LTL-Transfer: Skill Transfer for Temporally-Extended Task Specifications

Jason Xinyu Liu*, Ankit Shah*, Eric Rosen, George Konidaris, Stefanie Tellex

Brown University

GenPlan @ IJCAI22
July 23, 2022

Motivations

- Autonomous agents deployed in real-world need to perform novel tasks on demand
- Household, Manufacture/Assembly, Disaster Response, Open-World Games

PR2, Kuka, Darpa, Minecraft

Mobile Manipulation Domain

Training Tasks

$\neg desk_a \; \mathbf{U} \; book \wedge \mathbf{F} desk_a$

first pick up book, then deliver it to desk a

$\neg desk_b \; \mathbf{U} \; juice \wedge \mathbf{F} desk_b$

first pick up juice, then deliver it to desk b

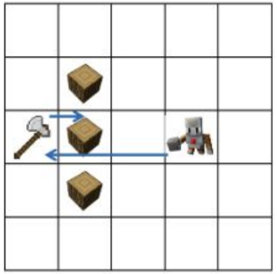
Test Tasks

$\mathbf{F}(book \wedge \mathbf{F}(desk_a \wedge \mathbf{F}(juice \wedge \mathbf{F} desk_a)))$

pick up book, deliver it to desk a,
pick up juice, deliver it to desk b

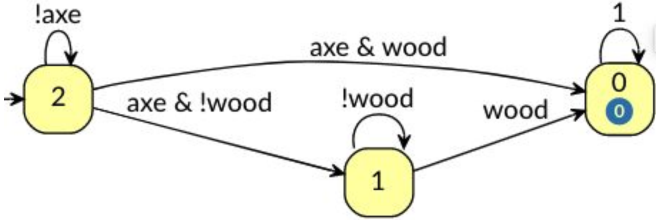
Preliminaries

- **Markov Decision Process** (MDP) to model the environment
  - MDP/R: $\mathcal{M}_\mathcal{S} = \langle \mathcal{S}, \mathcal{A}, T_\mathcal{S} \rangle$
- **Linear Temporal Logic** (LTL) to specify tasks
  - Syntax: $\varphi := p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \mathbf{X}\varphi \mid \varphi_1 \,\mathbf{U}\, \varphi_2 \mid \mathbf{F}\varphi$
- **Reward Machines** (RM)
  - RM: $\mathcal{M}_\varphi = \langle \mathcal{Q}_\varphi, q_{0,\varphi}, \mathcal{Q}_{term,\varphi}, \varphi, T_\varphi, R_\varphi \rangle$
- **Options** $o = \langle \mathcal{I}, \beta, \pi \rangle$



$$\mathbf{F}(\, axe \;\wedge\; \mathbf{F}\; wood)$$

pick up axe, then cut down wood



Camacho et al. 19 LTL and Beyond: Formal Languages for Reward Function Specification in Reinforcement Learning.

Problem Definition

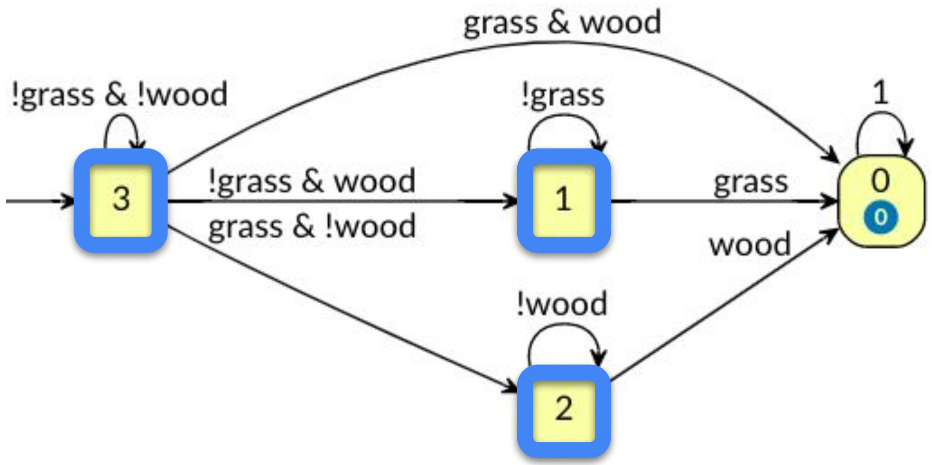
- Input
  - Environment model: $\mathcal{M}_{\mathcal{S}} = \langle \mathcal{S}, \mathcal{A}, T_{\mathcal{S}} \rangle$
  - A set of training tasks: $\Phi_{train} = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$
  - A novel test task: $\varphi_{test} \notin \Phi_{train}$
  - LTL tasks translated to reward machines
- Output
  - State-centric options: $o_{e_{\text{self}}} = \langle \mathcal{S}, \beta_{e_{\text{self}}}, \pi \rangle$
  - Transition-centric options: $o_{(e_{\text{self}}, e_{\text{out}})} = \langle e_{\text{self}}, e_{\text{out}}, \mathcal{S}, f_{e_{\text{out}}}, \beta_{e_{\text{self}}}, \pi \rangle$
  - Plan that solves the novel test task

Algorithm: LPOPL

- Learn a policy for each progressed RM state
- Store state-centric options $o_{e_{\text{self}}} = \langle \mathcal{S}, \beta_{e_{\text{self}}}, \pi \rangle$
- LTL-Transfer works for any algorithm that can solve tasks modeled reward machines

$$\mathbf{F}\ wood \wedge \mathbf{F}\ grass$$

collect wood and grass in any order



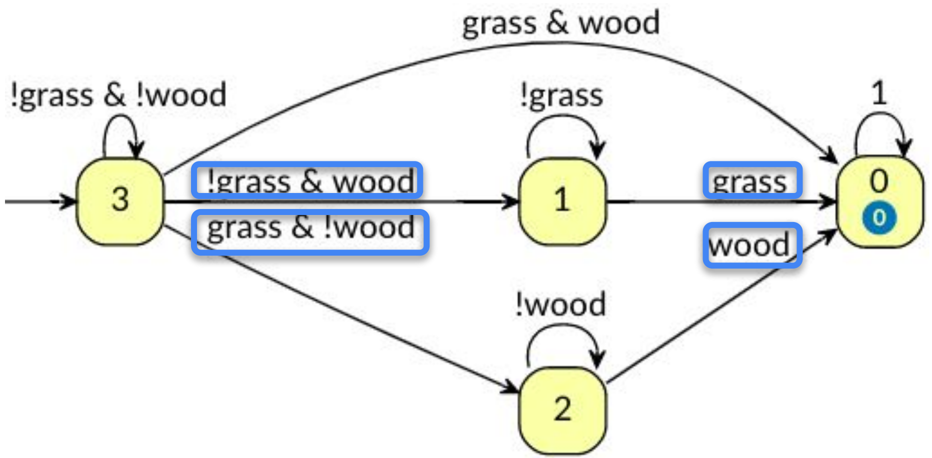
Icarte et al. 18. Teaching Multiple Tasks to an RL Agent using LTL

Algorithm: compile state-centric options to transition-centric options

- Learn initiation set classifiers
  - By rolling out each state-centric policy
- Decompose a state-centric option to possibly multiple transition-centric options
- Store transition-centric options $o_{(e_{\text{self}}, e_{\text{out}})} = \langle \mathcal{S}, \beta_{e_{\text{self}}}, \pi, e_{\text{self}}, e_{\text{out}}, f_{e_{\text{out}}} \rangle$

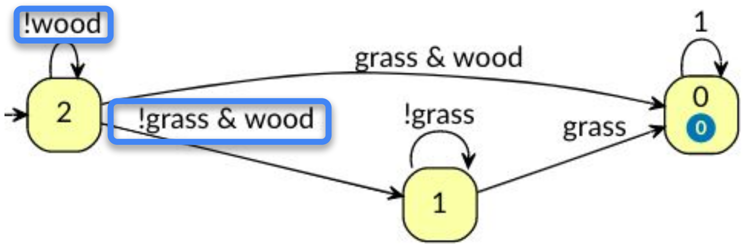
$$\mathbf{F}\, wood \wedge \mathbf{F}\, grass$$

collect wood and grass in any order

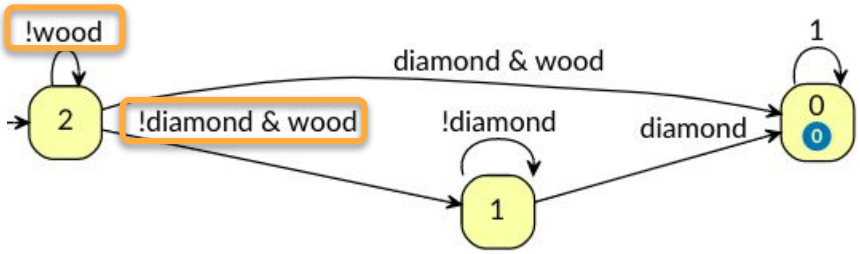


Algorithm: zero-shot transfer to novel tasks

- At each RM state, find all transition-centric options with matching edges
  - edge matching: Constrained and Relaxed
- Select the option with highest success probability by its initiation set classifier
- Execute the option
- Repeat until terminal RM state



Training Task



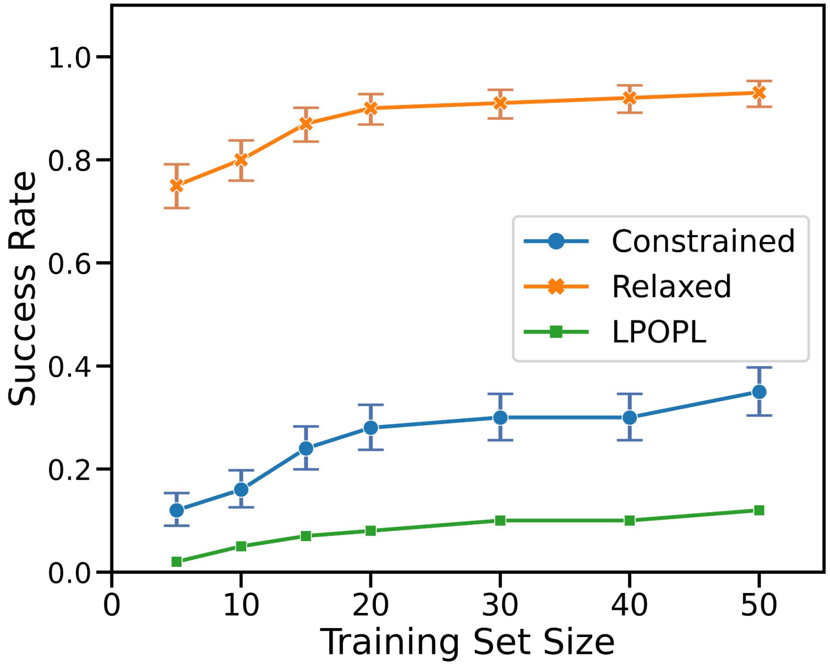
Test Task

Experiments: setup

- **Binary Ordering**
  - defined on every pair of propositions a and b
  - Hard: $\neg b \,\mathbf{U}\, a$
  - Soft: $\mathbf{F}(a \wedge \mathbf{F}b)$
  - Strictly Soft: $\mathbf{F}(a \wedge \mathbf{XF}b)$
- **LTL Types**
  - Hard
  - Soft
  - Strictly Soft
  - Mixed: $(\neg b \,\mathbf{U}\, a) \wedge \mathbf{F}(a \wedge \mathbf{F}c)$
  - No-orders: $\mathbf{F}a \wedge \mathbf{F}b \wedge \mathbf{F}c$
- **5 Training Sets**
  - 50 LTLs per type
- **5 Test Sets**
  - 100 LTLs per type
- **Environment**
  - 4 Minecraft-like maps

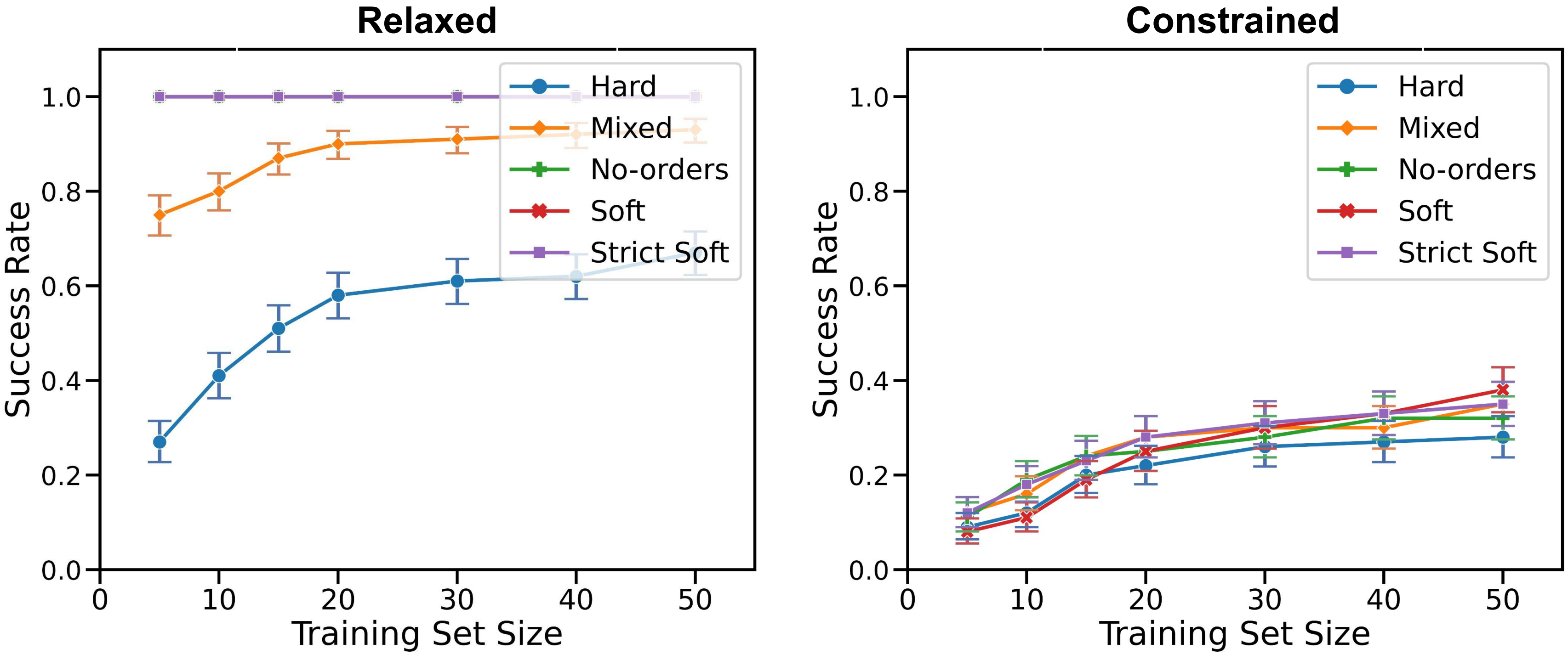
Experiments: results

- **H1:** LTL-Transfer exceeds LPOPL's capability to transfer to novel tasks



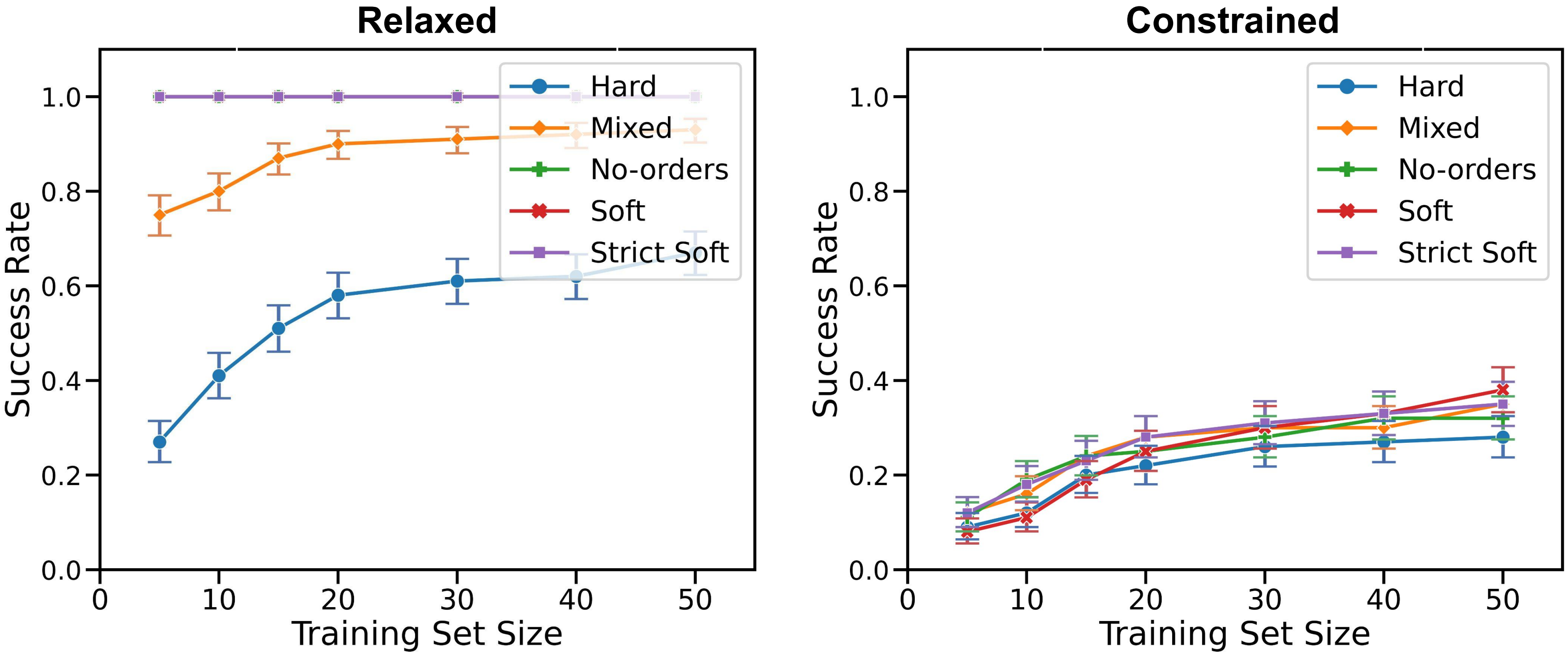
Experiments: results

- **H2:** Relaxed edge matching criterion results in higher success rate than Constrained



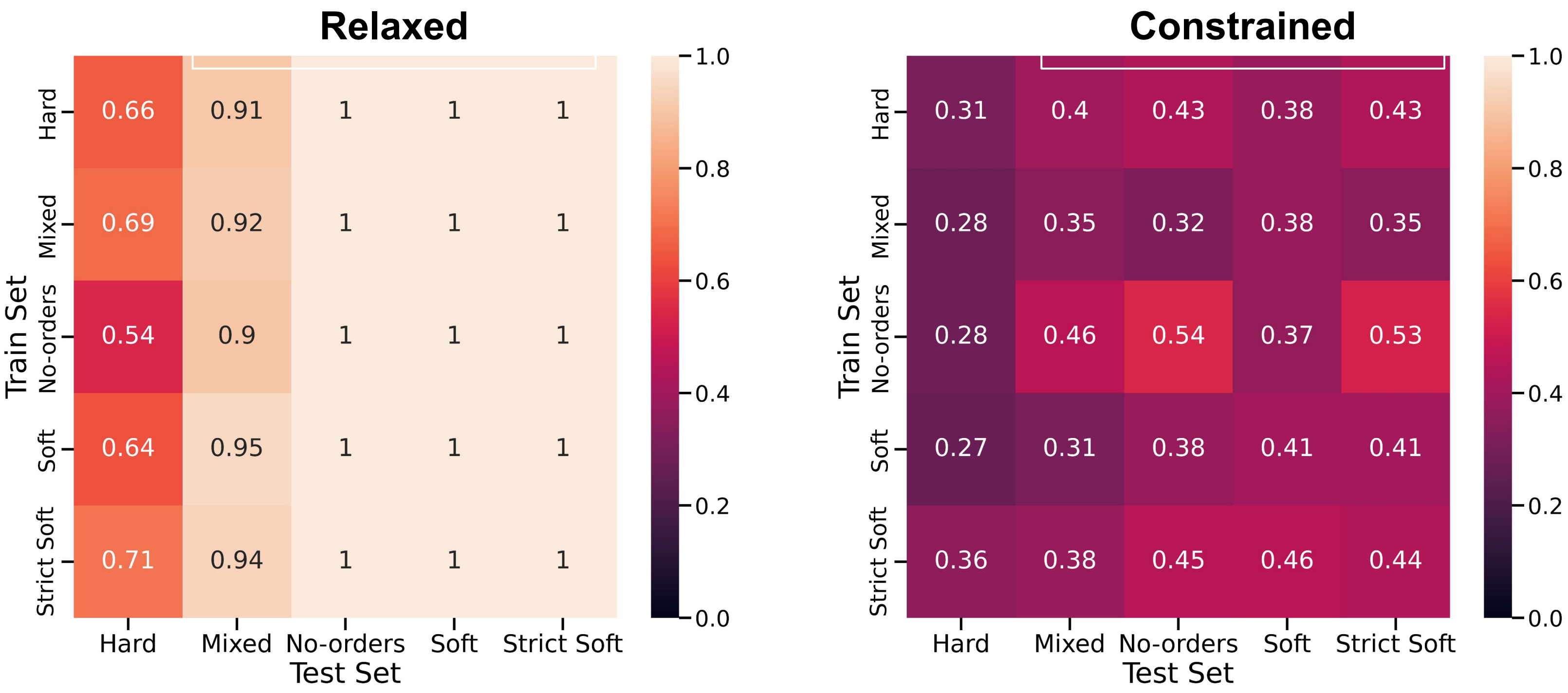
Experiments: results

- **H3:** Certain type of LTL tasks are more difficult to transfer to



Experiments: results

- **H4:** Training on certain type of LTLs leads to greater transfer success rate



LTL-Transfer: Skill Transfer for Temporally-Extended Task Specifications

- LTL-Transfer enables zero-shot transfer to novel LTL tasks by learning portable options

Jason Liu* (xinyu_liu@brown.edu)

Ankit Shah* (ankit_j_shah@brown.edu)

Eric Rose, George Konidaris Stefanie Tellex

Brown bigAI group