}$ of the robot's 12 joints. The base frame of the robot is specified as the origin at the center of the robot's trunk.

$$o_{prio} = [v_t, \omega_t, g_t, q_t, \dot{q}_t] \tag{2}$$

For a given observation o_t, the action $a_t \in \mathbb{R}^{12}$ is sampled from a Guassian distribution $\mathcal{N}(\mu, \sigma^2)$. The mean μ is the direct output of the actor network, while the standard deviation σ is a learned action noise parameter. The resulting action a_t specifies the desired target positions for the robot's 12 joints. These targets are then

Table 1: Randomization ranges

(a) Morphology

Parameter	Range	Parameter	Range
Base length	[0.24, 0.91] m	Thigh mass	[0.56, 1.69] kg
Base width	[0.16, 0.39] m	Calf radius	[0.02, 0.05] m
Base height	[0.06, 0.21] m	Calf length	[0.12, 0.39] m
Base mass	[4.8, 19.5] kg	Calf mass	[0.12, 0.39] kg
Thigh radius	[0.02, 0.05] m	Motor P gain	[0.7, 1.3]
Thigh length	[0.16, 0.46] m	Motor D gain	[0.7, 1.3]

(b) Domain

Parameter	Range
Static friction	[0.24, 0.91] m
Dynamic friction	[0.16, 0.39] m
Payload	[0.06, 0.21] m
Disturbance	[-0.5, 0.5] m/s

tracked by a low-level PD controller that generates the corresponding motor torques [13].

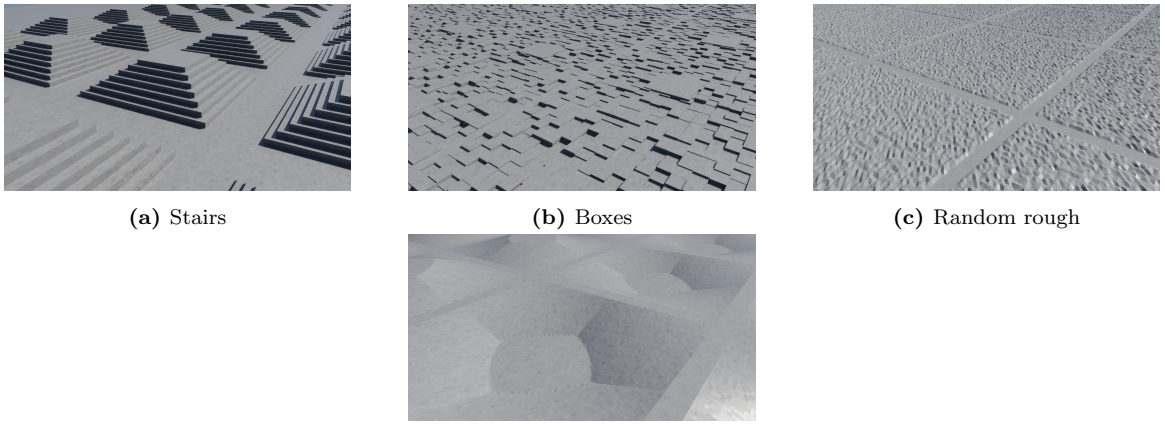
4 Methodology

The proposed architecture, shown in Figure 1, consists of two components: an embodiment-agnostic planner and a cross-embodied locomotion controller.

4.1 Embodiment-Agnostic Planner

To facilitate the planner, we adopt the architecture of NoMaD [22], a unified model designed for both goal-directed navigation and undirected exploration. The model takes as input a sequence of the robot's most recent visual observations, o_{visual}, from an onboard RGB camera, along with a goal image, o_{goal}, from the desired location [22]. EfficientNet-B0 [23] encoders process the observation and goal images to generate input tokens [22]. These tokens are fed into a Transformer backbone to generate a single context vector c_t [22].

The context vector is then simultaneously fed into two separate prediction heads. One head uses a fully-connected layer to predict the temporal distance to the goal, $d(o_{visual}, o_{goal})$ [22], while the other head uses the context vector to condition a diffusion policy, which models a distribution over future actions, as relative waypoints, $p(w_t|c_t)$. This diffusion policy iteratively refines a randomly sampled action sequence over a series of denoising steps to produce a final, collision-free trajectory of future waypoints, w_t [22]. NoMaD is trained end-to-end using supervised learning on the combination of GNM [19] and SACSoN [4] datasets [22].



(a) Stairs **(b)** Boxes **(c)** Random rough

(d) Slope

Figure 2: Curriculum subterrain types.

A PD controller serves as an interface between the NoMaD planner and the locomotion controller. It takes a relative waypoint from NoMaD and converts the resulting positional error into a target velocity command, $v_{com} \in \mathbb{R}^3$. This command, defined as $v_{com} := [v_x, v_y, \omega]$ specifies the robot's target linear velocities along the x and y axes and its angular velocity about the z-axis. This velocity command is then passed as input to the locomotion controller, detailed in the subsequent section.

4.2 Cross-Embodied Locomotion Controller

The cross-embodied locomotion controller is trained in simulation using DRL. The policy is trained on a diverse population of randomly generated quadruped morphologies to enable the policy to generalize across different embodiments. The physical parameters of each agent, such as body mass, leg lengths, and joint friction, are randomized according to the ranges specified in Table 1. In addition to the embodiment randomization, domain randomization, within pre-defined ranges, is introduced to improve the robustness of the controller. This includes body friction coefficients and payload at the beginning of each episode, push disturbances applied by adding velocity to the agent's base at intervals, and the addition of noise to observations and actions. The policy is trained using PPO [18] algorithm.

At each timestep $t(0 \leq t < T)$, the policy's input consists of the current proprioceptive observation o_{prio} and temporal observations $p_t^H = [o_{t-1}, o_{t-2}, ..., o_{t-H}]$, where p_t^H denotes history of state and action over the past H timesteps. This history allows the model to infer a latent representation of the agent's specific morphology. Conditioned on both the current and temporal observations, the policy outputs an action a_t, corresponding to the target positions for the quadruped's motors.

4.2.1 Rewards

Inspired by prior work on legged locomotion [25, 14, 13, 6], the reward function r is structured as a sum of a task-oriented component r_{task} and a behavioural regularization component r_{reg}.

$$r = r_{task} + r_{reg} \tag{3}$$

Table 2: Reward Terms

Term	Reward	Equation	Weight
Task	Lin. velocity tracking	$-e^{4\|\|v_{xy}^{com}-v_{xy}\|\|^2}$	3.0
	Ang. velocity tracking	$-e^{4\|\|\omega_{com}-\omega\|\|^2}$	1.5
	Collision	$\mathbb{1}_{collision}$	-40.0
	Fall	$\mathbb{1}_{tilt}(tilt_{base} > 60°)$	-300.0
Regularization	Lin. velocity (z)	v_z^2	-2.0
	Ang. velocity (xy)	$\|\|\omega_{xy}\|\|^2$	-0.05
	Joint torque	$\|\|\tau\|\|^2$	$-2e^{-4}$
	Joint Accel.	$\|\|\ddot{q}\|\|^2$	$-2.5e^{-8}$
	Action rate	$\|\|a_t - a_{t-1}\|\|^2$	-0.05
	Smooth action	$\|\|a_t - 2a_{t-1} + a_{t-2}\|\|^2$	-0.05
	Swing phase tracking	$\sum_{\text{foot}}[(1 - C_{\text{foot}})e^{\frac{-\|\|\mathbf{f}^{foot}\|\|}{100}^2}]$	-5.0
	Stance phase tracking	$\sum_{\text{foot}}[C_{\text{foot}}e^{\frac{-\|\|v_{xy}^{foot}\|\|}{2}^2}]$	-15.0
	Feet slide	$\|\|v_{xy}^{foot}\|\|^2$	-5.0
	Join deviation	$\|\|q - q_{default}\|\|^2$	-0.5

The task reward incentivizes the primary objective of accurately tracking the target velocity command, while the regularization term helps maintain a stable and smooth gait. Table 2 details the breakdown of the reward function.

4.2.2 Curriculum

A game-inspired curriculum [26] is used to train the locomotion policy progressively using increasingly difficult terrains. The curriculum is structured around a training environment composed of four distinct sub-terrain types: stairs, boxes, random rough and slopes, as illustrated in Figure 2. Training begins at the lowest difficulty level and dynamically adjusts based on the agent's performance in each episode. If the agent successfully traverses a predefined distance threshold, it is promoted to the next difficulty level in the subsequent episode. Conversely, failure to meet this threshold results in the agent being demoted to the previous, easier difficulty level.

4.3 Training

The locomotion controller was trained for 10,000 iterations. 4096 agents with different morphologies were trained in parallel using NVIDIA's Isaac Sim simulator and Isaac Lab learning framework [15]. The policy is a 4-layer MLP with three hidden layers

Table 3: Locomotion controller evaluation results (success rate and velocity tracking error)

(a) Stairs

Quadruped	SR	VTE		
		v_x	v_y	ω
GO2	76%	0.52	0.52	0.70
A1	86%	0.54	0.52	0.45

(b) Boxes

Quadruped	SR	VTE		
		v_x	v_y	ω
GO2	82%	0.51	0.50	0.75
A1	71%	0.51	0.51	0.66

(c) Random Rough

Quadruped	SR	VTE		
		v_x	v_y	ω
GO2	100%	0.52	0.51	0.32
A1	100%	0.51	0.53	0.26

containing [512, 256, 128] units, respectively. The curriculum terrain consisted of 200 subterrains, each subterrain measuring 8×8 meters, and a distance threshold of 4 meters. The policy was trained using the Adam optimizer [9] with a learning rate of 0.001. The training was performed on a desktop PC with an NVIDIA RTX 4080 SUPER GPU, an AMD 5900X CPU, and 32GB of RAM. The training of the controller took approximately 3 hours.

For NoMaD, the open-source model and pre-trained weights provided by the authors were used.

4.4 Evaluation

The performance of both the locomotion controller and the complete, end-to-end navigation stack was evaluated through two distinct sets of simulation experiments. To assess the locomotion controller's generalization, it was deployed on two unseen commercial quadrupeds across a series of challenging terrains, quantifying its performance using Success Rate (SR) and Velocity Tracking Error (VTE). The complete navigation stack was evaluated separately in a photorealistic hospital environment, where its ability to perform collision-free navigation was measured by its goal-reaching SR and Collision Rate (CR).

4.4.1 Locomotion Controller

The generalization of the locomotion controller was assessed by deploying it on two commercially available quadruped models that were previously unseen during the training phase: the Unitree Go2 and the Unitree A1. Performance was quantified using two primary metrics: SR, which measures the percentage of trials where the robot successfully traverses the terrain without falling, and VTE, calculated as the difference between the commanded and actual base velocity along each axis. To gather these metrics, each robot was tested for 1000 episodes on the same categories of subterrain used during training, but with updated parameters such as step height and box height. The aggregated results, showing the average success rate and velocity tracking error for each robot on each subterrain type, are presented in Table 3.

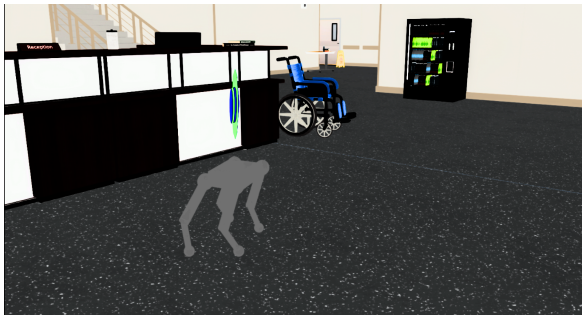


Figure 3: Quadruped navigating through Isaac Sim's hospital environment.

The locomotion controller achieves a high success rate on both the Go2 and A1 platforms. The slight decrease in success rate on the stairs and boxes subterrains is an expected outcome, as these test environments were intentionally designed with greater step and box heights than those seen during training. This was done to evaluate the controller's capabilities in more challenging scenarios. Similarly, the average velocity error can be attributed to these harsher subterrains, which require the controller to prioritize stability by taking corrective actions that may temporarily deviate from precise velocity tracking.

4.4.2 Navigation Stack

The performance of the complete navigation stack, which combines the NoMaD planner and the locomotion controller, was evaluated in a photorealistic hospital environment in Isaac Sim. For the evaluation, a randomly generated quadruped from the locomotion controller training set was equipped with a front-facing fisheye camera with a 170° field of view (FOV). The navigation goal for each trial was an image, captured by an identical camera, from a position randomly sampled from six pre-defined locations. An episode was considered a success if the robot navigated within a 5-meter radius of the goal location. Conversely, an episode was marked as a failure if the robot exceeded the 5-minute time limit or became irrecoverably stuck on an obstacle. The system's performance was evaluated over 100 episodes, and the resulting average SR, which measures if the robot was able to successfully reach the goal location, and CR, which is the average number of collisions per episode, are provided in Table 4.

The framework was able to successfully plan and execute a collision-free path to the goal in 55.4% of the trials, with a CR of 1.46. This SR is notably lower than the results reported in Section 4.4.1, which initially suggested that the episode time limit was the primary constraint. To test this hypothesis, ten additional trials were conducted using the farthest goal location and without the time limit. In all trials, the agent failed to reach the goal (0% SR), becoming irreversibly stuck against obstacles in each case. This behaviour suggests a domain gap, as the NoMaD policy was trained on real-world data, the simulation environment is an out-of-distribution setting. Fine-tuning the policy with data gathered from the simulation environment might yield better results.

Table 4: Navigation stack evaluation results

SR	CR
55.4%	1.46

5 Conclusion

This project successfully developed and evaluated a hierarchical framework for cross-embodied quadruped navigation, demonstrating the potential of decoupling high-level visual planning from low-level locomotion control. The core contribution is a modular system combining a pre-trained, embodiment-agnostic planner (NoMaD) with a novel, cross-embodied locomotion controller trained through deep reinforcement learning with extensive morphology and domain randomization.

Our evaluation yielded two key findings. First, the locomotion controller demonstrated strong zero-shot generalization, successfully executing velocity commands on two unseen commercial quadrupeds (Unitree A1 and Go2) across challenging terrains with high success rates. This result validates the effectiveness of using proprioceptive history and randomization to learn adaptive, morphology-aware locomotion policies. Second, while the integrated navigation stack demonstrated basic functionality, its performance in a simulated hospital environment was limited, achieving a 55.4% success rate. Subsequent analysis revealed that these failures were not due to time constraints but rather a domain gap; the planner, trained on real-world data, struggled to navigate effectively in the simulation, leading to collisions with obstacles.

Future work should prioritize closing this domain gap by fine-tuning the NoMaD planner with data collected from the target simulation environment. This would likely improve obstacle avoidance and significantly increase the overall navigation success rate. Following successful in-simulation fine-tuning, the next step would be to deploy the complete framework on real-world hardware to validate its performance and robustness in complex, unstructured environments.

6 Acknowledgment

I want to express my sincere gratitude to Professor Goldie Nejat for allowing me to pursue this M.Eng project. I am also deeply thankful to Haitong Wang for the tremendous support and mentorship provided throughout this project. I would also like to thank Joel Abraham for providing the GPU used for simulations and training of the model.

References

[1] Hui Chai et al. "A survey of the development of quadruped robots: Joint configuration, dynamic locomotion control method and mobile manipulation approach". In: *Biomimetic Intelligence and Robotics* 2.1 (Mar. 1, 2022), p. 100029. ISSN: 2667-3797. 10.1016/j.birob.2021.100029. https://www.sciencedirect.com/science/article/pii/S2667379721000292 (visited on 08/04/2025).

[2] Gilbert Feng et al. *GenLoco: Generalized Locomotion Controllers for Quadrupedal Robots*. Sept. 12, 2022. 10.48550/arXiv.2209.05309. arXiv: 2209.05309[cs]. http://arxiv.org/abs/2209.05309 (visited on 08/03/2025).

[3] Hassan Taheri and Nasser Mozayani. "A study on quadruped mobile robots". In: *Mechanism and Machine Theory* 190 (Dec. 1, 2023). MAG ID: 4385748641 S2ID: a320c183aa7e9f2fdebd240f0f0b94fdea6b455b, pp. 105448–105448. 10.1016/j.mechmachtheory.2023.105448.

[4] Noriaki Hirose et al. *SACSoN: Scalable Autonomous Control for Social Navigation*. en. arXiv:2306.01874 [cs]. Oct. 2023. 10.48550/arXiv.2306.01874. http://arxiv.org/abs/2306.01874 (visited on 08/08/2025).

[5] Wenlong Huang, Igor Mordatch, and Deepak Pathak. "One Policy to Control Them All:Shared Modular Policies for Agent-Agnostic Control". en. In: ().

[6] Jemin Hwangbo et al. "Learning agile and dynamic motor skills for legged robots". In: *Science Robotics* 4.26 (Jan. 30, 2019), eaau5872. ISSN: 2470-9476. 10.1126/scirobotics.aau5872. arXiv: 1901.08652[cs]. http://arxiv.org/abs/1901.08652 (visited on 08/07/2025).

[7] Tobias Johannink et al. *Residual Reinforcement Learning for Robot Control*. arXiv:1812.03201 [cs]. Dec. 2018. 10.48550/arXiv.1812.03201. http://arxiv.org/abs/1812.03201 (visited on 08/13/2025).

[8] Joshua Hooks et al. "ALPHRED: A Multi-Modal Operations Quadruped Robot for Package Delivery Applications". In: *IEEE Robotics and Automation Letters* 5.4 (July 7, 2020). MAG ID: 3041790910, pp. 5409–5416. 10.1109/lra.2020.3007482.

[9] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. en. arXiv:1412.6980 [cs]. Jan. 2017. 10.48550/arXiv.1412.6980. http://arxiv.org/abs/1412.6980 (visited on 08/08/2025).

[10] Joonho Lee et al. "Learning Quadrupedal Locomotion over Challenging Terrain". In: *Science Robotics* 5.47 (Oct. 21, 2020), eabc5986. ISSN: 2470-9476. 10.1126/scirobotics.abc5986. arXiv: 2010.11251[cs]. http://arxiv.org/abs/2010.11251 (visited on 08/04/2025).

[11] Lee Milburn et al. "Computer-Vision Based Real Time Waypoint Generation for Autonomous Vineyard Navigation with Quadruped Robots". In: *2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (Apr. 26, 2023). ARXIV_ID: 2305.01700 MAG ID: 4378190663. 10.1109/icarsc58346.2023.10129563.

[12] Wei Liu et al. *COMPASS: Cross-embodiment Mobility Policy via Residual RL and Skill Synthesis*. en. arXiv:2502.16372 [cs]. Feb. 2025. 10.48550/arXiv.2502.16372. http://arxiv.org/abs/2502.16372 (visited on 08/13/2025).

[13] Zeren Luo et al. "MorAL: Learning Morphologically Adaptive Locomotion Controller for Quadrupedal Robots on Challenging Terrains". In: *IEEE Robotics and Automation Letters* 9.5 (May 2024), pp. 4019–4026. ISSN: 2377-3766, 2377-3774. 10.1109/LRA.2024.3375086. https://ieeexplore.ieee.org/document/10463132/ (visited on 08/02/2025).

[14] Gabriel B. Margolis and Pulkit Agrawal. *Walk These Ways: Tuning Robot Control for Generalization with Multiplicity of Behavior*. Dec. 6, 2022. 10.48550/arXiv.2212.03238. arXiv: 2212.03238[cs]. http://arxiv.org/abs/2212.03238 (visited on 08/02/2025).

[15] Mayank Mittal et al. "Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments". en. In: *IEEE Robotics and Automation Letters* 8.6 (June 2023). arXiv:2301.04195 [cs], pp. 3740–3747. ISSN: 2377-3766, 2377-3774. 10.1109/LRA.2023.3270034. http://arxiv.org/abs/2301.04195 (visited on 08/08/2025).

[16] Pranav Putta et al. "Embodiment Randomization for Cross Embodiment Navigation". In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). ISSN: 2153-0866. Oct. 2024, pp. 5527–5534. 10.1109/IROS58592.2024.10802792. https://ieeexplore.ieee.org/document/10802792 (visited on 08/08/2025).

[17] Alexander Reske et al. "Imitation Learning from MPC for Quadrupedal Multi-Gait Control". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. May 30, 2021, pp. 5014–5020. 10.1109/ICRA48506.2021.9561444. arXiv: 2103.14331[cs]. http://arxiv.org/abs/2103.14331 (visited on 08/04/2025).

[18] John Schulman et al. *Proximal Policy Optimization Algorithms*. en. arXiv:1707.06347 [cs]. Aug. 2017. 10.48550/arXiv.1707.06347. http://arxiv.org/abs/1707.06347 (visited on 08/08/2025).

[19] Dhruv Shah et al. *GNM: A General Navigation Model to Drive Any Robot*. May 22, 2023. 10.48550/arXiv.2210.03370. arXiv: 2210.03370[cs]. http://arxiv.org/abs/2210.03370 (visited on 08/08/2025).

[20] Dhruv Shah et al. *ViNT: A Foundation Model for Visual Navigation*. en. arXiv:2306.14846 [cs]. Oct. 2023. 10.48550/arXiv.2306.14846. http://arxiv.org/abs/2306.14846 (visited on 08/13/2025).

[21] Phani Teja Singamaneni et al. "A Survey on Socially Aware Robot Navigation: Taxonomy and Future Challenges". en. In: *The International Journal of Robotics Research* 43.10 (Sept. 2024). arXiv:2311.06922 [cs], pp. 1533–1572. ISSN: 0278-3649, 1741-3176. 10.1177/02783649241230562. http://arxiv.org/abs/2311.06922 (visited on 08/12/2025).

[22] Ajay Sridhar et al. *NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration*. Oct. 11, 2023. 10.48550/arXiv.2310.07896. arXiv: 2310.07896[cs]. http://arxiv.org/abs/2310.07896 (visited on 08/02/2025).

[23] Mingxing Tan and Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. arXiv:1905.11946 [cs]. Sept. 2020. 10.48550/arXiv.1905.11946. http://arxiv.org/abs/1905.11946 (visited on 08/19/2025).

[24] Chen Tang et al. *Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes*. Sept. 16, 2024. 10.48550/arXiv.2408.03539. arXiv: 2408.03539[cs]. http://arxiv.org/abs/2408.03539 (visited on 08/02/2025).

[25] Haitong Wang and Goldie Nejat. "X-Nav: Learning End-to-End Cross-Embodiment Navigation for Mobile Robots". In: ().

[26] Xin Wang, Yudong Chen, and Wenwu Zhu. “A Survey on Curriculum Learning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (Sept. 2022), pp. 4555–4576. ISSN: 1939-3539. 10.1109/TPAMI.2021.3069908. https://ieeexplore.ieee.org/document/9392296 (visited on 08/08/2025).

[27] Zhenyu Wu et al. “A survey on legged robots: Advances, technologies and applications”. In: *Engineering Applications of Artificial Intelligence* 138 (Dec. 1, 2024), p. 109418. ISSN: 0952-1976. 10.1016/j.engappai.2024.109418. https://www.sciencedirect.com/science/article/pii/S0952197624015768 (visited on 08/08/2025).

[28] Yaosheng Huang et al. “Autonomous positioning and navigation of a quadruped robot for power transmission inspection integrating IMU and visual SLAM”. In: *2025 4th International Symposium on Computer Applications and Information Technology (ISCAIT)* (2025). S2ID: 9e44c31b8e7df81c7cd5dc8d78ac612fdbaae81f. 10.1109/iscait64916.2025.11010319.

[29] Maryam Zare et al. *A Survey of Imitation Learning: Algorithms, Recent Developments, and Challenges*. en. arXiv:2309.02473 [cs]. Sept. 2023. 10.48550/arXiv.2309.02473. http://arxiv.org/abs/2309.02473 (visited on 08/13/2025).

[30] Yuke Zhu et al. “Target-driven visual navigation in indoor scenes using deep reinforcement learning”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. May 2017, pp. 3357–3364. 10.1109/ICRA.2017.7989381. https://ieeexplore.ieee.org/abstract/document/7989381 (visited on 08/12/2025).