



BROWN



# Skill Transfer by Leveraging Temporal Abstractions

29<sup>th</sup> March 2023

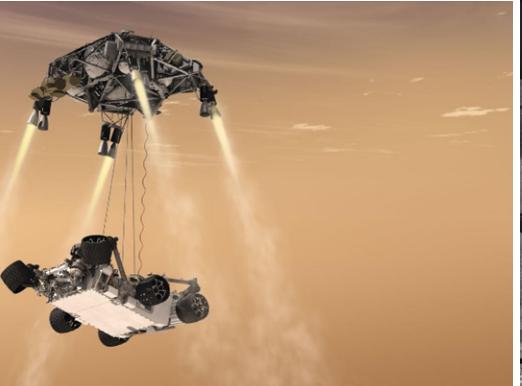
Ankit Shah

Joint work with

Jason Xinyu Liu, Eric Rosen, George Konidaris, Stefanie Tellex

**AAAI Spring Symposium 2023**

# The Promise of Autonomous Systems



Nasa JPL

Space Exploration



DARPA

Disaster Response



KUKA

Flexible Factory Floors

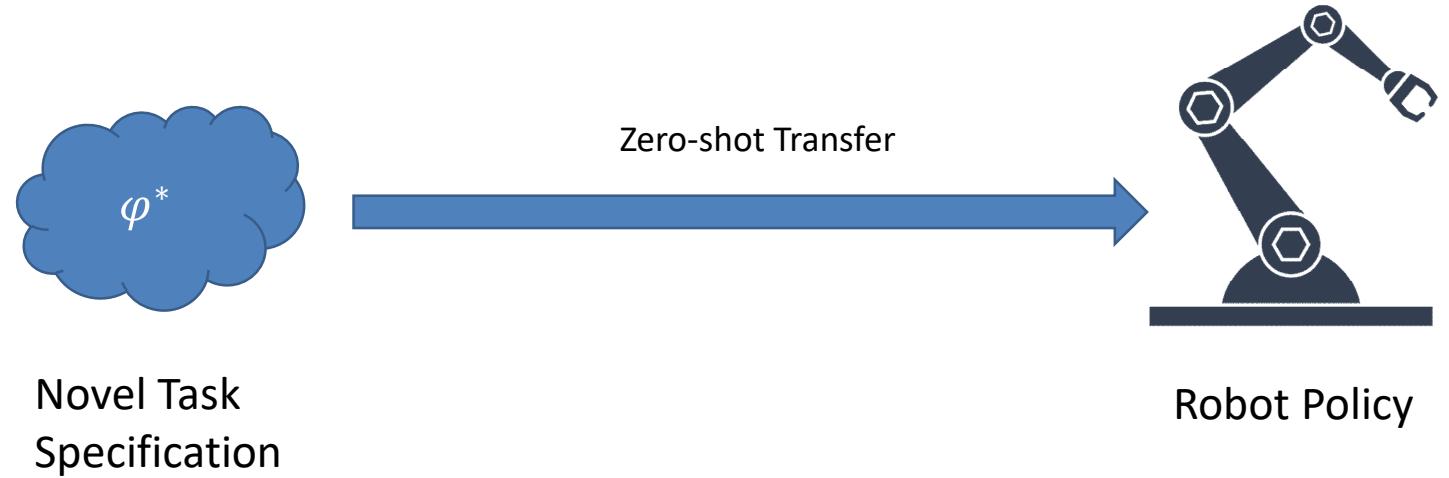


Flights to/from  
uncontrolled airports

One domain but many tasks  
Difficult to anticipate tasks  
Many shared components and actions across tasks



# Automation Deployment: The Vision



One domain but many tasks  
Difficult to anticipate tasks  
Many shared components and actions across tasks

Train on a representative sample of tasks  
Must have capability to generalize to many tasks  
Zero-shot policy through composing policies

# Portable Temporal Skills

Get iron and wood

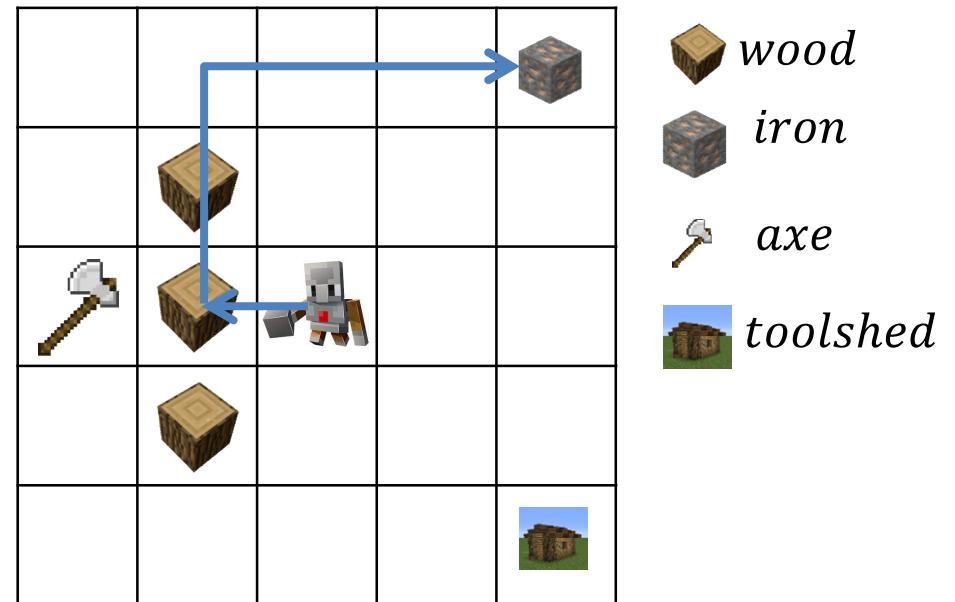
$F \text{ iron} \wedge F \text{ wood}$

Get axe and only then get wood

$F \text{ wood} \wedge !\text{wood} U \text{axe}$

Get axe

$F \text{ axe}$



# Portable Temporal Skills

Get iron and wood

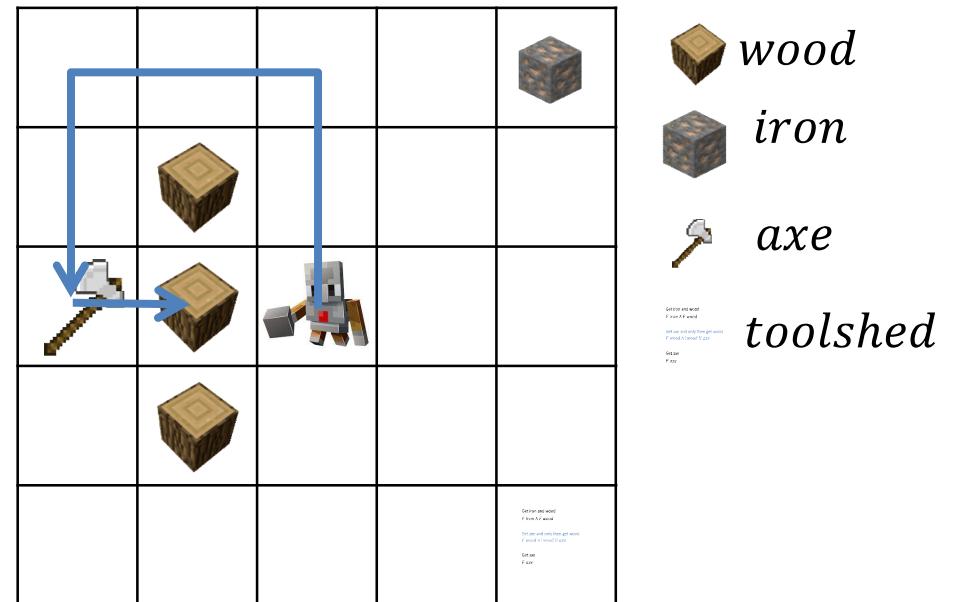
$F \text{ iron} \wedge F \text{ wood}$

Get axe and only then get wood

$F \text{ wood} \wedge !\text{wood} \cup \text{axe}$

Get axe

$F \text{ axe}$



# Portable Temporal Skills

Get iron and wood

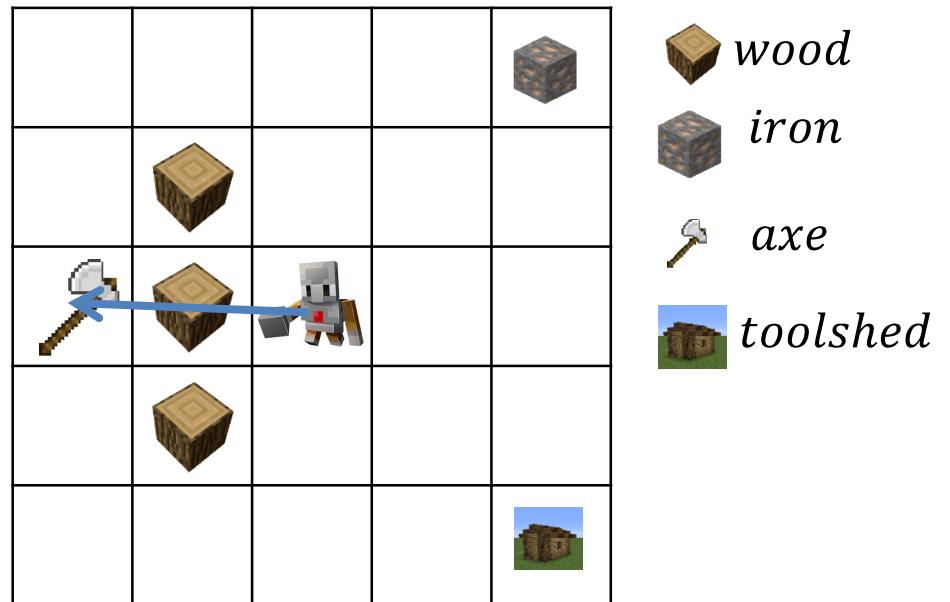
$F \text{ iron} \wedge F \text{ wood}$

Get axe and only then get wood

$F \text{ wood} \wedge !\text{wood} \cup \text{axe}$

Get axe

$F \text{ axe}$



# Portable Temporal Skills

Get axe and get iron

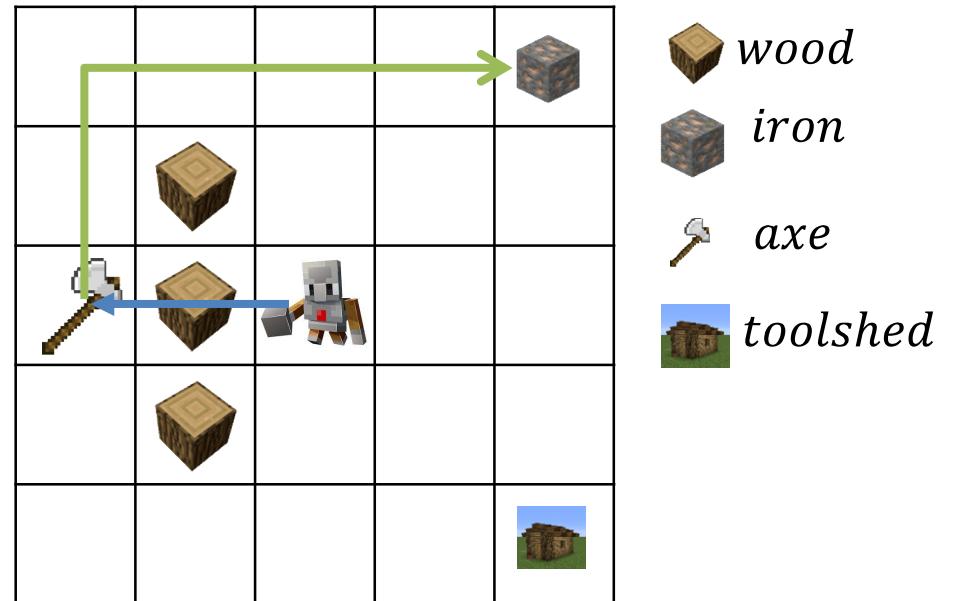
$F \text{ axe} \wedge F \text{ iron}$

Get Iron, then get axe, then get wood

$F(\text{iron} \wedge F(\text{axe} \wedge F \text{ wood}))$

Get iron, then go to axe, but don't visit wood before getting axe

$F(\text{iron} \wedge F \text{ axe}) \wedge (\neg \text{wood} U \text{ axe})$



	<i>wood</i>
	<i>iron</i>
	<i>axe</i>
	<i>toolshed</i>

# Portable Temporal Skills

Get axe and get iron

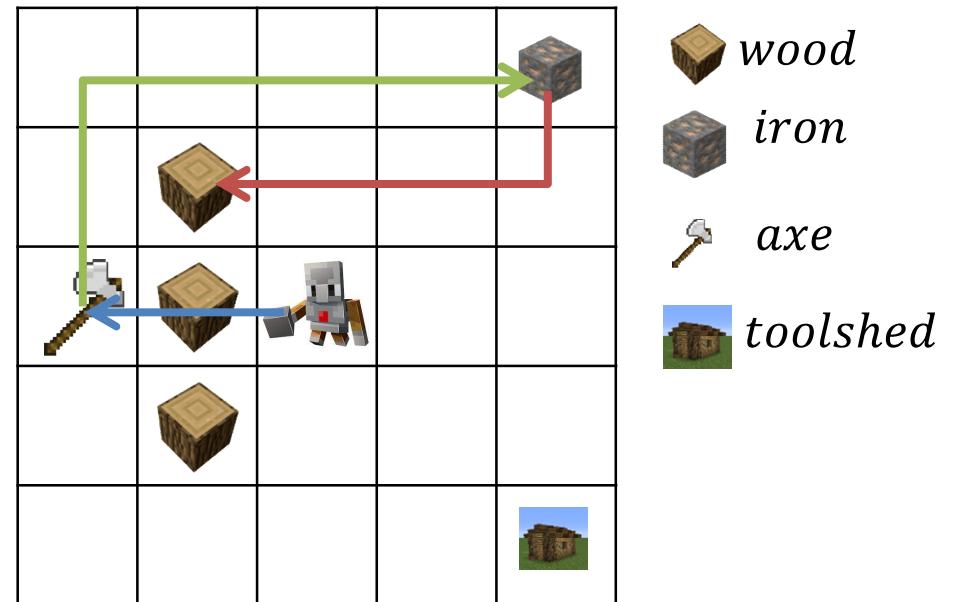
$F \text{ axe} \wedge F \text{ iron}$

Get axe, then get iron, then get wood

$F(\text{axe} \wedge F(\text{iron} \wedge F \text{ wood}))$

Get iron, then go to axe, but don't visit wood before getting axe

$F(\text{iron} \wedge F \text{ axe}) \wedge (\neg \text{wood} U \text{ axe})$



# Portable Temporal Skills

Get axe and get iron

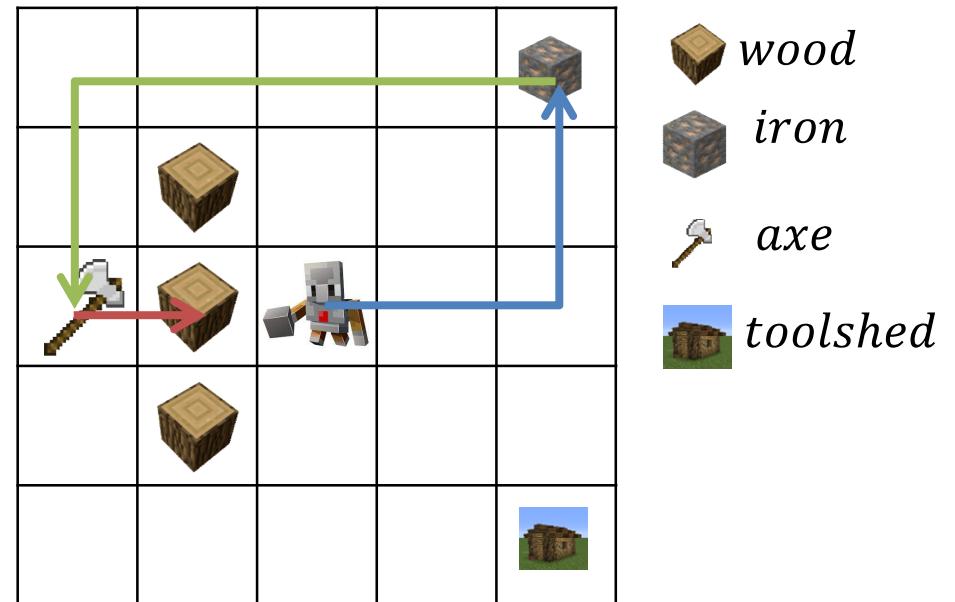
$F \text{ axe} \wedge F \text{ iron}$

Get Iron, then get axe, then get wood

$F(\text{iron} \wedge F(\text{axe} \wedge F \text{ wood}))$

Get iron, then go to axe, but don't visit wood before getting axe

$F(\text{iron} \wedge F \text{ axe}) \wedge (\neg \text{wood} U \text{ axe})$



# Problem Setting

- Input:
  - Set of training tasks  $\Phi_{train} = \{\varphi_1, \varphi_2, \dots\}$
  - Policies for satisfying all training tasks  $\{\pi_{\varphi_1}, \pi_{\varphi_2}, \dots\}$
  - Novel task specification  $\varphi^*$
- Problem 1:
  - Compile policies into set of portable temporal skills
- Problem 2:
  - Compute policy to satisfy  $\varphi^*$  as composition of portable temporal skills



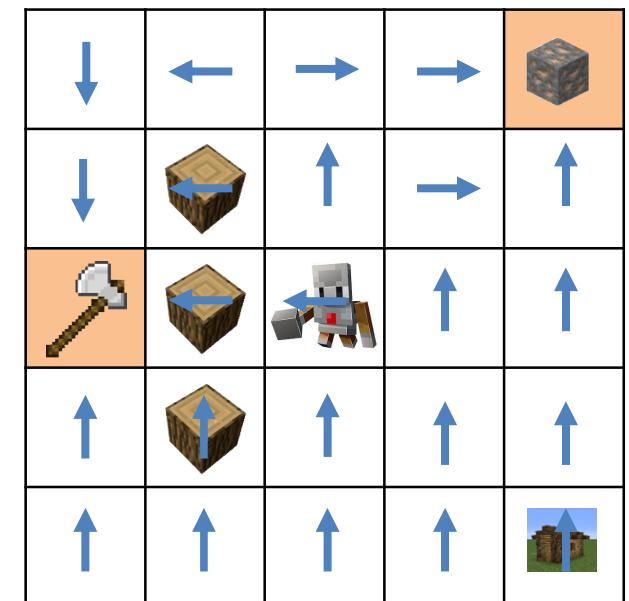
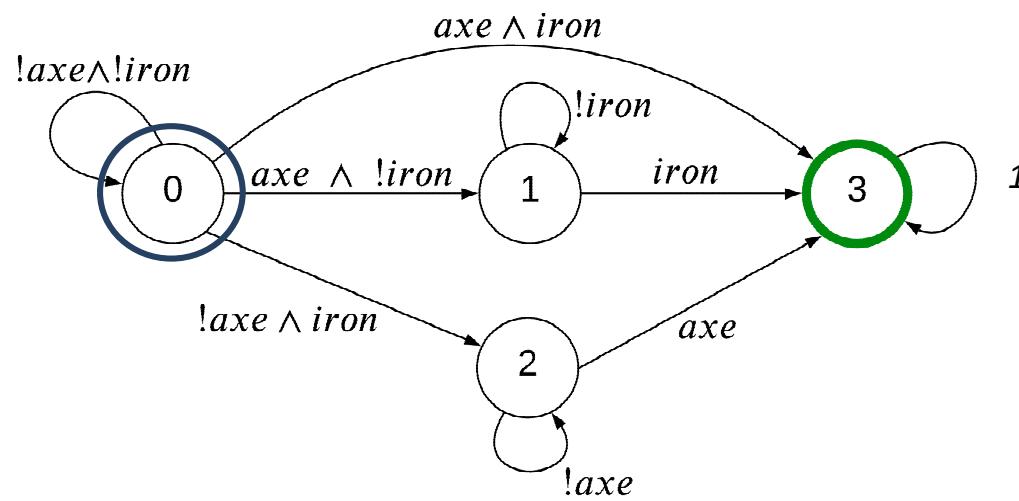
# LTL + RL: An Overview

Construct Markov representation of LTL task

Represent policy as state-centric Markov policies

Get axe and get iron

$F \text{ } axe \wedge F \text{ } iron$



# LTL + RL: An Overview

Construct Markov representation of LTL task

Represent policy as state-centric Markov policies

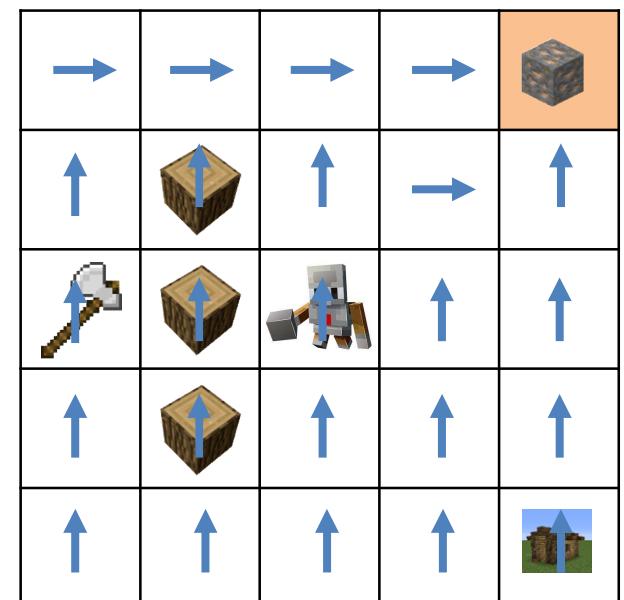
Get axe and get iron

$F \text{ axe} \wedge F \text{ iron}$

Get axe and get iron

$F \text{ axe} \wedge F \text{ iron}$

<sup>1</sup>



# LTL + RL: An Overview

Construct Markov representation of LTL task

Represent policy as state-centric Markov policies

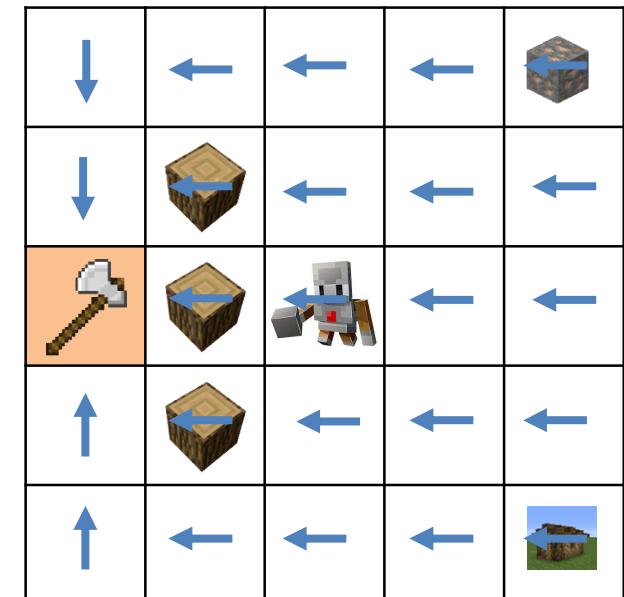
Get axe and get iron

$F \text{axe} \wedge F \text{iron}$

Get axe and get iron

$F \text{axe} \wedge F \text{iron}$

<sub>1</sub>



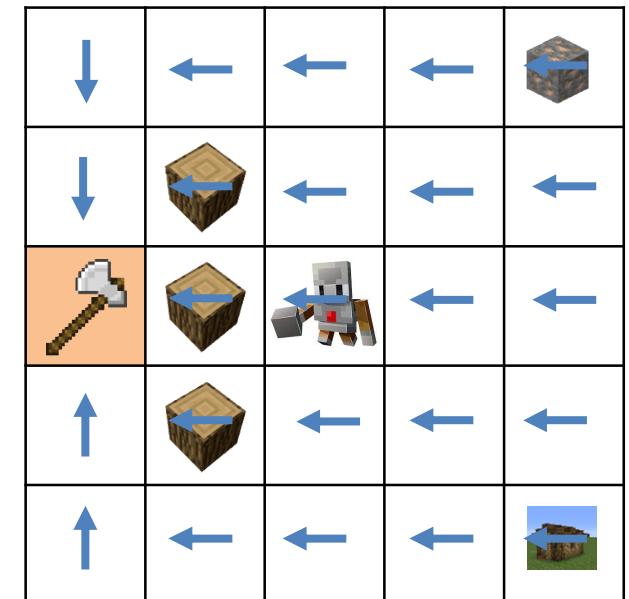
# LTL + RL: Generates a state-centric policy

Get axe and get iron

$F \text{ axe} \wedge F \text{ iron}$

Get axe and get iron

$F \text{ axe} \wedge F \text{ iron}$



# Transition-Centric View of Policies

Each Markov policy ensures a self-edge till it triggers a transition

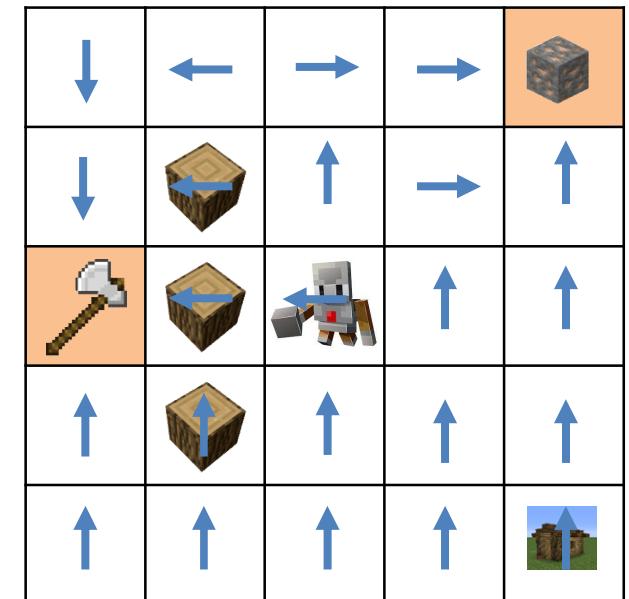
Statespace partitions as per favored transitions

Get axe and get iron

$F \text{ axe} \wedge F \text{ iron}$

Get axe and get iron

$F \text{ axe} \wedge F \text{ iron}$



# Transition-Centric View of Policies

Each Markov policy ensures a self-edge till it triggers a transition

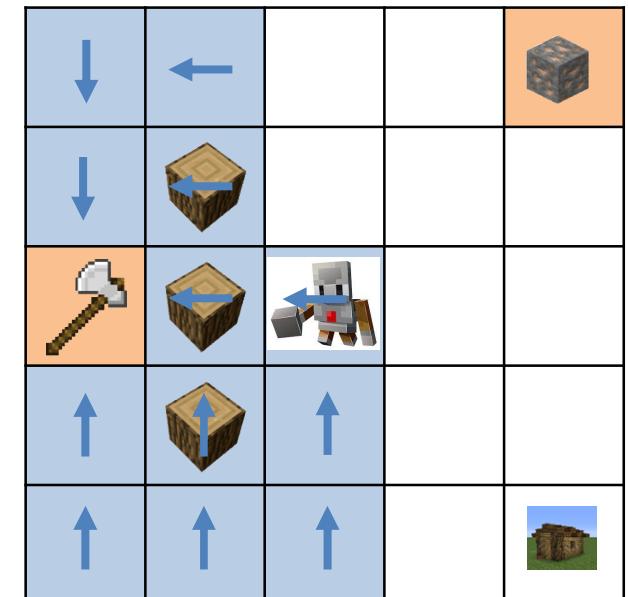
Statespace partitions as per favored transitions

Get axe and get iron

$F \text{ axe} \wedge F \text{ iron}$

Get axe and get iron

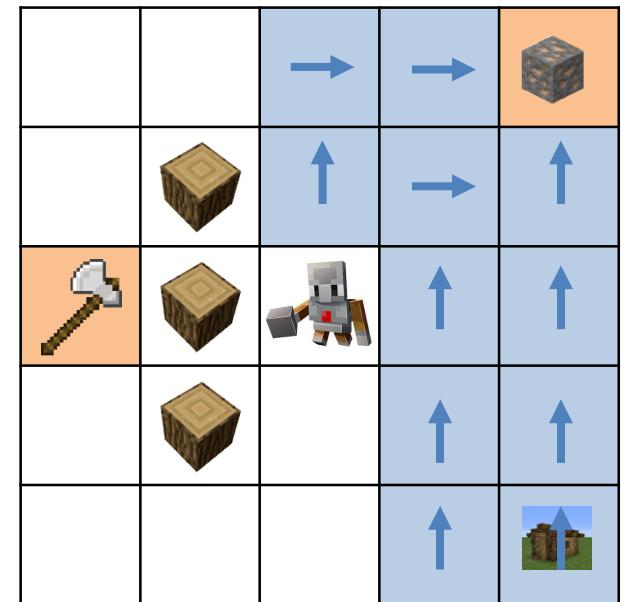
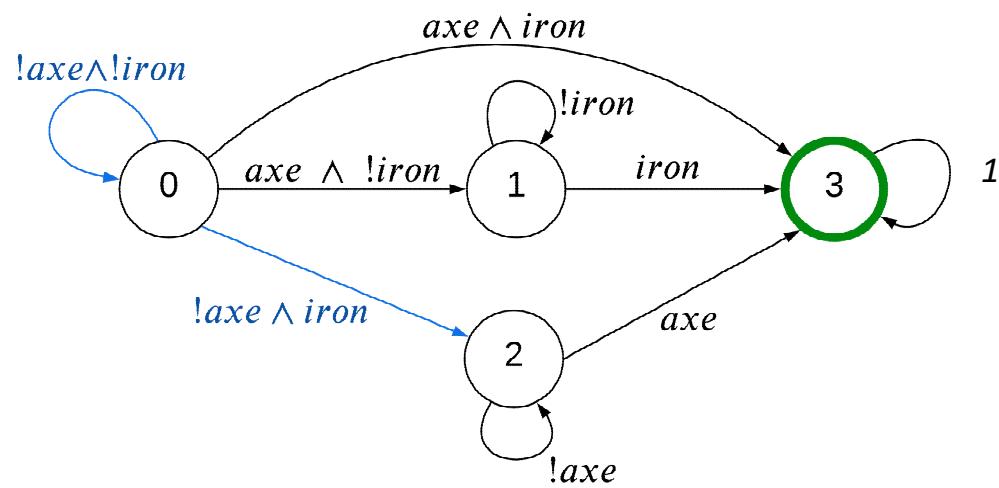
$F \text{ axe} \wedge F \text{ iron}$



# Transition-Centric View of Policies

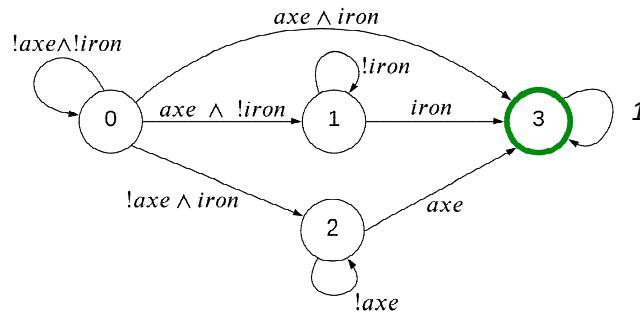
Each Markov policy ensures a self-edge till it triggers a transition  
Statespace partitions as per favored transitions

Get axe and get iron  
 $F \text{ } axe \wedge F \text{ } iron$

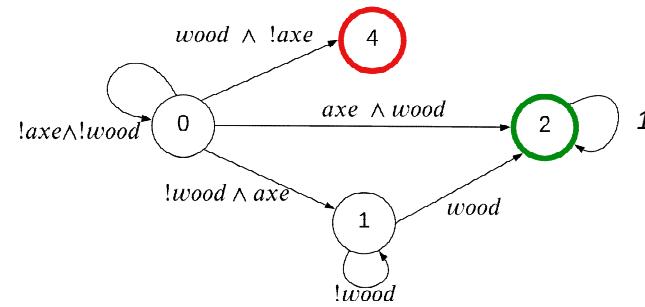


# LTL-Transfer: Zero-Shot Transfer to Novel Tasks

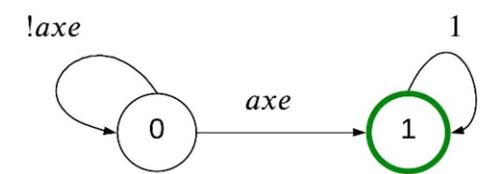
$F \text{ iron} \wedge F \text{ axe}$



$F \text{ wood} \wedge !\text{wood} U \text{ axe}$



$F \text{ axe}$

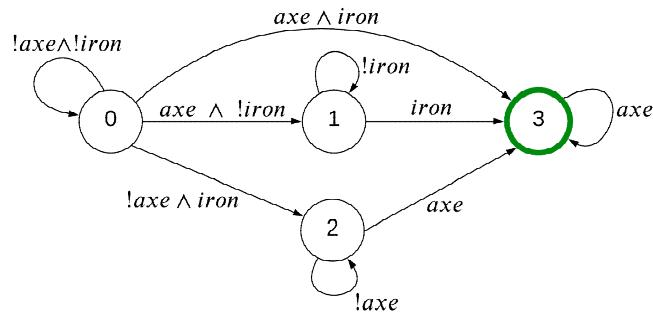


$F(\text{iron} \wedge F \text{ axe}) \wedge (!\text{wood} U \text{ axe})$

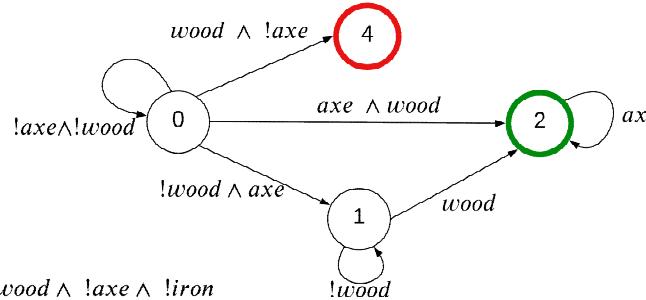


# LTL-Transfer: Zero-Shot Transfer to Novel Tasks

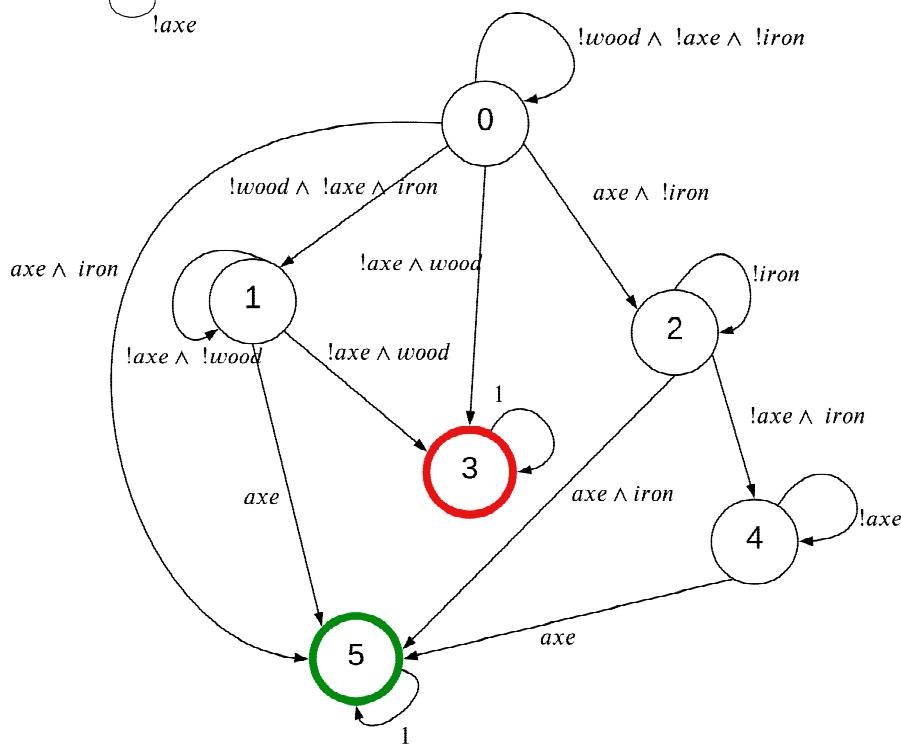
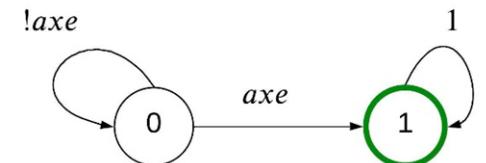
$F \text{ iron} \wedge F \text{ axe}$



$F \text{ wood} \wedge !\text{wood} U \text{ axe}$

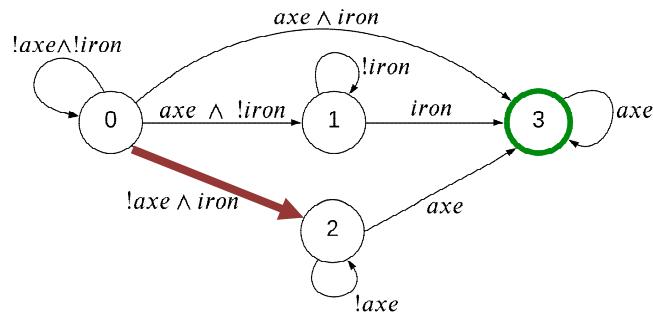


$F \text{ axe}$

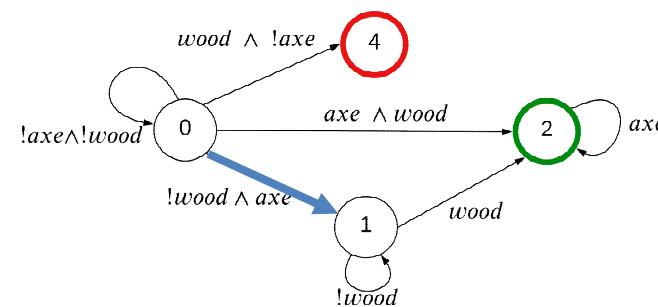


# LTL-Transfer: Zero-Shot Transfer to Novel Tasks

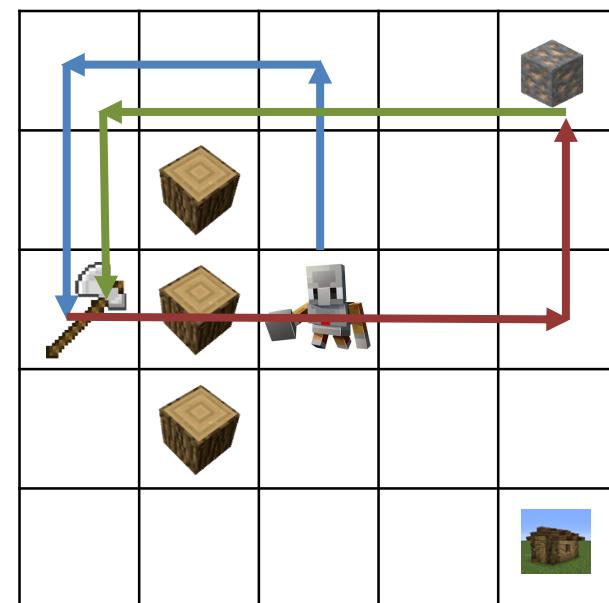
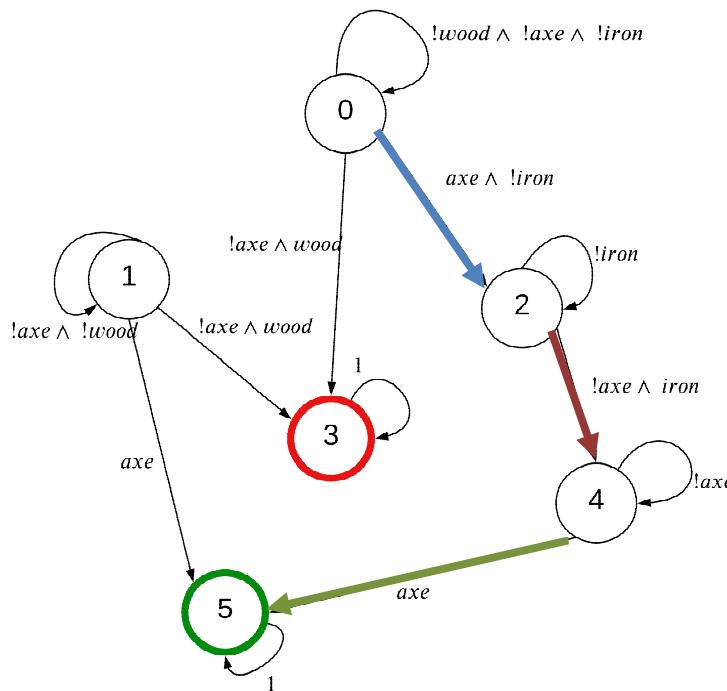
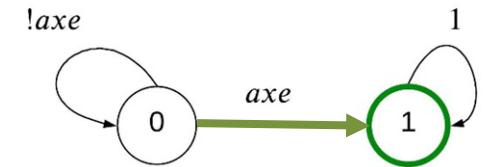
$F \text{ iron} \wedge F \text{ axe}$



$F \text{ wood} \wedge \neg \text{wood} U \text{ axe}$



$F \text{ axe}$

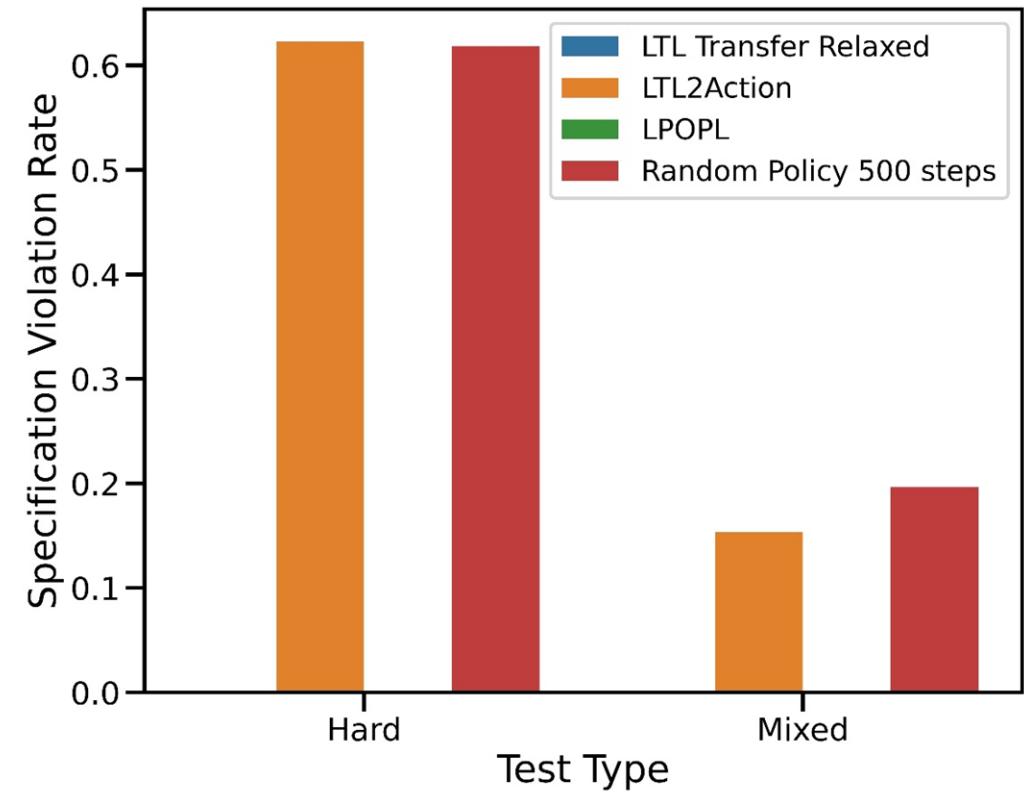
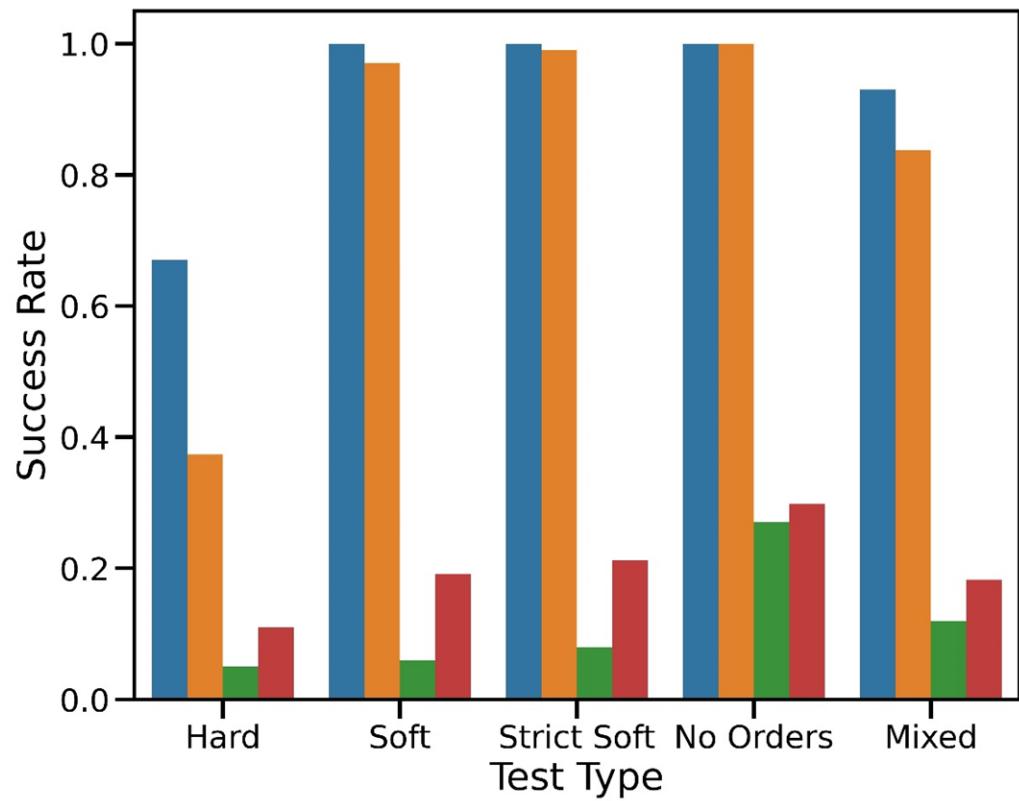


# Evaluations: Setup

- Various distributions over training tasks
- Various distributions of novel tasks
- Matching criterion for selecting eligible transition-centric options
- Binary ordering types:
  - Hard orders:  $a \sqcup b$
  - Soft orders:  $F(a \& Fb)$
  - Strict Soft orders:  $F(a \& XFb)$



# Results



# Deployment on Spot



# Take-aways

- Risk-averse to reusing learnt policies for novel task but suboptimal
- Compatible with few training time specifications
- Evaluation across multiple specification types

