
# GaitNet: Greedy, Dynamic Quadruped Gait Generation

**Owen Sullivan**

Supervisors:
Dr. Mohammad Mahdi Agheli Hajiabadi

Worcester Polytechnic Institute

September 12, 2025

# Contents

# List of Tables

# List of Figures

# 1 Introduction

Quadruped locomotion in unstructured environments remains a complex challenge, particularly when traditional gait cycles are insufficient. While many systems rely on periodic gait patterns or centralized planners, recent advances in learning-based methods have opened the door to more adaptive, non-gaited approaches. This proposal investigates a novel method for generating non-gaited, dynamic footstep plans using a CNN-based greedy planner, aiming for a balance between computational efficiency and dynamic behaviors.

## 1.1 Problem Statement

## 1.2 Motivation

## 1.3 Research Question

> Can a greedy, CNN-based planner match the expressiveness of tree search while improving runtime for dynamic, non-gaited locomotion?

## 1.4 Hypothesis

## 1.5 Contributions

## 1.6 Thesis Structure

# 2 Background

Recent advances in legged locomotion have largely focused on either implicit gait-based policies or perception-driven foothold selection modules. However, these approaches often trade off between agility, expressiveness, and computational cost. In contrast to gaited methods that impose rhythmic structure, non-gaited planning allows for more versatile, terrain-adaptive behaviors—but typically at the expense of increased planning complexity. Bridging this gap, our proposed approach draws on greedy methods for their computational efficiency and on convolutional neural networks (CNNs) for terrain-aware generalization, aiming to achieve dynamic, non-gaited footstep planning in real time. This section reviews the foundations upon which our approach builds: learning-based locomotion strategies, footstep planners, greedy selection mechanisms, and foothold classification techniques. Each informs a different aspect of our planner's design and situates our method within the broader landscape of quadruped control.

## 2.1 Locomotion Planning Paradigms

## 2.2 Learning-Based Locomotion Strategies

Recent work has shown the potential of deep learning to generate agile, robust locomotion policies, often without explicit footstep planning. Shi et al. [2] use a neural network to modulate trajectory generator parameters in real time for energy-efficient walking. Xie et al. [3] train reinforcement learning policies on centroidal dynamics models to output desired body accelerations, assuming a fixed foot-placement heuristic and gait pattern. Duan et al. [4] learn step-to-step transitions using proprioception, generating joint targets and varying step frequency for terrain-adaptive behaviors. Siekmann et al. [5] focus on blind locomotion by training an LSTM policy to handle randomized stairs using only proprioceptive feedback. Lee et al. [6] employ a temporal convolutional network to infer terrain structure from proprioceptive history, using an automated curriculum to adapt to progressively harder environments. While these approaches enable robust locomotion across diverse terrains, they generally rely on fixed or implicit gait patterns and lack explicit control over individual footstep selection, making them less suitable for non-gaited or footstep-specific planning.

## 2.3 Learning-Based Footstep Planners

Learning-based footstep planners combine perception and control through data-driven models, often using a learned component upstream of a model-predictive controller. FootstepNet [7] uses deep reinforcement learning to generate heuristic-free plans but is limited to bipeds in simple 2D environments. ContactNet [1] employs a CNN to produce non-gaited footstep sequences using greedy selection, though it is constrained to one-leg-at-a-time motion and coarse foothold discretization. DeepGait [8] separates footstep planning and motion optimization, enabling more modular design but still relying on static gaits. Omar et al. [9] use a CNN to identify safe stepping regions, which are then decomposed into convex regions and passed to an MPC for downstream planning. While efficient and fully onboard, the method relies on fixed gait sequences and is primarily focused on safe terrain classification. Villarreal et al. [10] similarly use a CNN for foothold classification, replacing expert heuristics and enabling real-time performance with significantly faster inference, but their method also depends on a fixed gait schedule. DeepLoco [11] introduces a hierarchical policy for footstep and motion planning, but its coarse terrain input and focus on gaited footstep generation

for bipeds limit dynamic adaptability. Taouil et al. [12] use Monte Carlo Tree Search (MCTS) to explore dynamic and non-gaited stepping combinations, making them first to generate dynamic, non-gaited footstep plans in real time. Their approach does suffer from a high computational cost due to its high branching factor, especially on certain types of terrain [1]. Overall, while these approaches advance learning-based planning, few simultaneously support both non-gaited and dynamically unstable footstep generation.

## 2.4 Greedy and Heuristic Footstep Planners

Greedy planners offer computationally efficient alternatives to exhaustive search, often prioritizing goal-directed heuristics or local stability metrics. Gao et al. [13] propose GH-QP, a greedy-heuristic hybrid that incorporates expected robot speed into a footstep planning heuristic, though it is limited to bipedal systems and lacks support for dynamic footstep plans. Zucker et al. [14] use A* with a learned terrain cost and Dubins car heuristic for footstep planning, enabling effective search in SE(2) but restricted to static, gated locomotion. Kalakrishnan et al. [**?**] combine greedy terrain-cost search with a coarse-to-fine pose optimization pipeline, but rely on fixed gait patterns and struggle with highly constrained environments. While these works demonstrate the potential of greedy methods, none support both dynamically unstable and non-gaited footstep generation. Notably, some prior methods already discussed—such as ContactNet [1], DeepLoco [11], and Villarreal et al. [10]—also incorporate greedy elements in their pipelines.

## 2.5 Foothold Classification Algorithms

Foothold classification algorithms focus on determining safe stepping regions, often using learned terrain models to support downstream planners that enforce safety constraints. Asselmeier et al. [15] generate steppability maps from visual inputs to guide a trajectory-optimization-based planner, validating their perception-to-control pipeline in simulation. Omar et al. [16] propose a two-stage classifier that first identifies steppable regions before selecting footholds, emphasizing safety over expressiveness by evaluating only candidate swing-foot locations. Similarly, Omar et al. [9] use a CNN to identify safe footholds, which are grouped into convex clusters and passed to an MPC for real-time planning, but the approach relies on fixed gait sequences. These methods provide robust terrain understanding but are typically designed as standalone perception modules, separate from the full footstep generation process, and thus are not directly applicable to dynamic or non-gaited motion planning.

## 2.6 Synthesis

# 3 Methodology

## 3.1 System Overview

This work aims to take ContactNet [1] and extend it to provide dynamic gait generation capabilities. ContactNet is able to generate acyclic gaits, and this work builds on top of that to enable moving multiple feet simultaneously. The proposed framework is shown in Figure 3.1.

## 3.2 Simulation Environment

The simulation environment used for this project is NVIDIA Isaac Lab [17]. This was chosen for a number of reasons. It is a modern framework with a Python interface specifically designed for machine learning, and GPU parallelism. The GPU parallelism enabled much faster simulation and data-collection at the cost of more complicated programming and reduced compatibility with older hardware. Additionally, there are a large number of example and community projects demonstrating many of the simulation features needed for this project.

NVIDIA Isaac Lab uses a declarative system of python dataclasses to define the simulation environment. For this work, a custom environment is used (Figure 3.2), including:

- Unitree Go 1—Configured to use force control for each joint.
- Terrain raycasts—Measure the height of the terrain at each possible footstep location. This bypasses the lidar processing steps of a full vision pipeline.
- Custom terrain—Planar terrain with sections missing on a grid pattern. The underlying grid is 8cm square, and the void density is tuned to create a challenging but possible environment. The terrain is designed to mirror that used in [1]

## 3.3 Greedy Dynamic Quadruped Gait Generation

### 3.3.1 Terrain Representation

### 3.3.2 Network Architecture

The footstep evaluation neural network architecture is shown in Figure 3.4 accomplishes a similar task to ContactNet [1]. It takes in terrain and state information and ranks each of the possible footstep positions. This network uses a multi-modal middle fusion architecture. This architecture was chosen to effectively combine the spatial features from the terrain data with the contextual features from the robot state data, while still maintaining flexibility in case re-training is needed [18]. The terrain data is processed through a series of convolutional layers, while the robot state data is processed through fully connected layers. The outputs of these two branches are then concatenated and passed through additional fully connected layers to produce the final output, which consists of preference ratings for each potential footstep position.

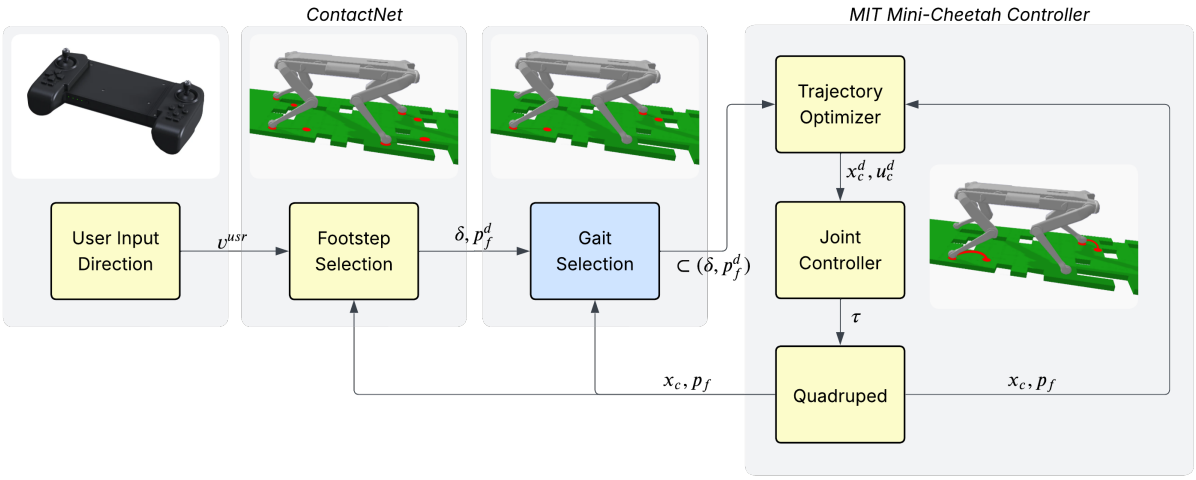


Figure 3.1: A block diagram of the proposed framework. The user defines an input direction $v^{usr}$ which the *footstep selector* [1] uses along with, the robot state $x_c$, and the current foot positions $p_f$ to generate the swing durations $\delta$ and touchdown points $p_f^d$ for all currently grounded feet. The *gait selector* (novel) takes these desired foot movements and selects an appropriate subset $\subset (\delta, p_f^d)$ based on $x_c$, $p_f$, and the terrain data. $\subset (\delta, p_f^d)$ is then passed into the MIT Mini-Cheetah Controller as MPC constraints to perform lower level control.

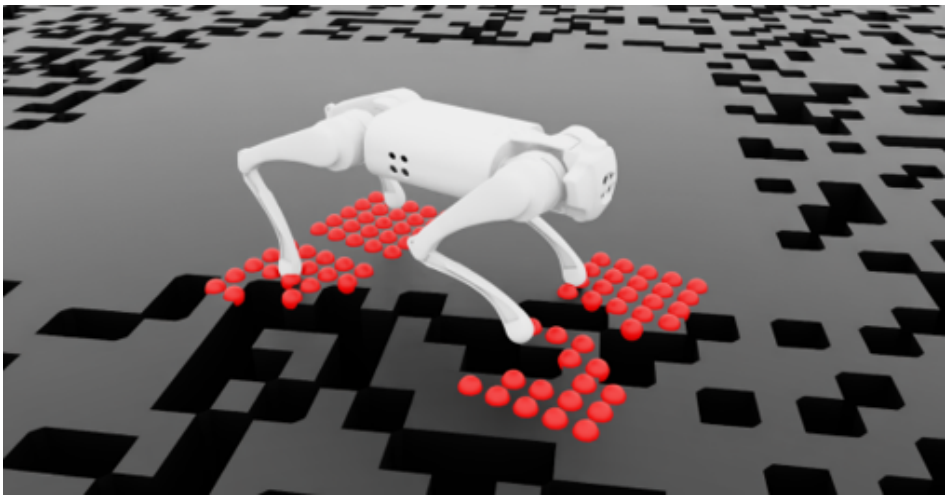

Figure 3.2: Image of a Unitree Go 1 navigating the terrain. Red spheres show the hit location of raycasts. Black regions show voids in the terrain.

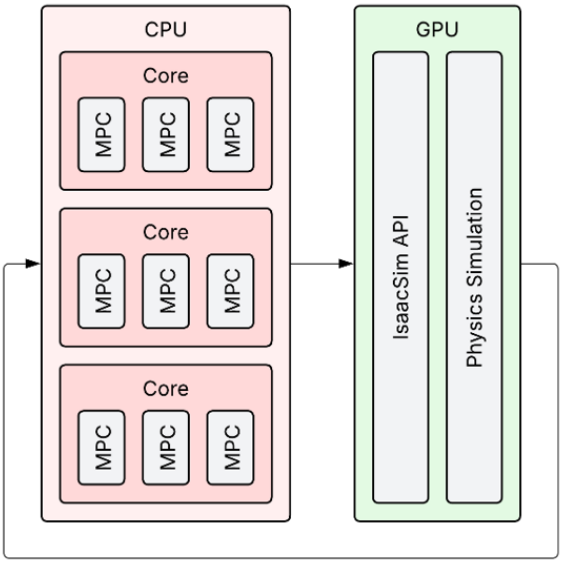


Figure 3.3: A block diagram showing the programming tasks computed on the CPU vs GPU. The full simulation is run in parallel using Nvidia Isaac Lab on the GPU, while the robot MPCs are run in parallel on the CPU.

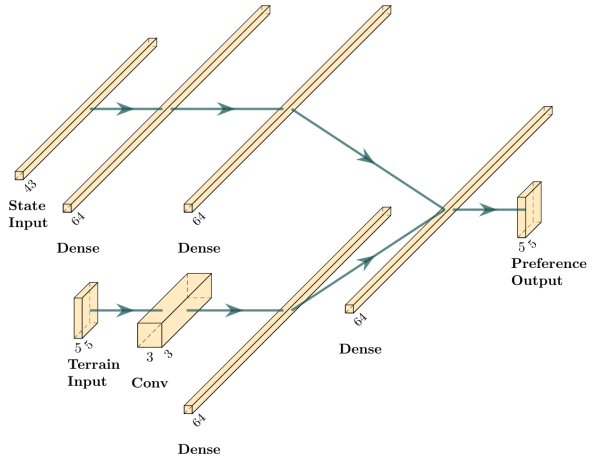


Figure 3.4: Footstep evaluation neural network architecture.

### 3.3.3 Greedy Decision-Making

## 3.4 Training Approach

# 4 Results and Discussion

## 4.1 Experimental Setup

## 4.2 Performance Metrics

## 4.3 Qualitative Analysis

## 4.4 Qualitative Analysis

## 4.5 Discussion

## 4.6 Comparison with Baseline Methods

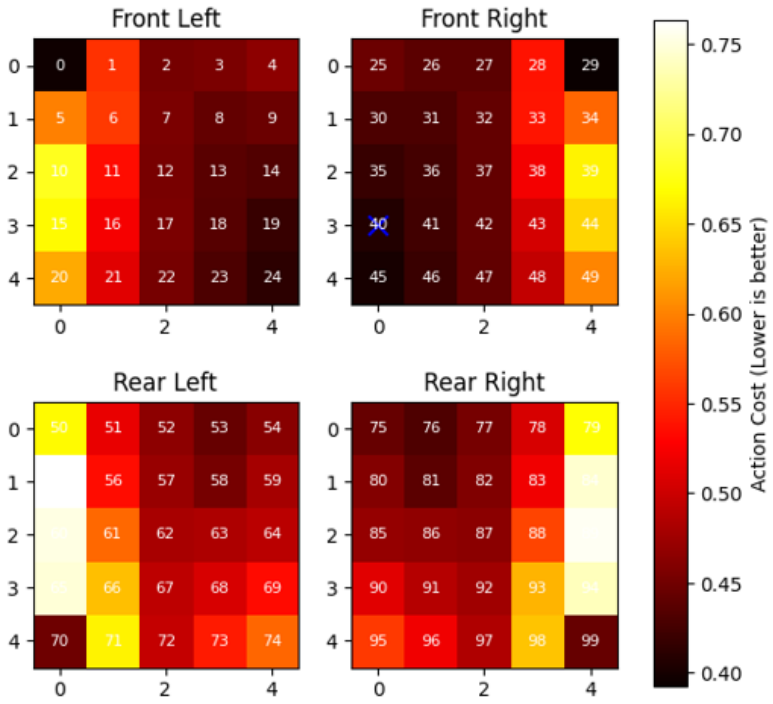


Figure 4.1: Footstep cost map. Shows the heuristic cost for each possible footstep location. Numbers in each plot show the index of the location when all possible locations are combined and flattened to be passed into the NN. Blue "x" indicates the state which the data gathering pipeline will explore next.

# 5 Conclusions and Future Work

## 5.1 Summary of Findings

## 5.2 Limitations

## 5.3 Future Work

## 5.4 Final Remarks

# References


[1] A. Bratta, A. Meduri, M. Focchi, L. Righetti, and C. Semini, "ContactNet: Online multi-contact planning for acyclic legged robot locomotion,"

[2] H. Shi, Q. Zhu, L. Han, W. Chi, T. Li, and M. Q.-H. Meng, "Terrain-aware quadrupedal locomotion via reinforcement learning."

[3] *GLiDE: Generalizable Quadrupedal Locomotion in Diverse Environments with a Centroidal Model*, pp. 523–539. Springer International Publishing. ISSN: 2511-1256, 2511-1264.

[4] H. Duan, A. Malik, J. Dao, A. Saxena, K. Green, J. Siekmann, A. Fern, and J. Hurst, "Sim-to-real learning of footstep-constrained bipedal dynamic walking," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 10428–10434, IEEE.

[5] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning."

[6] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," vol. 5, no. 47.

[7] C. Gaspard, G. Passault, M. Daniel, and O. Ly, "FootstepNet: an efficient actor-critic method for fast on-line bipedal footstep planning and forecasting," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13749–13756, IEEE.

[8] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter, "DeepGait: Planning and control of quadrupedal gaits using deep reinforcement learning."

[9] S. Omar, L. Amatucci, G. Turrisi, V. Barasuol, and C. Semini, "Fast convex visual foothold adaptation for quadrupedal locomotion,"

[10] O. Villarreal, V. Barasuol, M. Camurri, L. Franceschi, M. Focchi, M. Pontil, D. G. Caldwell, and C. Semini, "Fast and continuous foothold adaptation for dynamic locomotion through CNNs," vol. 4, no. 2, pp. 2140–2147.

[11] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne, "DeepLoco: dynamic locomotion skills using hierarchical deep reinforcement learning," vol. 36, no. 4, pp. 1–13. Publisher: Association for Computing Machinery (ACM).

[12] I. Taouil, L. Amatucci, M. Khadiv, A. Dai, V. Barasuol, G. Turrisi, and C. Semini, "Non-gaited legged locomotion with monte-carlo tree search and supervised learning," vol. 10, no. 2, pp. 1265–1272. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

[13] Z. Gao, X. Chen, Z. Yu, C. Li, L. Han, and R. Zhang, "Global footstep planning with greedy and heuristic optimization guided by velocity for biped robot," vol. 238, p. 121798. Publisher: Elsevier BV.

[14] M. Zucker, N. Ratliff, M. Stolle, J. Chestnutt, J. A. Bagnell, C. G. Atkeson, and J. Kuffner, "Optimization and learning for rough terrain legged locomotion," vol. 30, no. 2, pp. 175–191. Publisher: SAGE Publications.

[15] M. Asselmeier, Y. Zhao, and P. A. Vela, "Steppability-informed quadrupedal contact planning through deep visual search heuristics."

[16] S. Omar, L. Amatucci, V. Barasuol, G. Turrisi, and C. Semini, "SafeSteps: Learning safer footstep planning policies for legged robots via model-based priors," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pp. 1–8, IEEE.

[17] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," vol. 8, no. 6, pp. 3740–3747.

[18] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Gläser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2021.