# Deep Bayesian Nonparametric Learning of Rules and Plans from Demonstrations with a Learned Automaton Prior


Brandon Araki[1], Kiran Vodrahalli[2], Thomas Leech[1,3], Cristian-Ioan Vasile[1], Mark Donahue[3], and Daniela Rus[1]

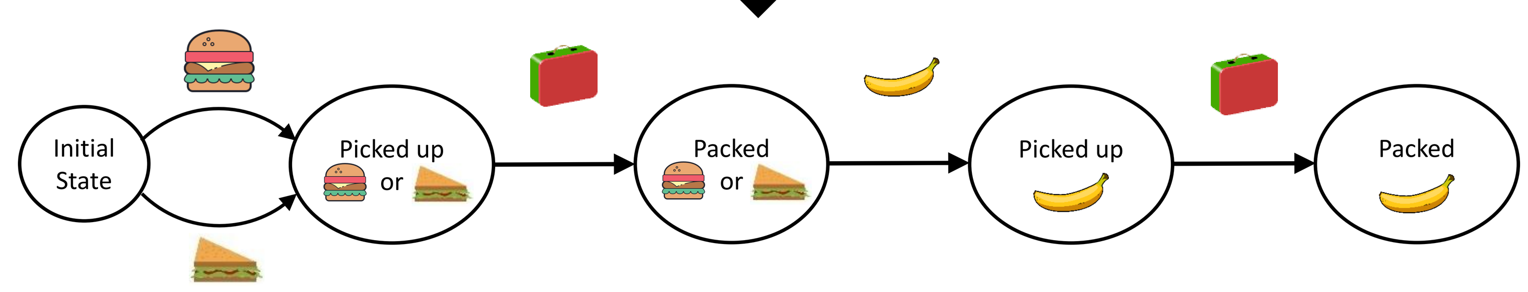
[1]CSAIL, MIT [2]Columbia University [3]MIT Lincoln Laboratory


## Goal: Learn rules and policies for rule-based environments in an **interpretable** and **manipulable** way

## Interpretability

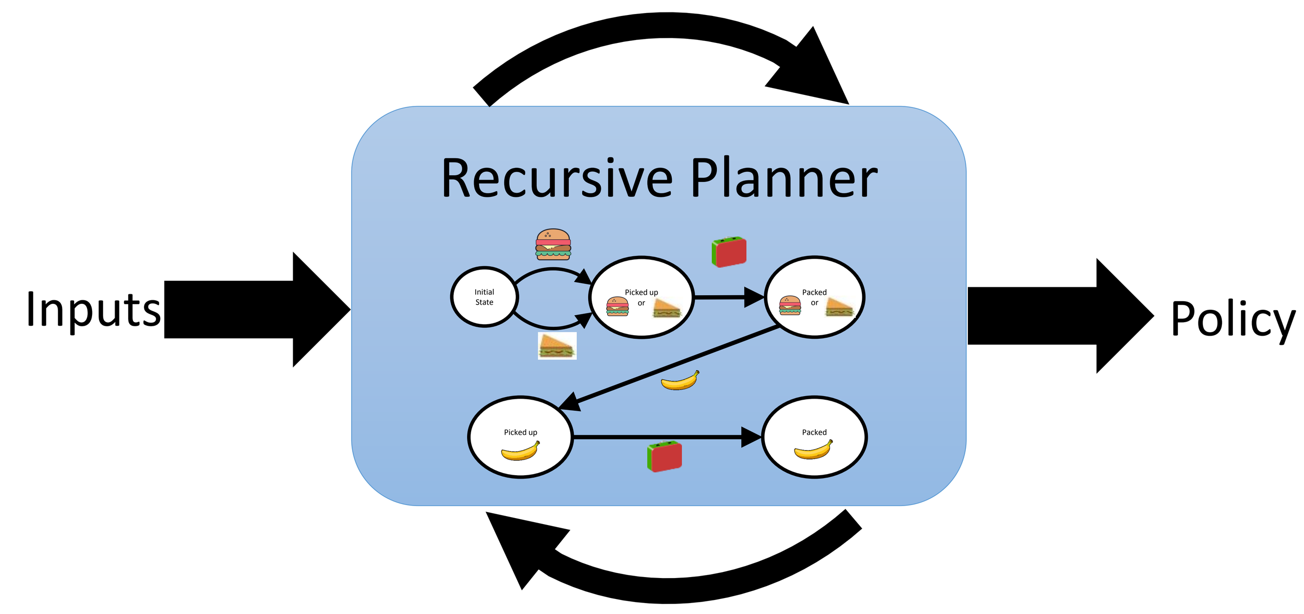
*Pick up a sandwich or hamburger and pack it, then pick up a banana and pack it*

Represent rules as a finite state automaton (FSA)



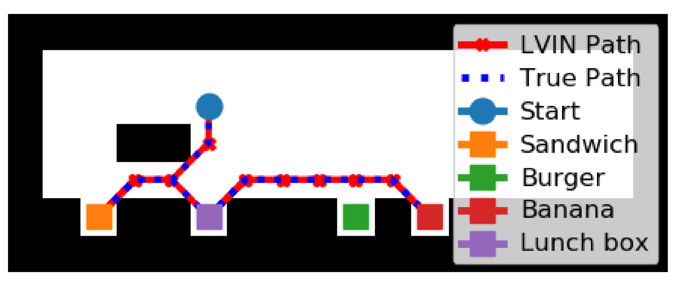

## Manipulability

FSA is an input to a recursive planner, so changes to the FSA result in changes to the policy

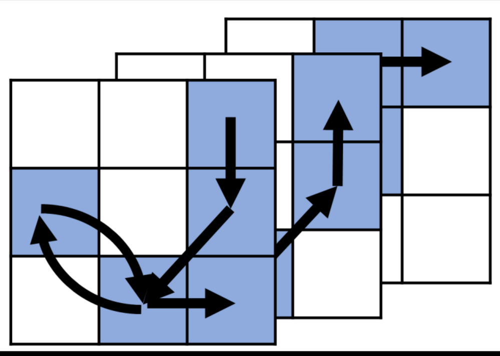



## Data

A low-level environment

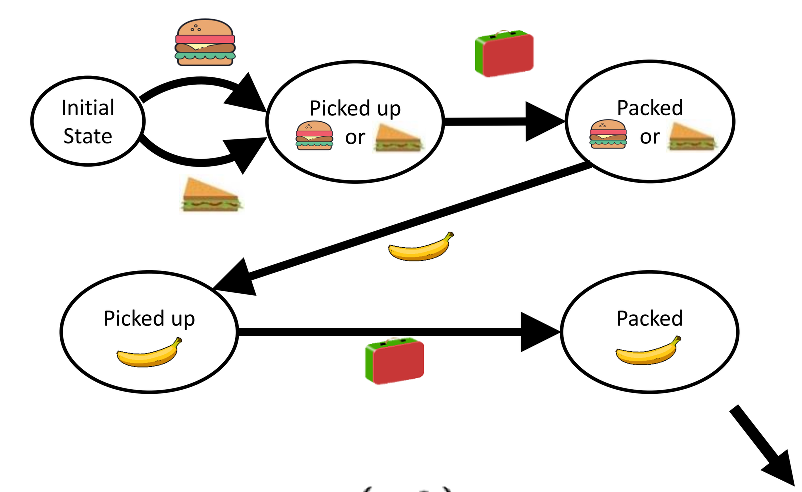



A dataset of trajectories



## Learn

An FSA representation of the rules




A reward function $\mathcal{R}(f) \longrightarrow$ A policy

## Summary of our Approach

**1.** Model the environment as a POMDP

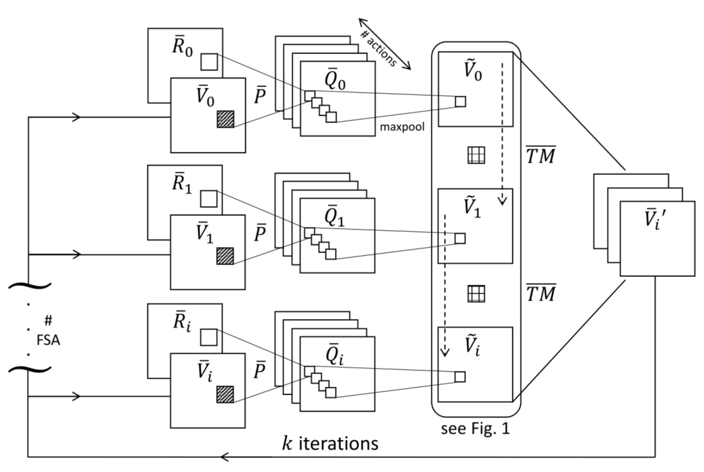
$$(\mathcal{S} \times \mathcal{P} \times \underline{\mathcal{F}}, \mathcal{A}, T \times M \times \underline{TM}, \underline{\mathcal{R}}, \mathcal{S} \times \mathcal{P}, \mathcal{O}, \gamma_d)$$

**2.** Parameterize the policy as a function of an FSA, a reward function, and a low-level environment

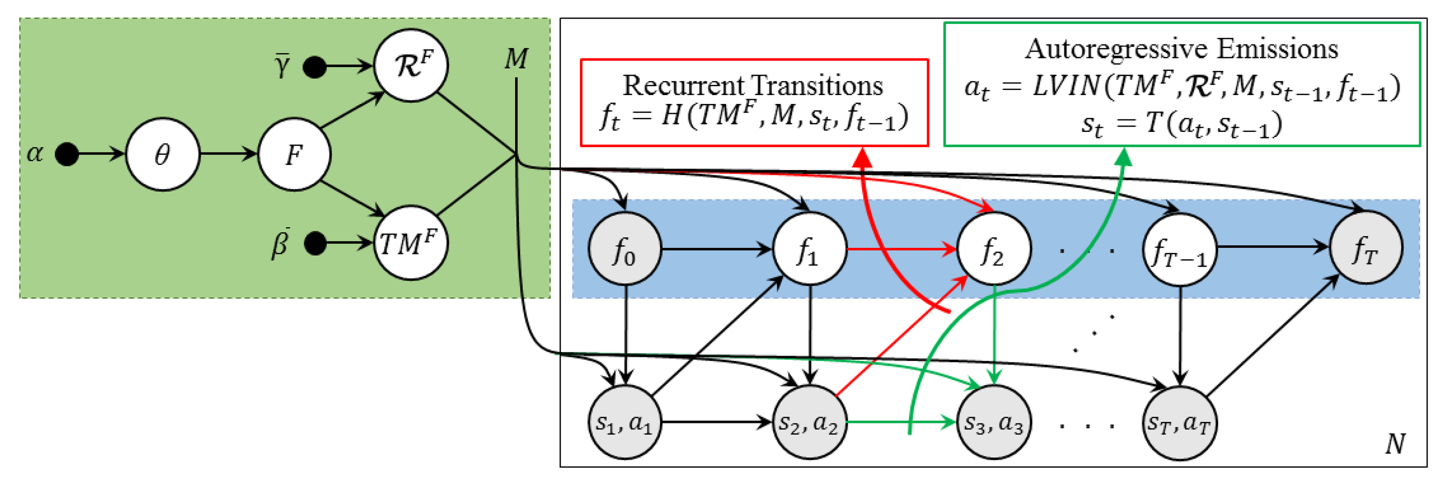
$$Q^{t+1}(s,f,a) \leftarrow R(s,f,a) + \gamma \sum_{s' \in \mathbb{S}} T(s'|s,a) V^t(s',f)$$

$$\widehat{V}^{t+1}(s,f) \leftarrow \max_a Q^{t+1}(s,f,a)$$

$$V^{t+1}(s,f) \leftarrow \sum_{f' \in \mathbb{F}} TM(f'|f,M(s)) \widehat{V}^t(s,f')$$




**3.** Model the policy rollout on the POMDP as an HMM

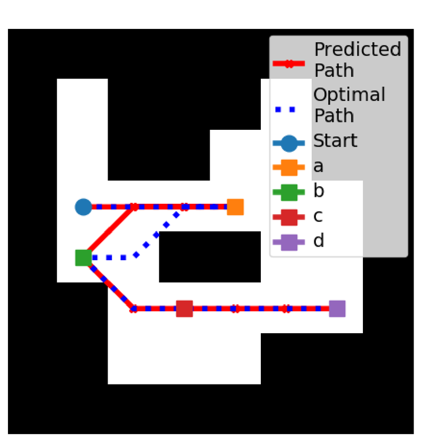



**4.** Use variational inference to infer the latent variables of the HMM including FSA and reward function

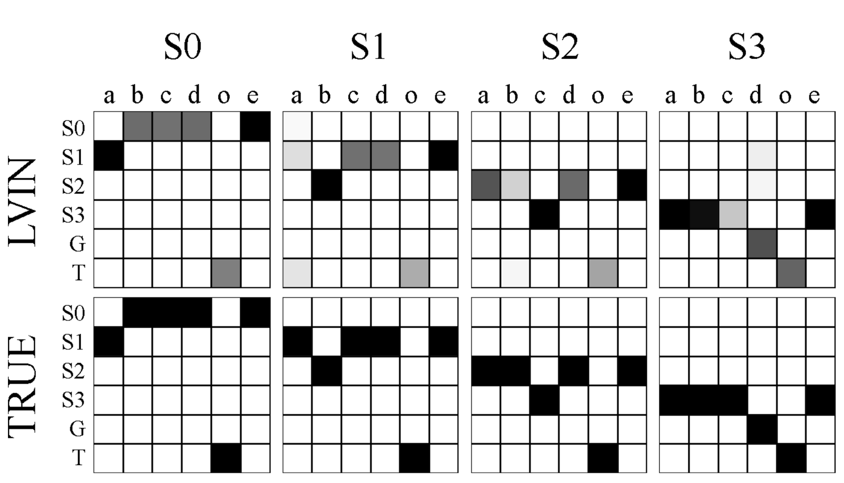
$$p(\overline{TM}, \overline{\mathcal{R}}, \theta | \mathcal{D}, \alpha, \overline{\beta}, \overline{\gamma}) = \frac{p(\mathcal{D}, \overline{TM}, \overline{\mathcal{R}}, \theta | \alpha, \overline{\beta}, \overline{\gamma})}{p(\mathcal{D}|\alpha, \overline{\beta}, \overline{\gamma})}$$

## Experiments and Results

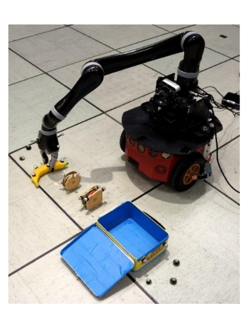
### Gridworld

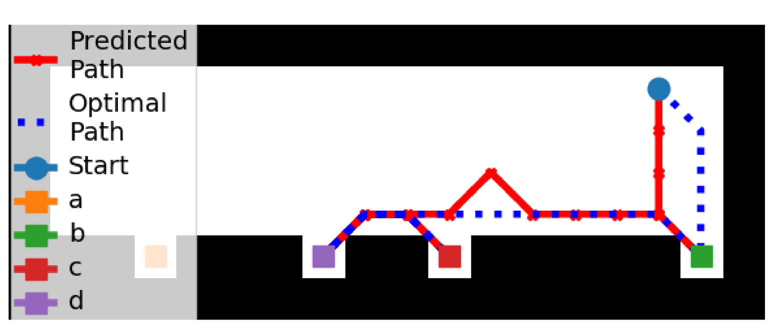



Go to a, then b, then c, then d

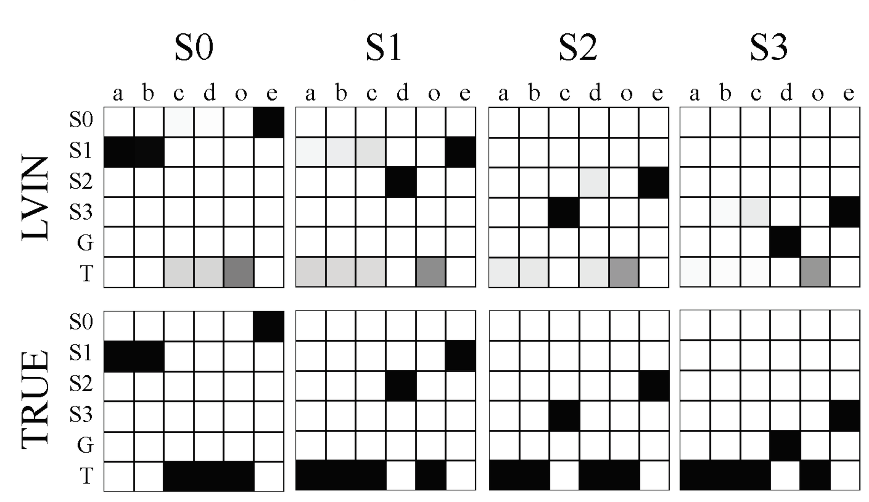



### Lunchbox Packing

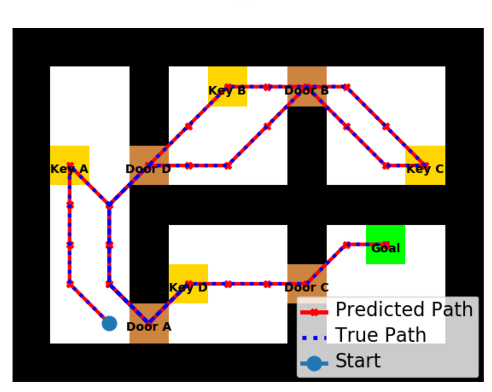




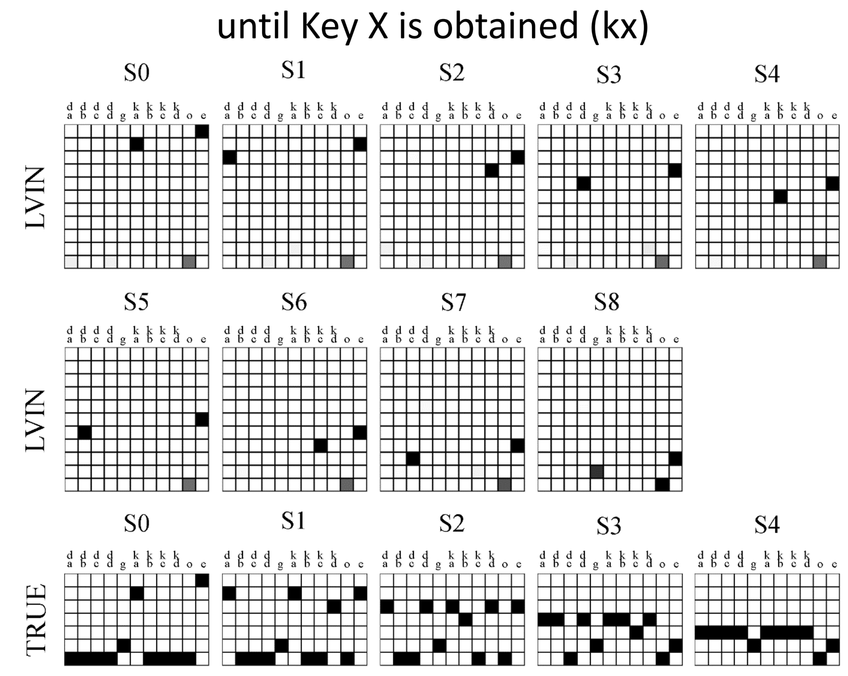

Pick up sandwich (a) or hamburger (b), put it in the lunchbox (d), then pick up banana (c) and put it in the lunchbox (d)



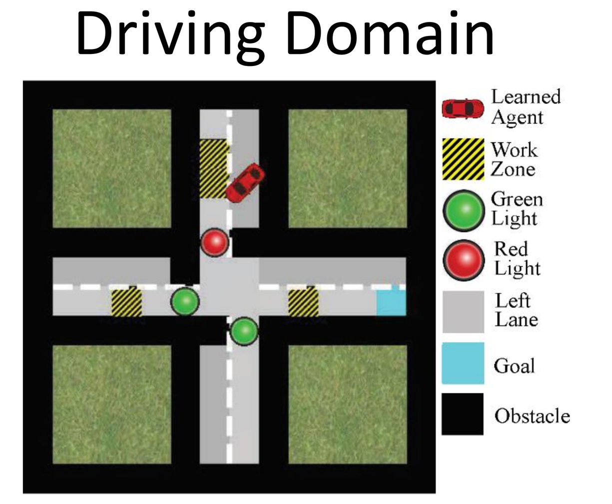

### Dungeon




Go to the goal (g); can't pass through Door x (dx) until Key X is obtained (kx)




### Driving Domain




- Go to the goal and avoid work zones and obstacles
- Stop if there's a red light in front of you and go if there's a green light
- Prefer the right lane to the left lane