Learning Quadruped Locomotion Policies with Reward Machines

David DeFazio and **Shiqi Zhang**
SUNY Binghamton

Introduction

Background:

- Quadruped robots are useful, and can navigate in challenging terrain such as stairs or rocky hills.
- Learning locomotion policies for quadrupeds is challenging, due to the large state and action spaces.

Challenges:

Traditional methods need a dynamics model, which is often unavailable in real-world settings. While possible to learn policies via reinforcement learning (RL), which does not need a dynamics model, the following issues emerge:

- A prohibitive amount of experience is required to learn good policies.
- It is difficult to specify different locomotion styles.

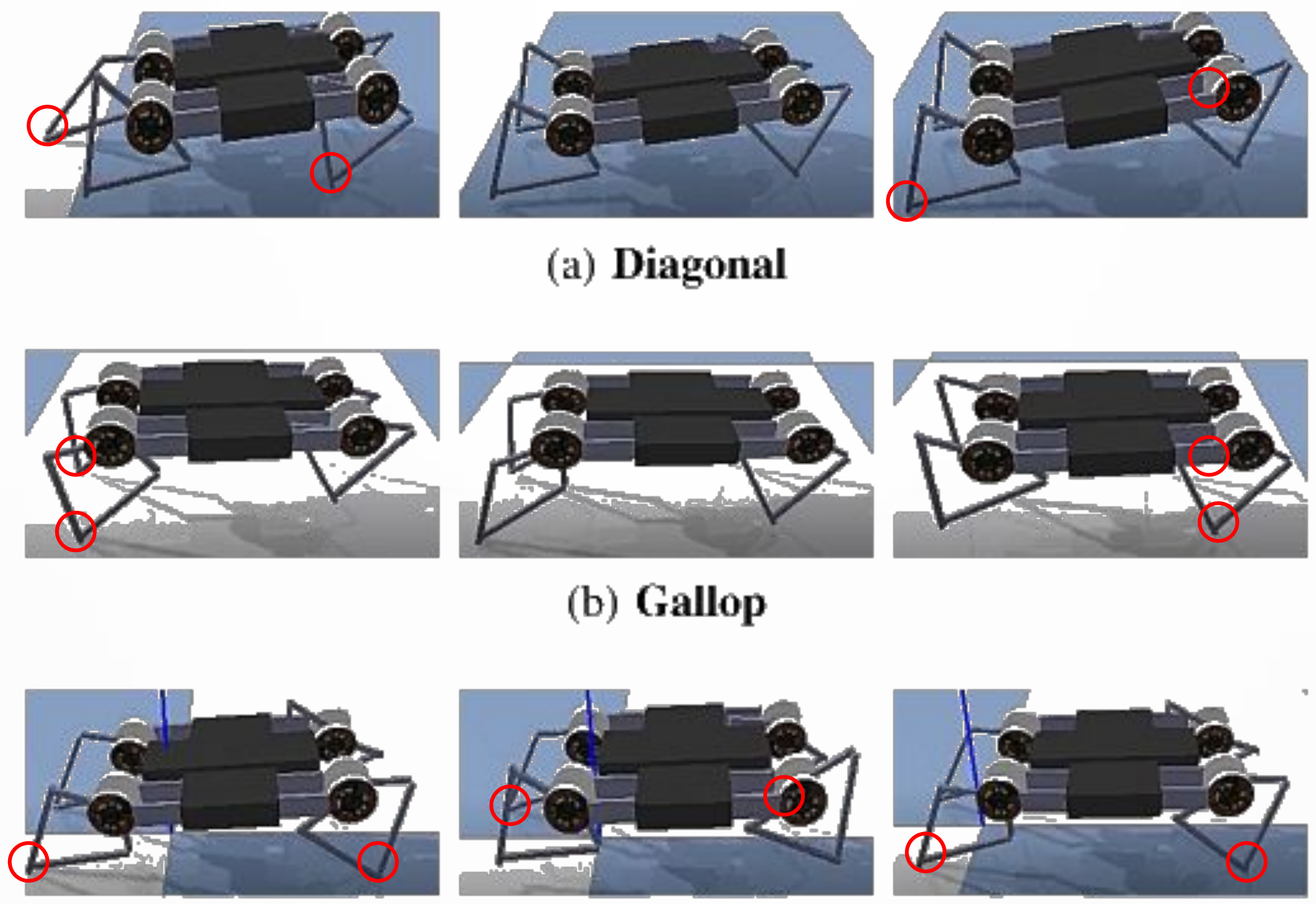
Our Approach:

We leverage human knowledge in the form of LTL formulas via the Reward Machine (RM) framework. We find it simple to specify diverse locomotion behaviors.

Experimental results indicate that our approach improves learning efficiency compared to a non-RM baseline.

Acknowledgments

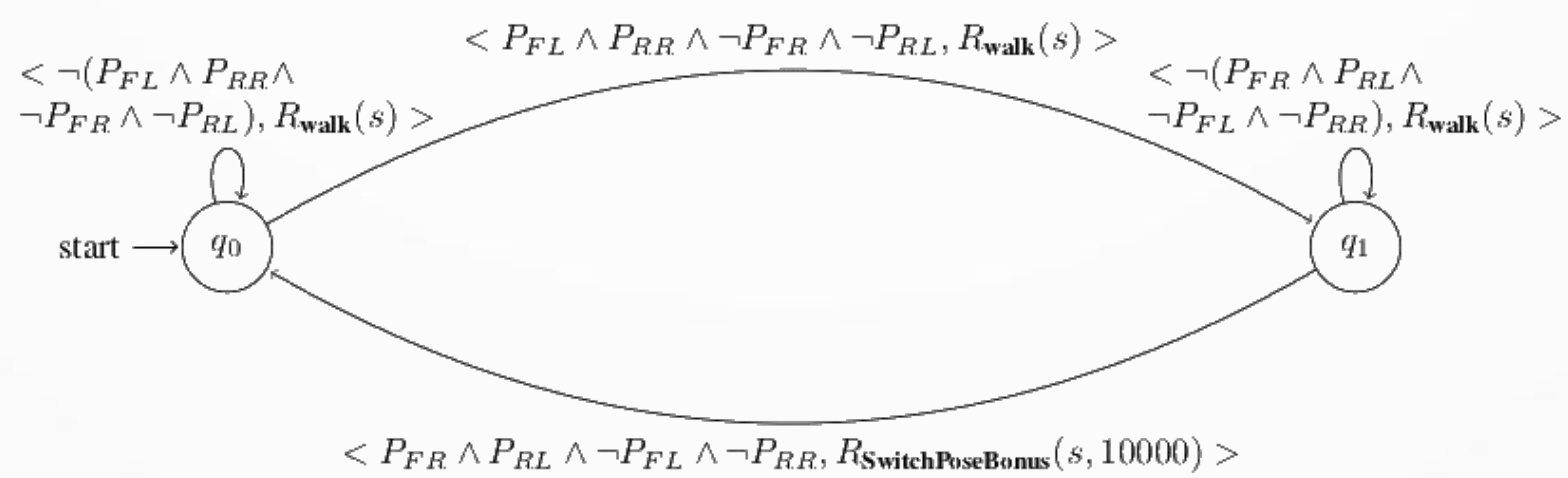
This research has taken place at the Autonomous Intelligent Robotics (AIR) Group, SUNY Binghamton. AIR research is supported in part by grants from the National Science Foundation (NRI-1925044), Ford Motor Company (URP Awards 2019-2021), OPPO (Faculty Research Award 2020), and SUNY Research Foundation.



(a) **Diagonal**

(b) **Gallop**

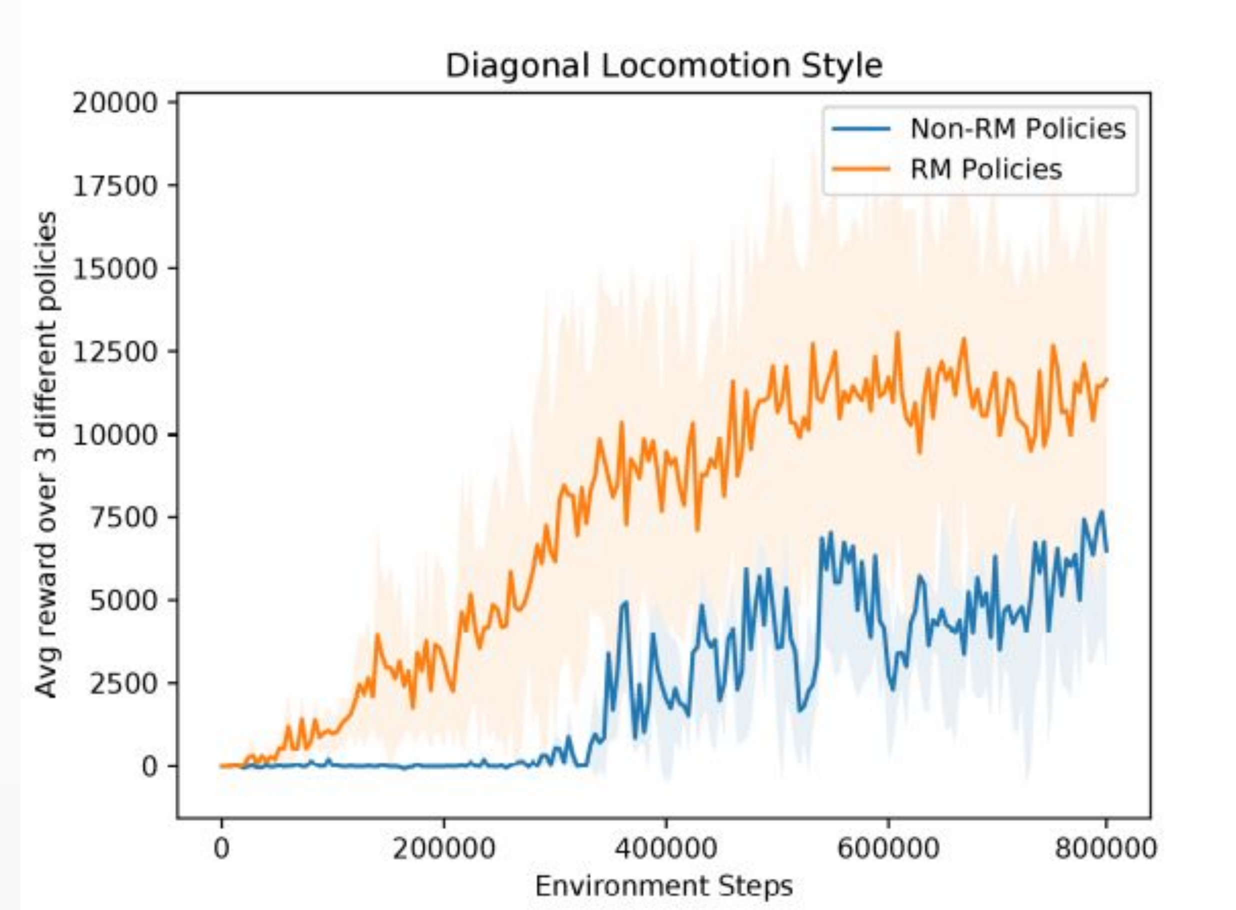
(c) **Dog**



Results

We present a reward curve for the **Diagonal** locomotion style.

Our results indicate our approach improved learning efficiency, due to the fact that the RM algorithm used to leverage the automaton is capable of more efficiently learning from when the robot changes pose.



Methodology

We use the Reward Machine framework to specify our task via an automaton with LTL formulas defining transitions:

$$P_{FR} \wedge P_{RL} \wedge \neg P_{FL} \wedge \neg P_{RR}$$

Front-Right and Rear-Left legs in the air, Front-Left and Rear-Right legs on the ground

$$P_{FL} \wedge P_{RR} \wedge \neg P_{FR} \wedge \neg P_{RL}$$

Front-Left and Rear-Right legs in the air, Front-Right and Rear-Left legs on the ground

The automaton decomposes the locomotion task into two components. The robot must first lift the front-left and rear-right legs, and then must lift the front-right and rear-left legs.

The RM framework constructs an MDP from this automaton, and provides algorithms to efficiently solve the MDP by taking advantage of the automaton structure.

Experiments

We simulate a Minitaur robot, and compare our approach against a non-RM baseline with the same reward function across three different locomotion styles:

Diagonal: Synchronize FR with RL leg, and FL with RR leg.

Gallop: Synchronize FR with FL leg, and RR with RL leg.

Dog: Synchronize FR with RR leg, and FL with RL leg.

Both approaches use the Soft Actor Critic algorithm to learn policies in their respective environments.