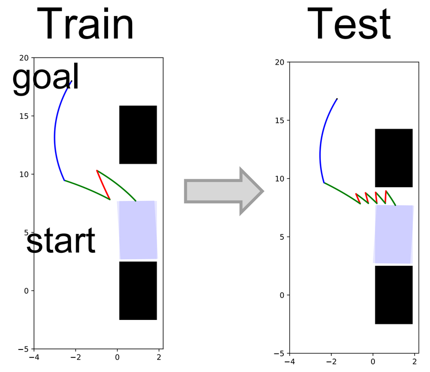
Neurosymbolic learning for Robust Robots

Jeevana Priya Inala
Advisor: Armando Solar-Lezama
MIT CSAIL

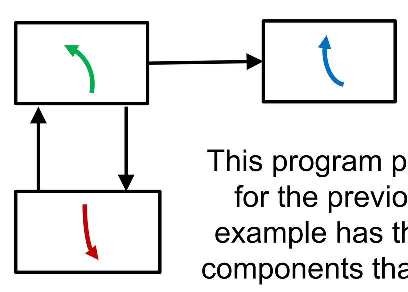
OBJECTIVES

- Our goal is to learn **symbolic/programmatic** design and control for **robots.**
- Compared to neural network models, program models have several benefits:

Generalization
Can learn models from a few examples and generalize to other examples

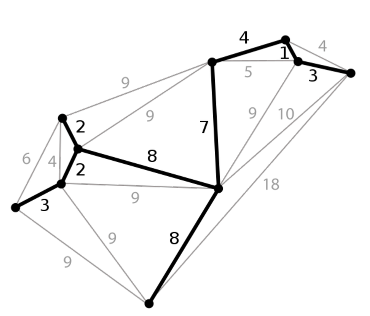


Interpretability
Program policies are easy to understand



This program policy for the previous example has three components that are connected as shown here.

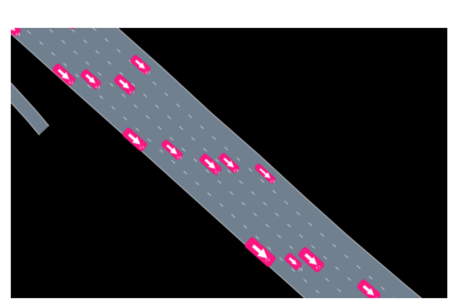
Constraint enforcement
It is easy to enforce combinatorial constraints regarding robots with programs

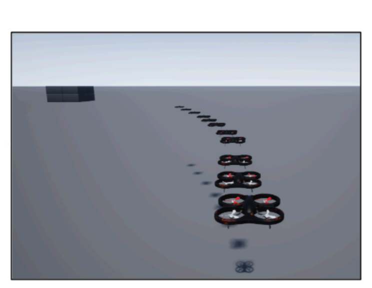


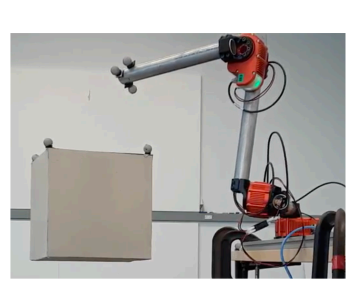
Composability
Can learn how to compose a program (robot) from modular pieces

APPLICATIONS

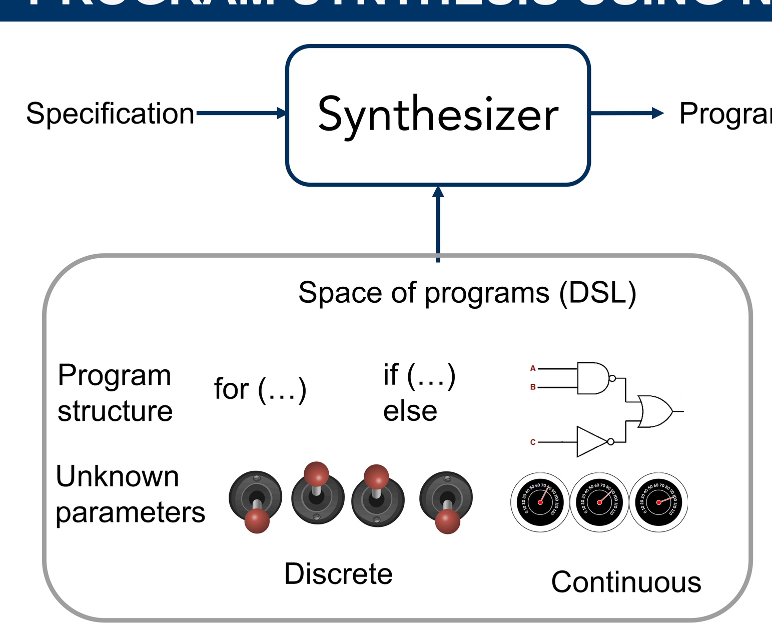
Autonomous Systems [1] | Drone Swarms [2] | Manipulators [3]







PROGRAM SYNTHESIS USING NEUROSYMBOLIC LEARNING



Challenge: Programs have both discrete and continuous components

Solution: A ***neurosymbolic*** learning approach that combines

1. Neural networks / numerical optimization
2. Discrete search techniques

Generalizable Policies for Reinforcement Learning [1]

Generalization

- Many **control tasks** such as walking, and swimming require performing **repeating behaviors**.
- Goal: Learn a policy that **generalizes** to task instances requiring **an arbitrary number of repetitions**.
- Neural networks **do not** generalize as desired (see below).

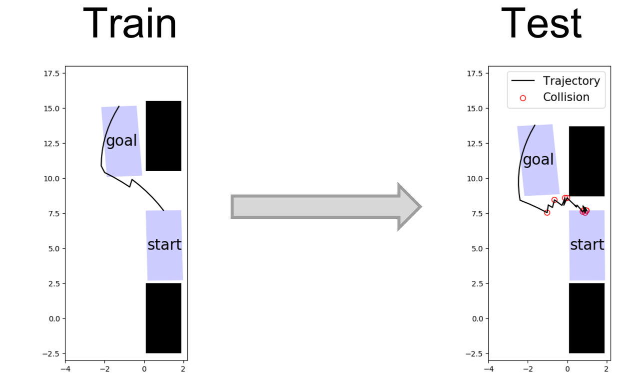


Fig. – The blue car (the agent) is parked between two stationary black cars. The goal is to drive out of the parked spot to an adjacent lane while avoiding collisions.
As we can see here, while the neural network policy solves the training tasks, it leads to collisions on the test task.

Programmatic Policies

- In our approach, we learn **programmatic policies** in the form of **state machines** that can **inductively generalize**.

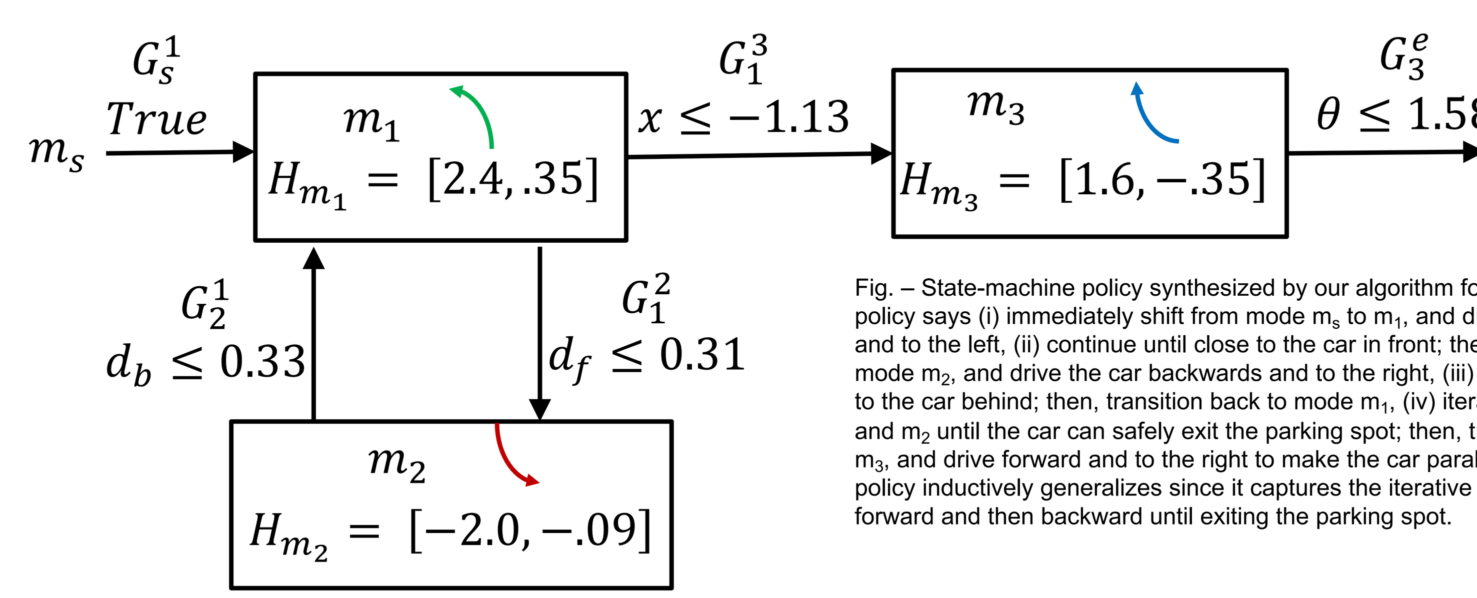
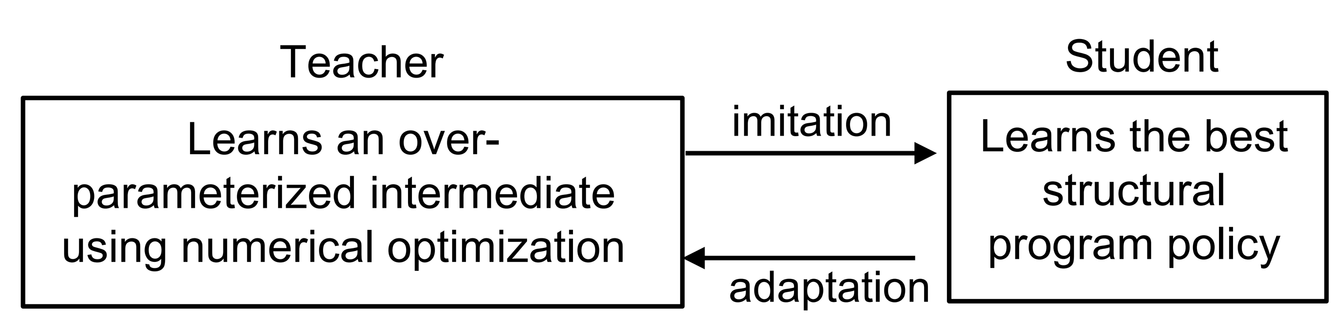


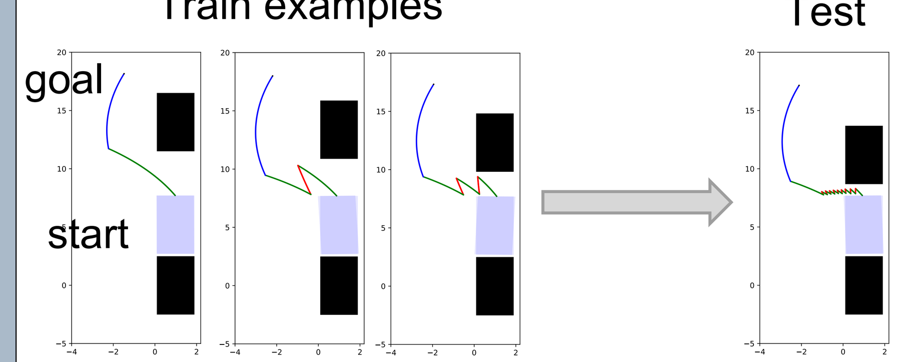
Fig. – State-machine policy synthesized by our algorithm for the car task. This policy says (i) immediately shift from mode m_s to m_1, and drive the car forward and to the left, (ii) continue until close to the car in front; then, transition to mode m_2, and drive the car backwards and to the right, (iii) continue until close to the car behind; then, transition back to mode m_1, (iv) iterate between m_1 and m_2 until the car can safely exit the parking spot; then, transition to mode m_3, and drive forward and to the right to make the car parallel to the lane. This policy inductively generalizes since it captures the iterative behavior of driving forward and then backward until exiting the parking spot.

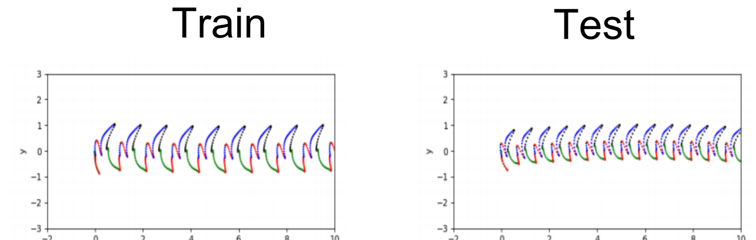
Neurosymbolic learning



Results

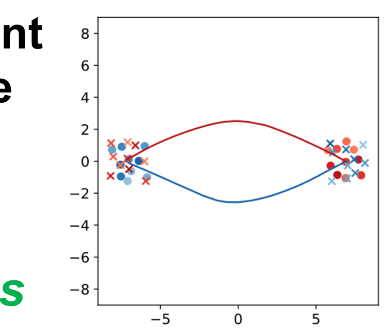
Programs can capture policies that generalize beyond the training distribution.





Communicating in Multi-Agent Systems [2]

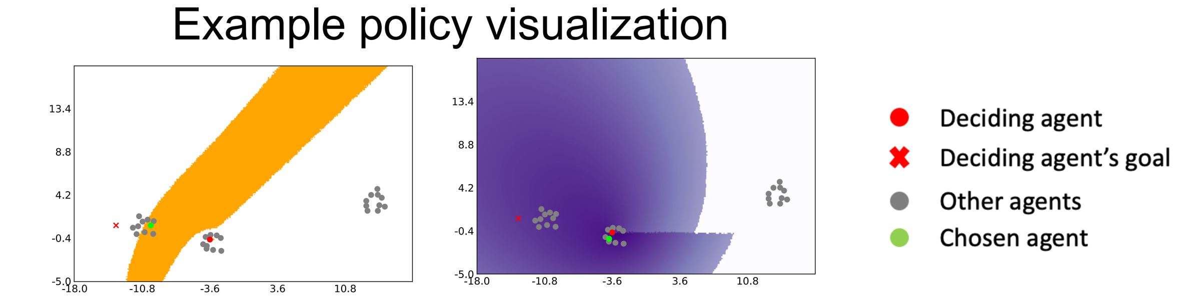
- Goal: Learn **decentralized** policies for **multi-agent** systems where the agents need to **communicate** to avoid collisions.
- ***To improve robustness, it is desirable to reduce the total number of communications***
- **Challenge: It is hard to enforce such combinatorial constraints with neural networks.**



Solution:

- Our approach learns programmatic communication policies
- Uses map, filter, and choice operations to reason over lists of agents

Example policy visualization



Rule 1: The agent chooses to communicate with a random agent in the direction of its goal (orange region)

Rule 2: The red chooses to communicate with a closer agent to avoid collisions (dark blue region)

Key results:

1. Achieves **same** performance as a baseline that uses transformers
2. Significantly **lowers** the number of communications

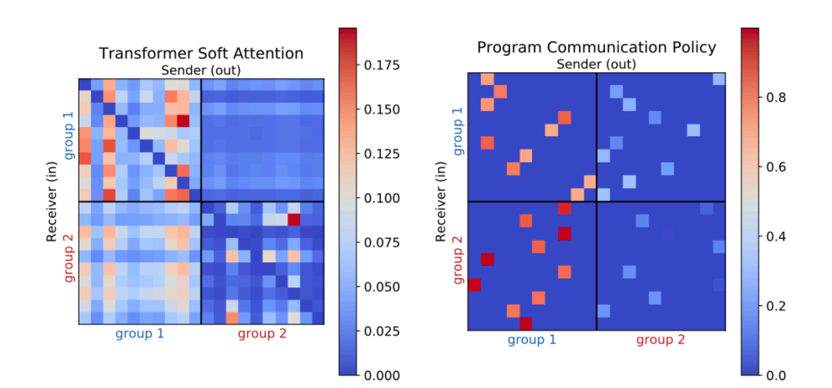


Fig. – Here are the attention plots for both the transformer baseline and our approach. Attention captures how important is a communication between a pair of agents (attn = zero means no communication is required). We can see that the plot on the right has fewer # of communications.

REFERENCES

[1] Jeevana Priya Inala, Osbert Bastani, Zenna Tavares, and Armando Solar-Lezama. Synthesizing programmatic policies that inductively generalize. ICLR 2020
[2] Jeevana Priya Inala, Yichen Yang, James Paulos, Yewen Pu, Osbert Bastani, Vijay Kumar, Martin Rinard, and Armando Solar-Lezama. Neurosymbolic Transformers for Multi-Agent Communication. NeurIPS, 2020.
[3] Thais Campos, Jeevana Priya Inala, Armando Solar-Lezama, and Hadas Kress-Gazit. Task-based design of ad-hoc modular manipulators. ICRA 2019

CONTACT

jinala@csail.mit.edu