# Learning to Plan with Logical Automata

Brandon Araki[1*], Kiran Vodrahalli[2*], Thomas Leech[1,3], Cristian-Ioan Vasile[1], Mark Donahue[3], Daniela Rus[1]
araki@mit.edu [1]MIT; [2]Columbia University; [3]MIT Lincoln Laboratory; *Equal contributors

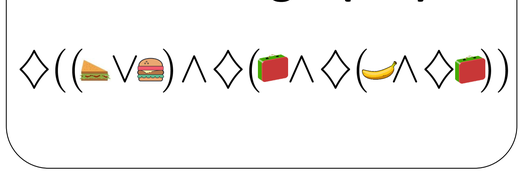
## Overview

**GOAL**: Learn from demonstrations not just a low-level policy but also a high-level policy that is *interpretable* and *manipulable*.
**Interpretable**: The structure and weights of the learned policy are grounded directly in a formal language.
**Manipulable**: A human operator can easily modify the learned policy to perform similar but different policies.

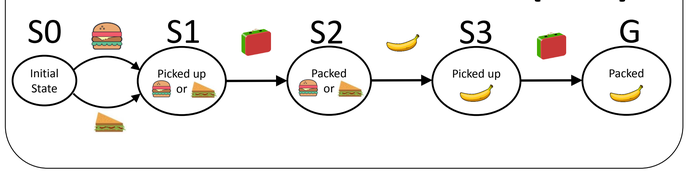
## Representation of Rules

**Rules:**
Pack sandwich or burger;
Then pack banana

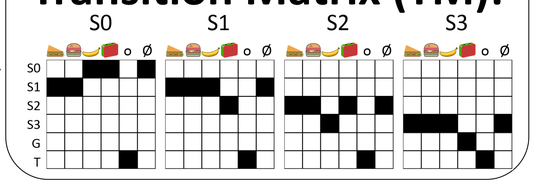
**Formal logic (LTL):**



**Finite State Automaton (FSA):**



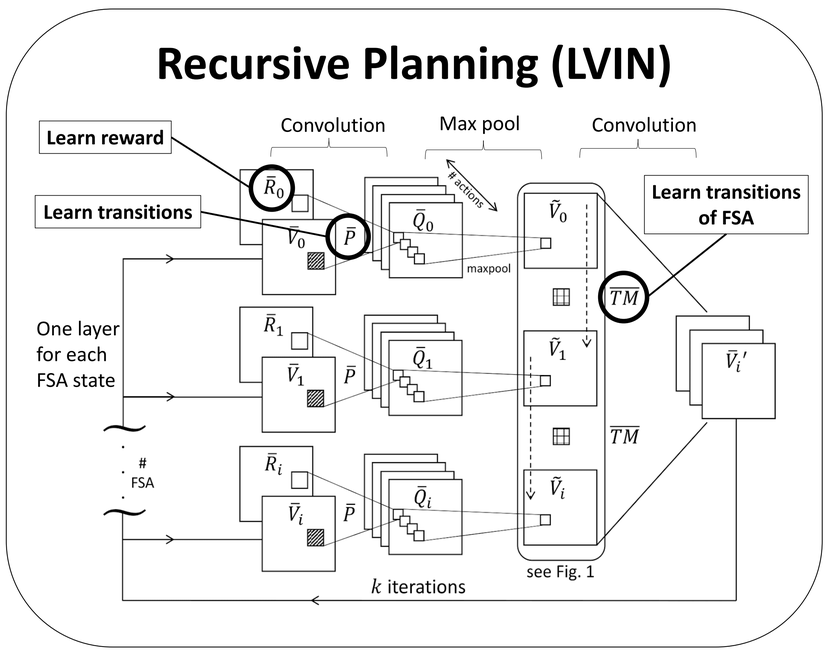

**Transition Matrix (TM):**




## Logic-based Value Iteration Networks (LVIN)

**Inputs**
($x_t$, $y_t$, $f_t$, $map_t$)

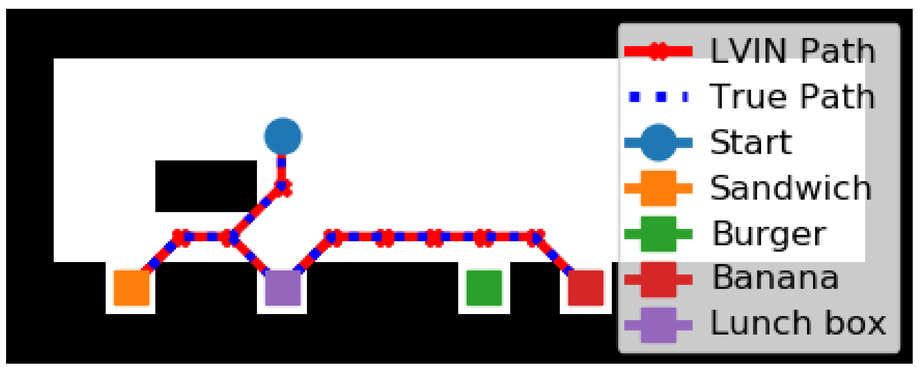
**Recursive Planning (LVIN)**




**Training Data (Expert Trajectories)**
$a_t$, logic $state_t$

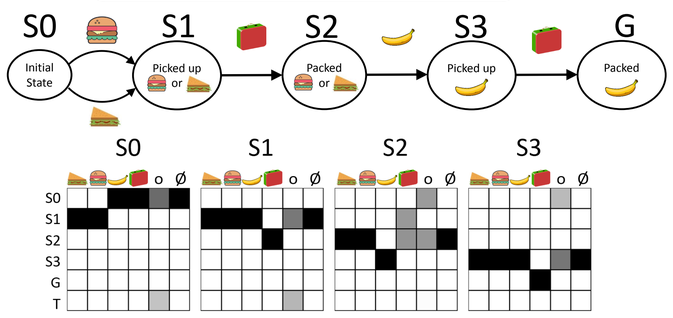
**Run on inputs**
(x0, y0, map)

## Case Study: Lunchbox Packing



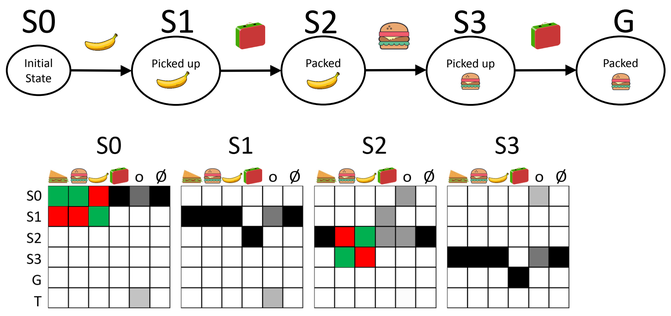

### Learned FSA

Pack sandwich or burger;
Then pack banana

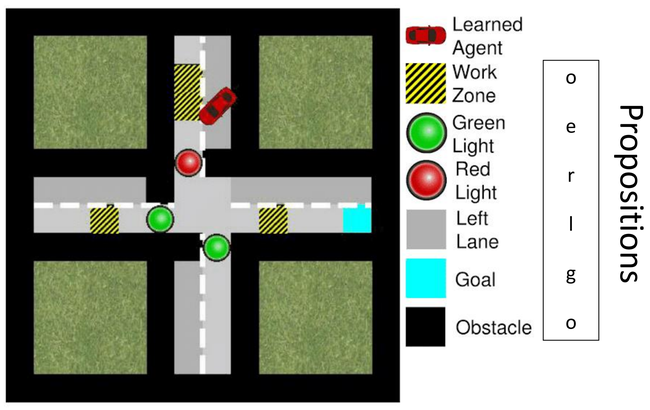



### Modified FSA

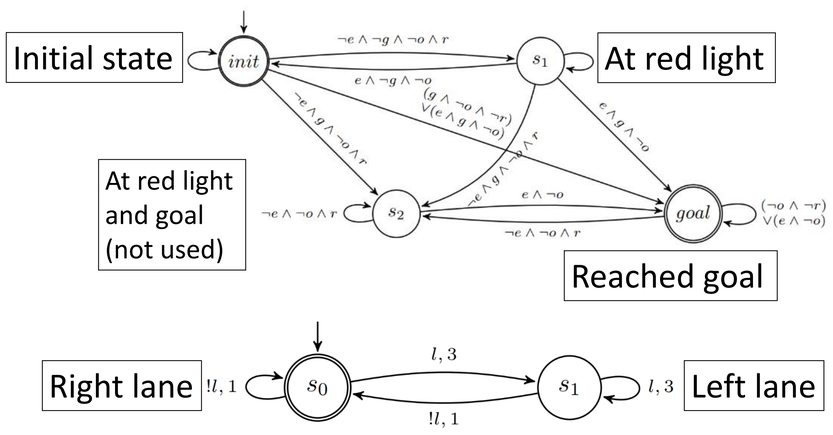
Pack banana;
Then pack only burger



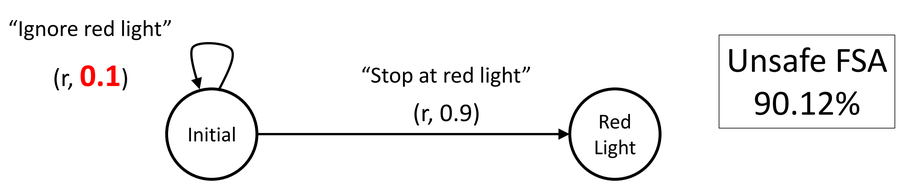

## Case Study: Driving



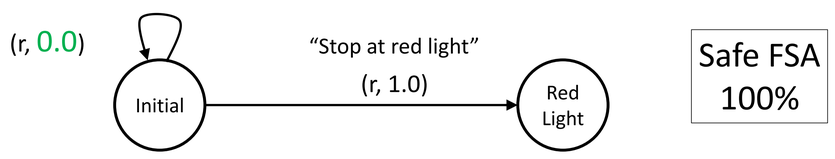

### Ground-truth FSA




### Unsafe Fragment of Learned FSA




### Safe Modified Fragment of Learned FSA




## What Makes LVIN Different?

- Interpret the high level of a hierarchical model as a FSA / logical specification
  - *Interpretable*
- Incorporate the FSA into value iteration so that changes to the FSA result in changes to the policy
  - *Manipulable*
- Interpretable and manipulable policies enable the crafting of *safe* policies