RePReL

Integrating Relational Planning and Reinforcement Learning for Effective Abstraction

Sriraam Natarajan

UT DALLAS

Harsha Kokel

UT DALLAS

Balaraman Ravindran

IIT MADRAS

Arjun Manoharan

IIT MADRAS

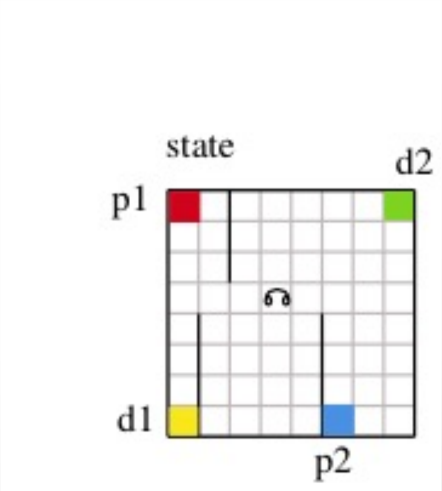
Prasad Tadepalli

Oregon State University

Overview

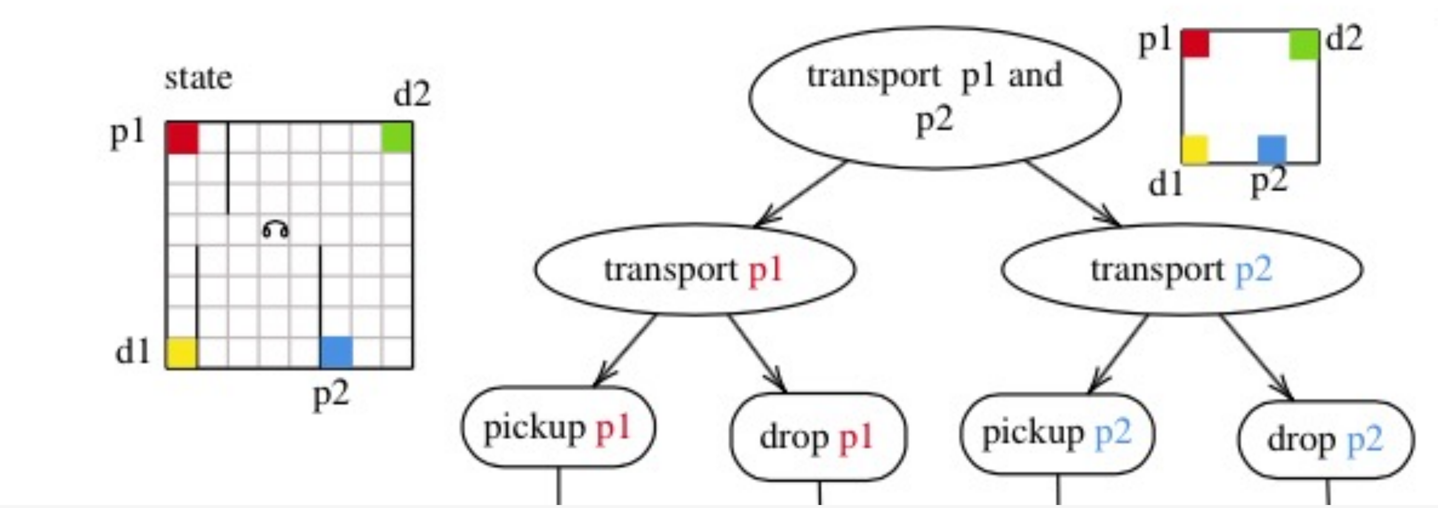
Goal:
Learning to act in relational domains with varying number of tasks and interacting objects

Proposed Solution:
Use a (relational) hierarchical planner to provide abstractions for the RL agents



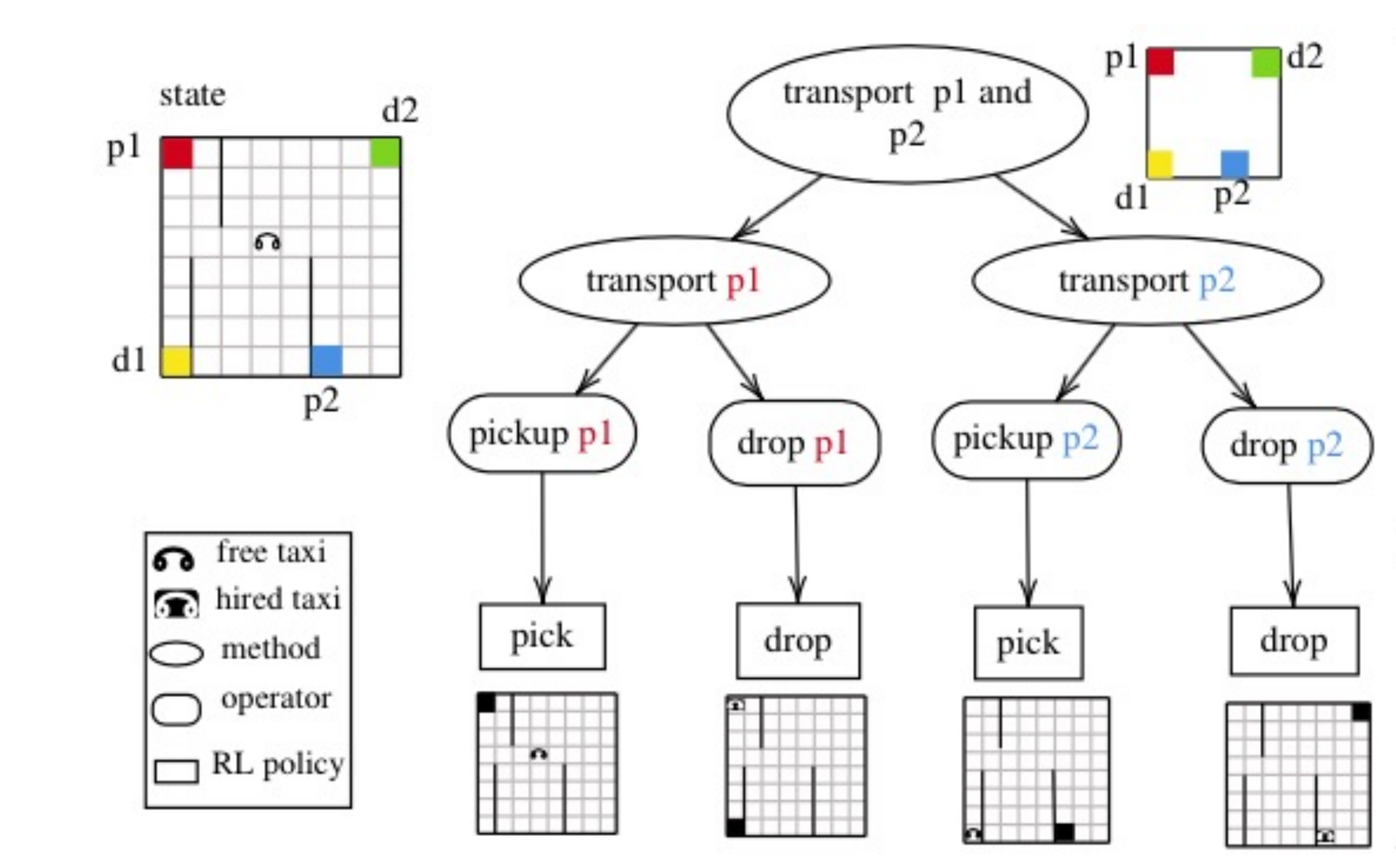
Toy Example

- Extended Taxi domain
- Planner provides sequence of passenger pickup and drop



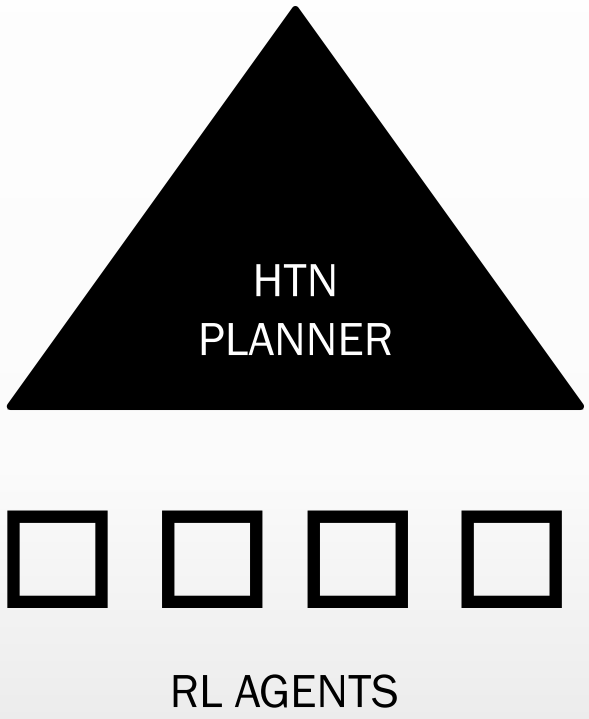
Toy Example

- Extended Taxi domain
- Planner provides sequence of passenger pickup and drop
- Learn driving route



RePReL

- Plan the sequence of operators at high level and learn to execute each operator at lower level
- Advantage:
  - *Compositionality*
  - *Task specific abstract representations*
- Relational MDP which is deterministic and fully observable
- Adapt First Order Conditional Influence statements to specify bisimulation conditions of MDPs for ‘safe’ abstractions.



Abstraction

First Order Conditional Influence (FOCI) statements

if $condition$ then X1 influence X2

Dynamic FOCI statements

$$\textbf{operator} \; : X1 \xrightarrow{+1} X2$$

Natarajan, Tadepalli, Dietterich, and Fern 2008

Abstraction

Given

State

$\{\text{at}(\text{p1}, \text{r}), \text{taxi-at}(13), \text{dest}(\text{p1}, \text{y}), \neg\,\text{at-dest}(\text{p1}), \neg\,\text{in-taxi}(\text{p1}), \text{at}(\text{p2}, \text{b}), \text{dest}(\text{p2}, \text{g}), \neg\,\text{at-dest}(\text{p2}), \neg\,\text{in-taxi}(\text{p1})\}$

subtask

$\langle\ \text{pickup}(P), \{P/p1, L/r\}\ \rangle$

D-FOCI

$\{\text{taxi-at(L1), move(Dir)}\} \xrightarrow{+1} \text{taxi-at(L2)}$

$\{\text{taxi-at(L1), move(Dir)}\} \longrightarrow R$

pickup(P):

$\{\text{taxi-at(L1), at(P, L), in-taxi(P)}\} \xrightarrow{+1} \text{in-taxi(P)}$

pickup(P): $\text{in-taxi(P)} \longrightarrow R_o$

Get

Abstract state

$\{\text{at}(\text{p1}, \text{r}), \text{taxi-at}(13), \neg\,\text{in-taxi}(\text{p1}), \text{move}(\text{Dir})\}$

Abstraction

Given

State: $\{\text{at}(\text{p1}, \text{r}), \text{taxi-at}(13), \text{dest}(\text{p1}, \text{y}), \neg\, \text{at-dest}(\text{p1}), \neg\, \text{in-taxi}(\text{p1}), \text{at}(\text{p2}, \text{b}), \text{dest}(\text{p2}, \text{g}), \neg\, \text{at-dest}(\text{p2}), \neg\, \text{in-taxi}(\text{p1})\}$

subtask: $\langle\ \text{pickup}(P), \{P/p1, L/r\}\ \rangle$

D-FOCI:

$$\{\text{taxi-at(L1), move(Dir)}\} \xrightarrow{+1} \text{taxi-at(L2)}$$
$$\{\text{taxi-at(L1), move(Dir)}\} \longrightarrow R$$
pickup(P):
$$\{\text{taxi-at(L1), at(P, L), in-taxi(P)}\} \xrightarrow{+1} \text{in-taxi(P)}$$
$$\text{pickup(P): in-taxi(P)} \longrightarrow R_o$$

Get

Abstract state: $\{\text{at}(\text{p1}, \text{r}), \text{taxi-at}(13), \neg\, \text{in-taxi}(\text{p1}), \text{move}(\text{Dir})\}$

safe model-agnostic abstraction

(Theorem 1)

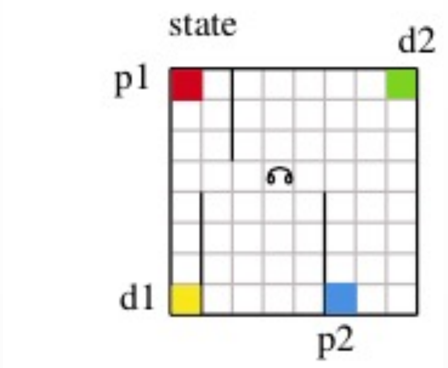
RePReL Learning Algorithm

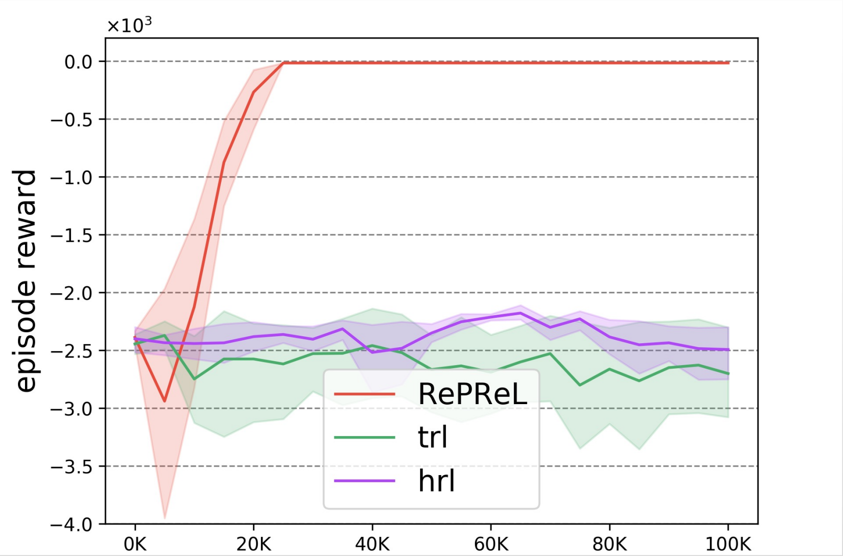
- Get high level plan
- For each sub-task
  - *Get resp. policy* π
  - *Loop till the sub-task is achieved*
    - Get the abstract state s
    - Get action a from the policy π
    - Step in env observe reward $\langle s, a, r, s' \rangle$
    - Update the policy π

*Q learning

Experiments

- Evaluate for
  - *Sample efficiency*
  - *Transfer across task*
  - *Generalization across objects*



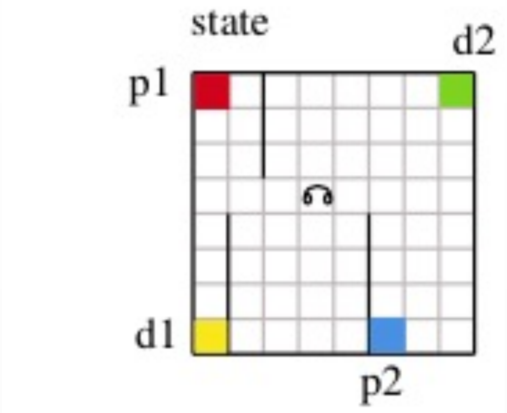


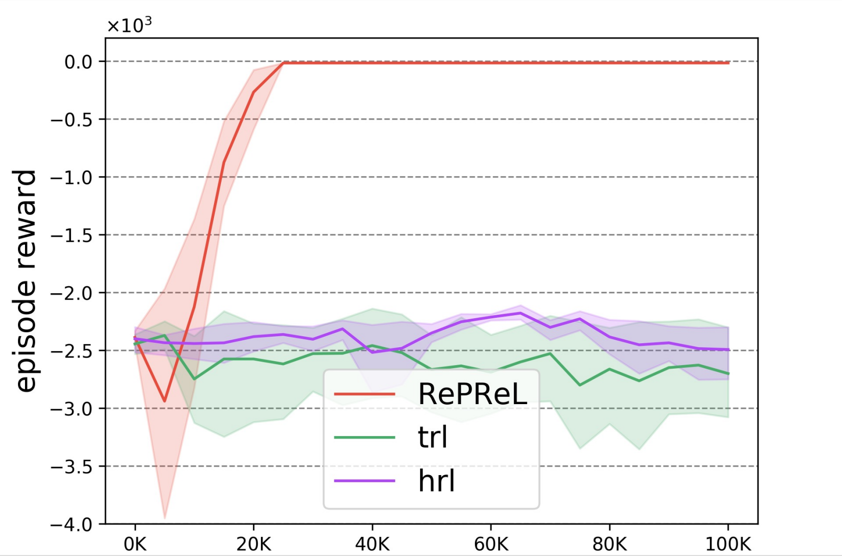
transport p1

trl: Taskable RL (Illanes et al. ICAPS 2020)

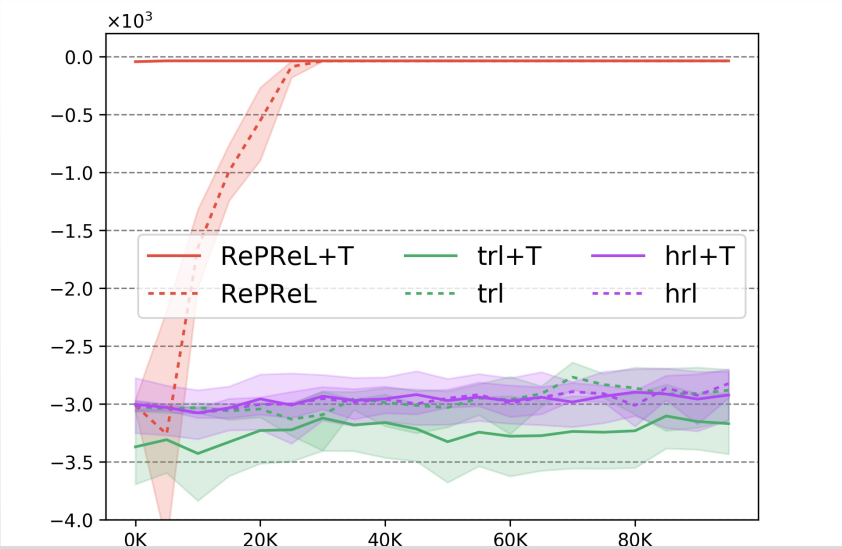
Experiments

- Sample efficiency
- Transfer across task
- Generalization across objects



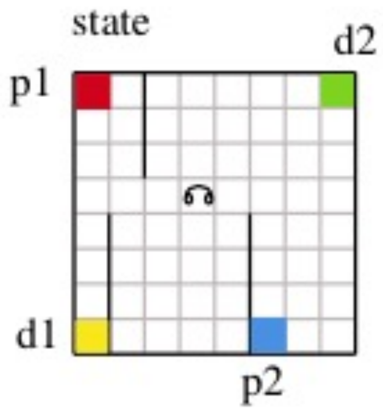


transport p1



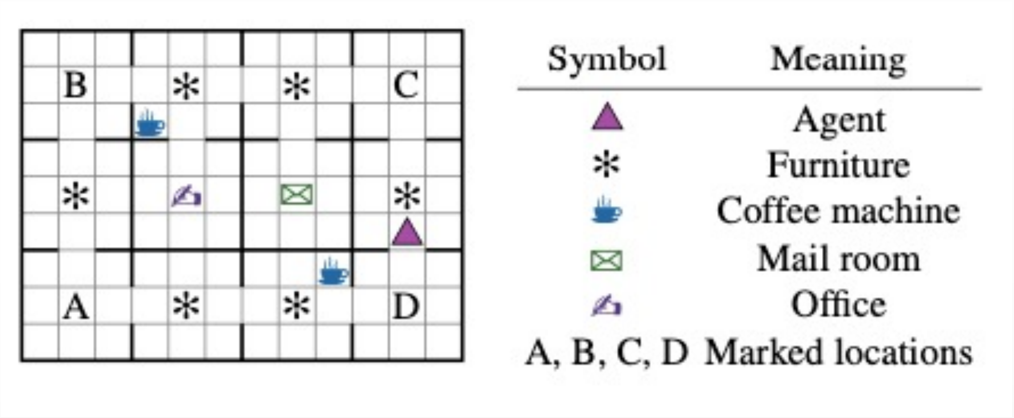
transport p1 and p2

Experiments

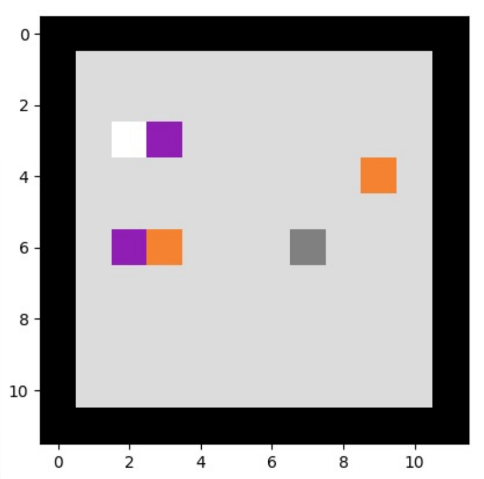


	state	abstract state (pick-up)
Relational {s}	$\mathrm{at}(\mathrm{p1},\mathrm{r}), \text{taxi-at}(13), \mathrm{dest}(\mathrm{p1},\mathrm{y}), \neg\,\text{at-dest}(\mathrm{p1}), \neg\,\text{in-taxi}(\mathrm{p1}), \mathrm{at}(\mathrm{p2},\mathrm{b}), \mathrm{dest}(\mathrm{p2},\mathrm{g}), \neg\,\text{at-dest}(\mathrm{p2}), \neg\,\text{in-taxi}(\mathrm{p1})\}$	$\{\mathrm{at}(\mathrm{p1},\mathrm{r}), \text{taxi-at}(13), \neg\,\text{in-taxi}(\mathrm{p1}), \mathrm{move}(\mathrm{Dir})\}$
Vector [s,a]	[at-p1, dest-p1, in-taxi-p1, at-dest-p2, at-p2, dest-p2, in-taxi-p2, at-dest-p2, taxi-at, move]	[at, taxi-at, in-taxi, move]

Experiments



Office world



Box world

Craft world

For human-level general intelligence, the ability to detect compositional structure in the domain and form task-specific abstractions are necessary.

THANKS