Dynamic Probabilistic Logic Models for Effective Abstractions in RL

Harsha Kokel, Arjun Manoharan, Sriraam Natarajan, Balaraman Ravindran, Prasad Tadepalli

UT DALLAS IIT Madras Oregon State University

Objective

Humans are quite natural at creating an abstract representation for efficient planning. They use different abstractions for different tasks. Can we leverage their domain knowledge?

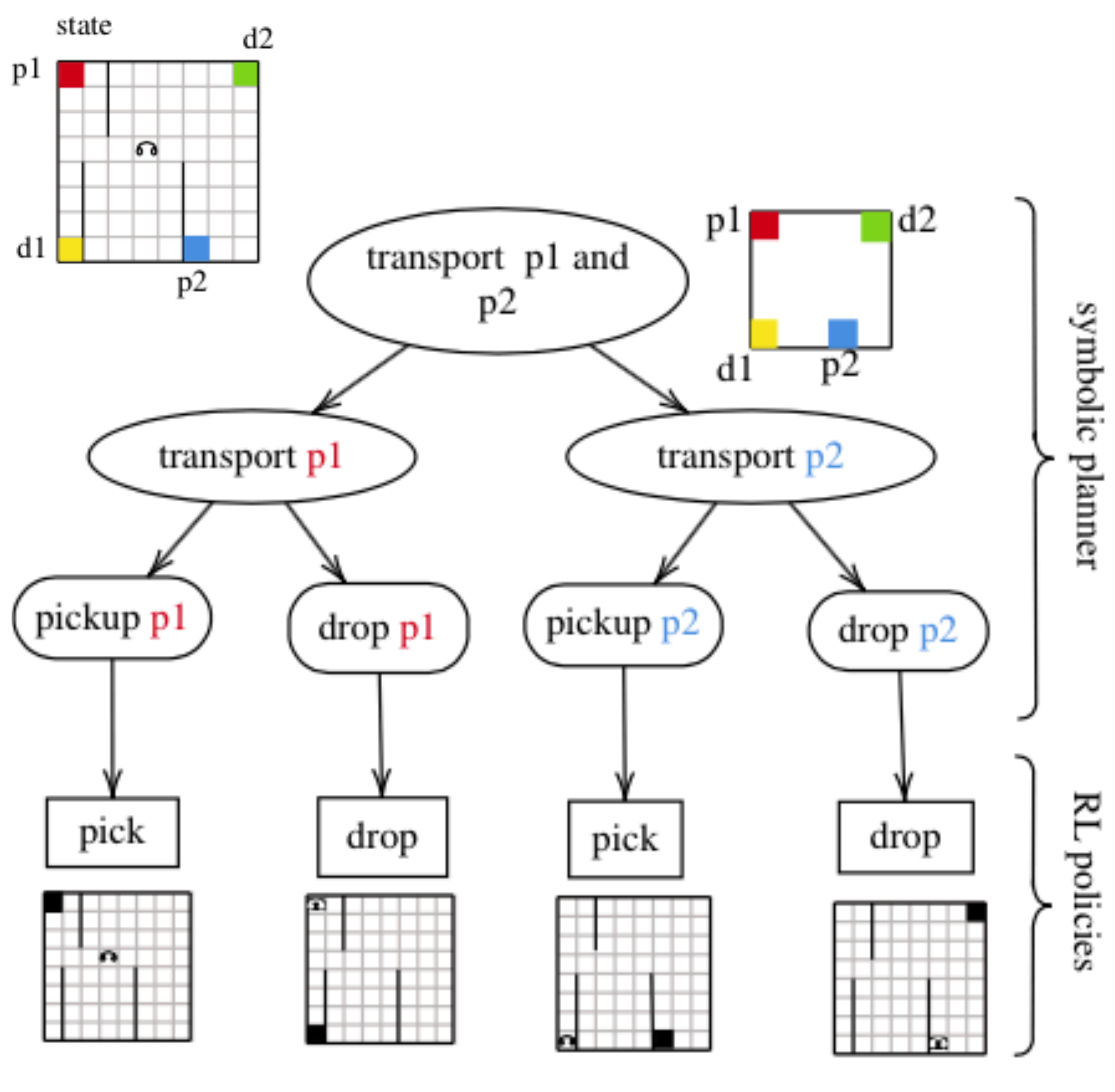
Here, we obtain task-specific state abstraction from humans in first-order statements to improve sample efficiency and generalization in RL.

RePReL

- Hierarchical Framework with Planner and RL
- Plan sequence of sub-tasks at high level using planner and learn to execute each subtask at lower level using RL
- Advantages
  - Planner provides compositionality
  - Different RL agent for execution allows separate state representations
- Use D-FOCI statements to identify relevant objects and relations for a given task.

Toy Example

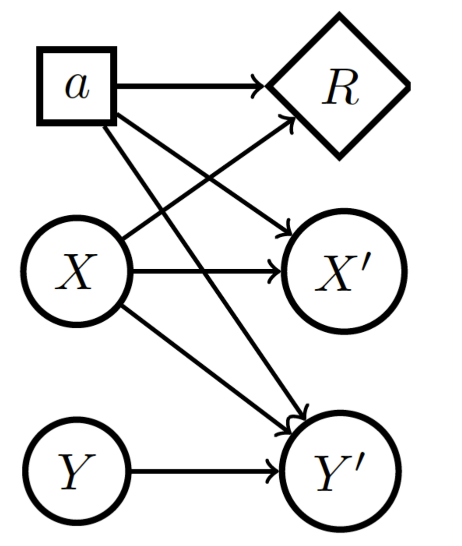
- Taxi domain with multiple passengers
- Acquire the sequence of passenger pickup and drop sub-tasks from planner
- Learn to execute the sub-task using RL
- Use different goal-conditioned RL for each sub-task
- Pickup subtask only needs pickup location, drop location is irrelevant
- Similarly, drop only needs destination, pickup location becomes irrelevant



Model-agnostic Abstraction

Definition: State variables Y are irrelevant in an MDP, if state variables can be partitioned into two disjoint subsets X and Y such that

$$P(x', y' \mid x, y, a) = P(x' \mid x, a) P(y' \mid x, y, a)$$
$$R(s, a, s') = R(x, a, x')$$



D-FOCI Statements

- First Order Conditional Influence (FOCI) statement is of the form
  "if *condition* then *X1* influence *X2*"
- FOCI statements encode the information that literal X2 is influenced only by the literals in X1 when the stated condition is satisfied
- We present Dynamic FOCI (D-FOCI) statements,

$$\text{sub-task} : \{\mathrm{p}(X_1), q(X_1)\} \xrightarrow{+1} \mathrm{q}(X_1)$$

- With +1, the D-FOCI encode the influence of literals in current time step on the literal in next time-step
- With sub-task, D-FOCI statement constraints the task being executed
- Given the current state and sub-task being executed we can obtain complete set of relevant state variables by grounding and rolling out the D-FOCI statements. This forms our abstract state
- Guarantees safe model-agnostic abstraction, if MDP satisfies the D-FOCI statement with fixed-depth unrolling

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

[at-p1 dest-p1, in-taxi-p1, at-dest-p2, at-p2, dest-p2, in-taxi-p2, at-dest-p2, taxi-at, move]

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

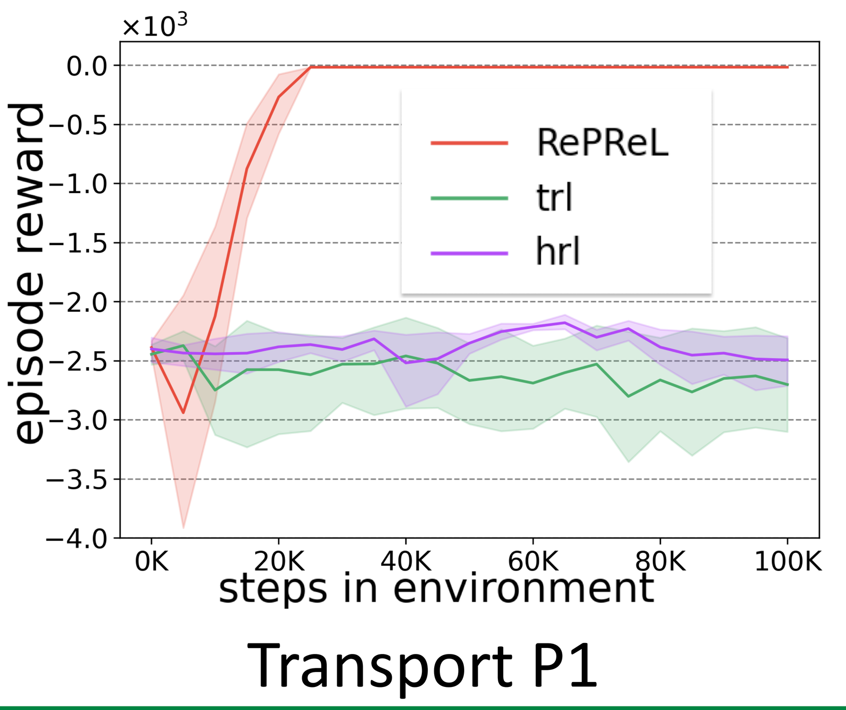
D-FOCI : $\{\text{taxi-at(L1), move(Dir)}\} \xrightarrow{+1} \text{taxi-at(L2)}$
$\{\text{taxi-at(L1), move(Dir)}\} \longrightarrow R$
pickup(P):
$\{\text{taxi-at(L1), at(P, L), in-taxi(P)}\} \xrightarrow{+1} \text{in-taxi(P)}$
pickup(P): in-taxi(P) $\longrightarrow R_o$

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

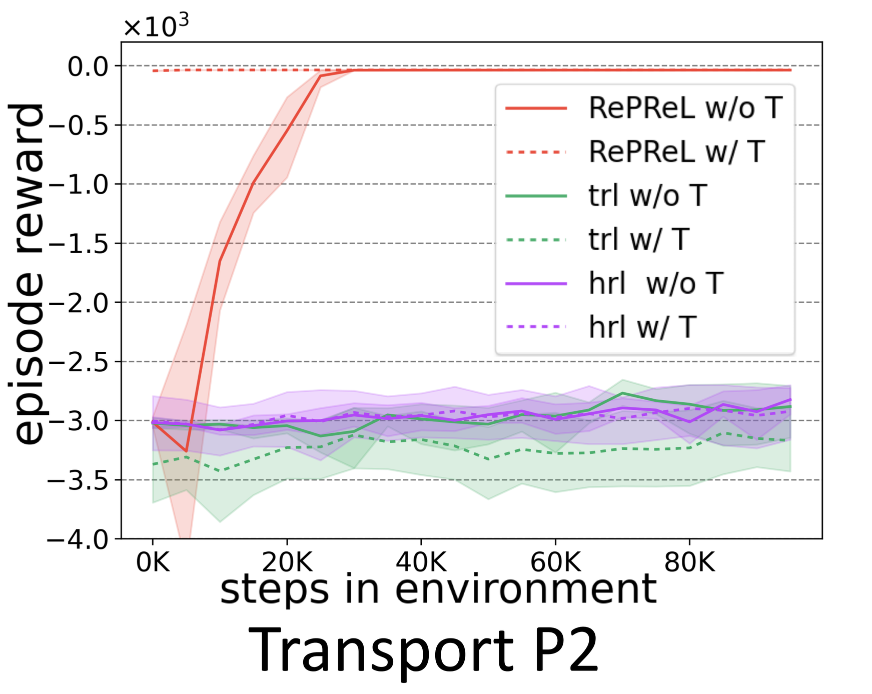
[at, taxi-at, in-taxi, move]

Experiments

Taxi domain



Transport P1



Transport P2

Human knowledge can help provide effective abstractions that enable efficient learning and effective generalization across tasks and objects.

More details on http://starling.utdallas.edu/papers/RePReL

STARLinG LAB