Symbolic Plans as High-Level Instructions for Reinforcement Learning

León Illanes, Xi Yan, Rodrigo Toro Icarte, Sheila A. McIlraith

ICAPS 2020

UNIVERSITY OF TORONTO

VECTOR INSTITUTE

CIFAR

What is this presentation about?

- **We want to tell an RL agent to do a specific task**

- We want declarative task specification...
  - like planning!
- ...without having a full description of the environment.
  - like RL!

Combine them?

Why use RL?

- Impressive results in low-level control problems
  - e.g., Rubik's cube manipulated by a robot hand
- Applicable without a given model
  - and without trying to learn one

...and why avoid it?

- Can be extremely inefficient
  - will need millions of training steps
- Is hard to use correctly!
  - specifying a reward is hard
  - value alignment problem

Why use AI Planning?

- It’s very efficient!
- Given a model, specifying new tasks is easy

...and why avoid it?

- Needs a model

A simple idea

- Use high-level model to define a task
  - Construct a high-level plan
  - Let RL deal with the low-level details

- Best of both worlds?

Our contributions

- Defined a new type of RL problem: **Taskable RL**
  - augments RL environments with high-level propositional symbols
  - this allows for easy representation of final-state goal problems
- Built a system to leverage symbolic models
  - high-level actions are used to identify options for hierarchical RL
  - learned option policies can be immediately transferred to new tasks
  - high-level plans are used as instructions, improving sample efficiency
- Showed that the approach is sound
  - Theoretically; when models are built properly
  - Empirically on some simple RL environments

Taskable RL Environments

$$E = \langle S, A, r, p, \gamma, P, L, R \rangle$$

- $\langle S, A, r, p, \gamma \rangle$ is an MDP

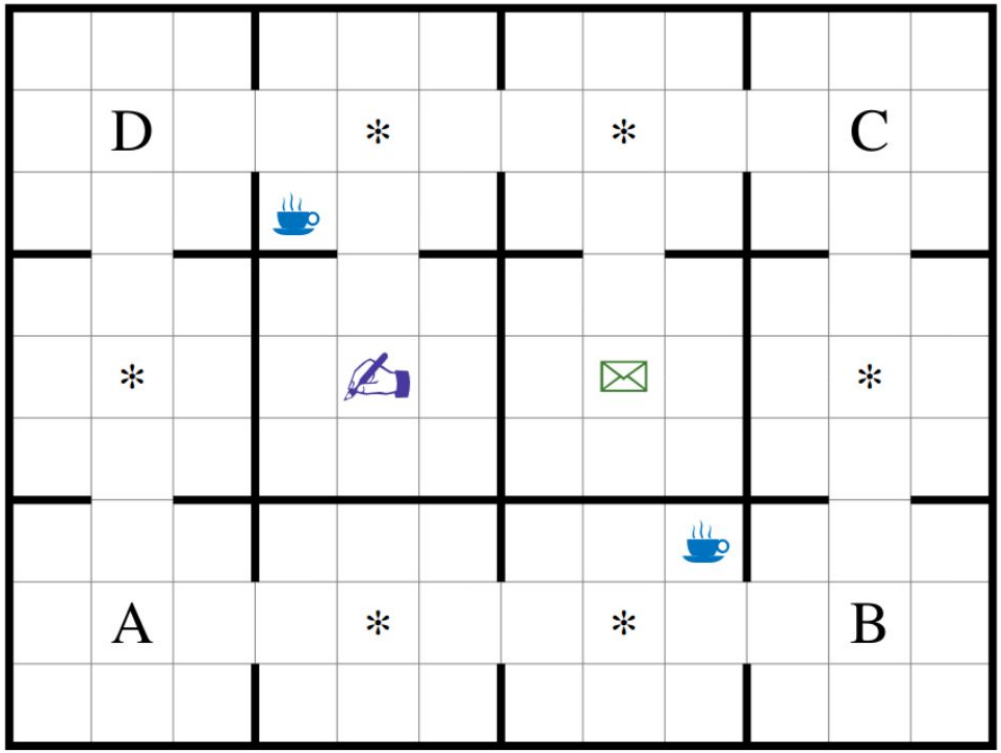
- P is a set of propositions
- $L : S \rightarrow 2^P$ is a labelling function
- $R \in \mathbb{R}$ is the goal reward parameter

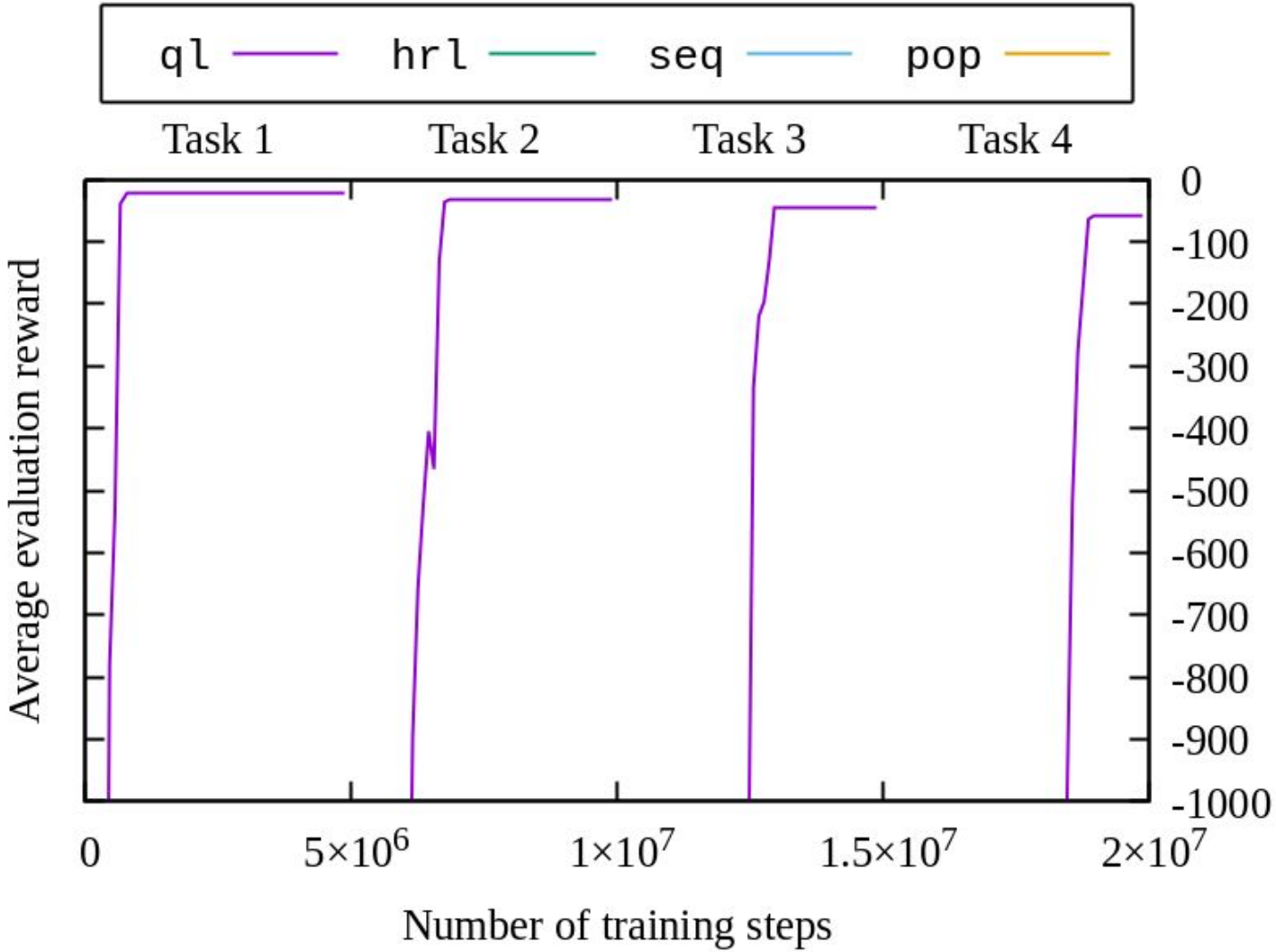
Plans as High-Level Instructions

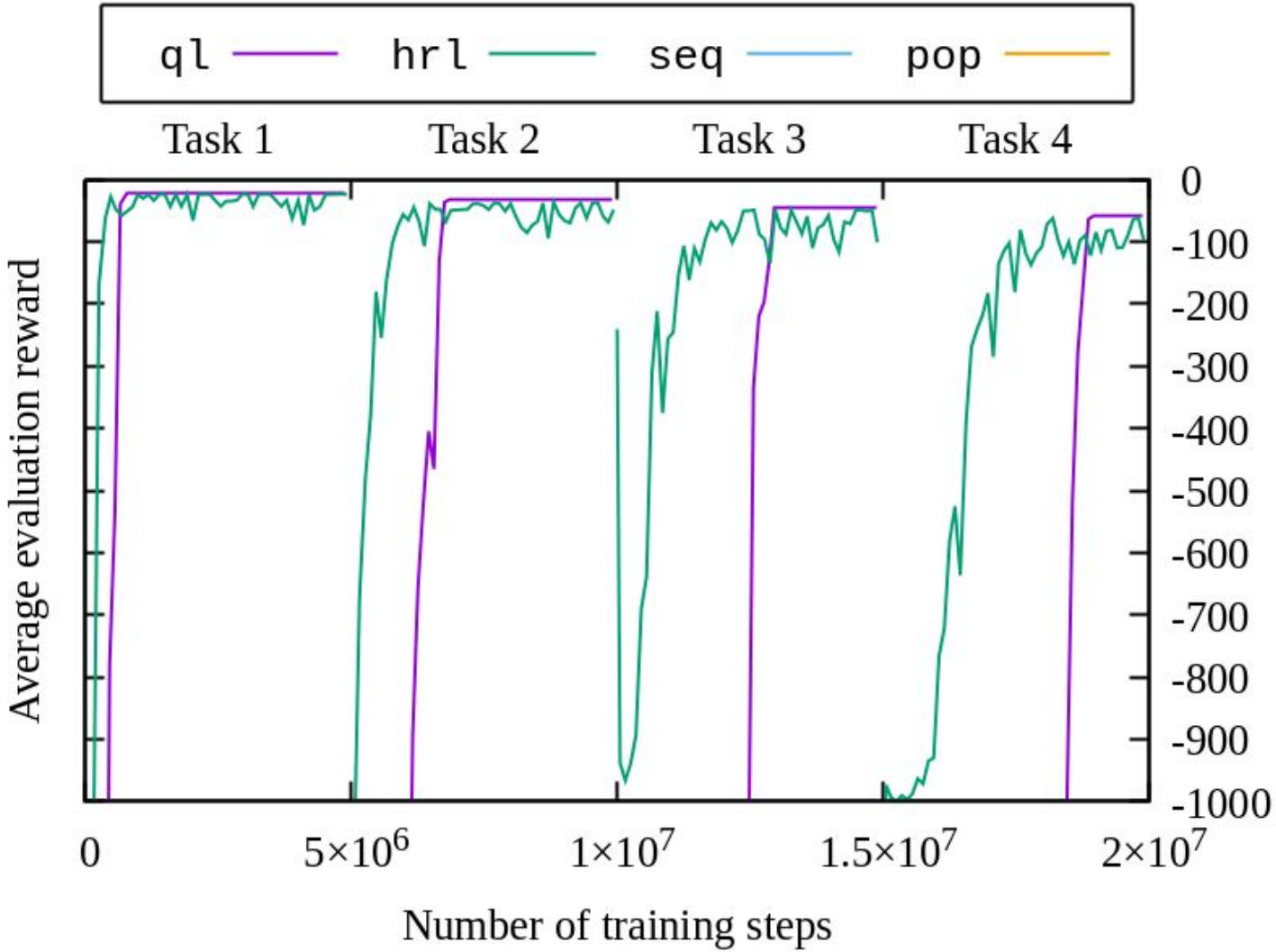
- Given a model, we can find plans
- Given a plan, we can try to execute it
  - Learn low-level policies for planning actions

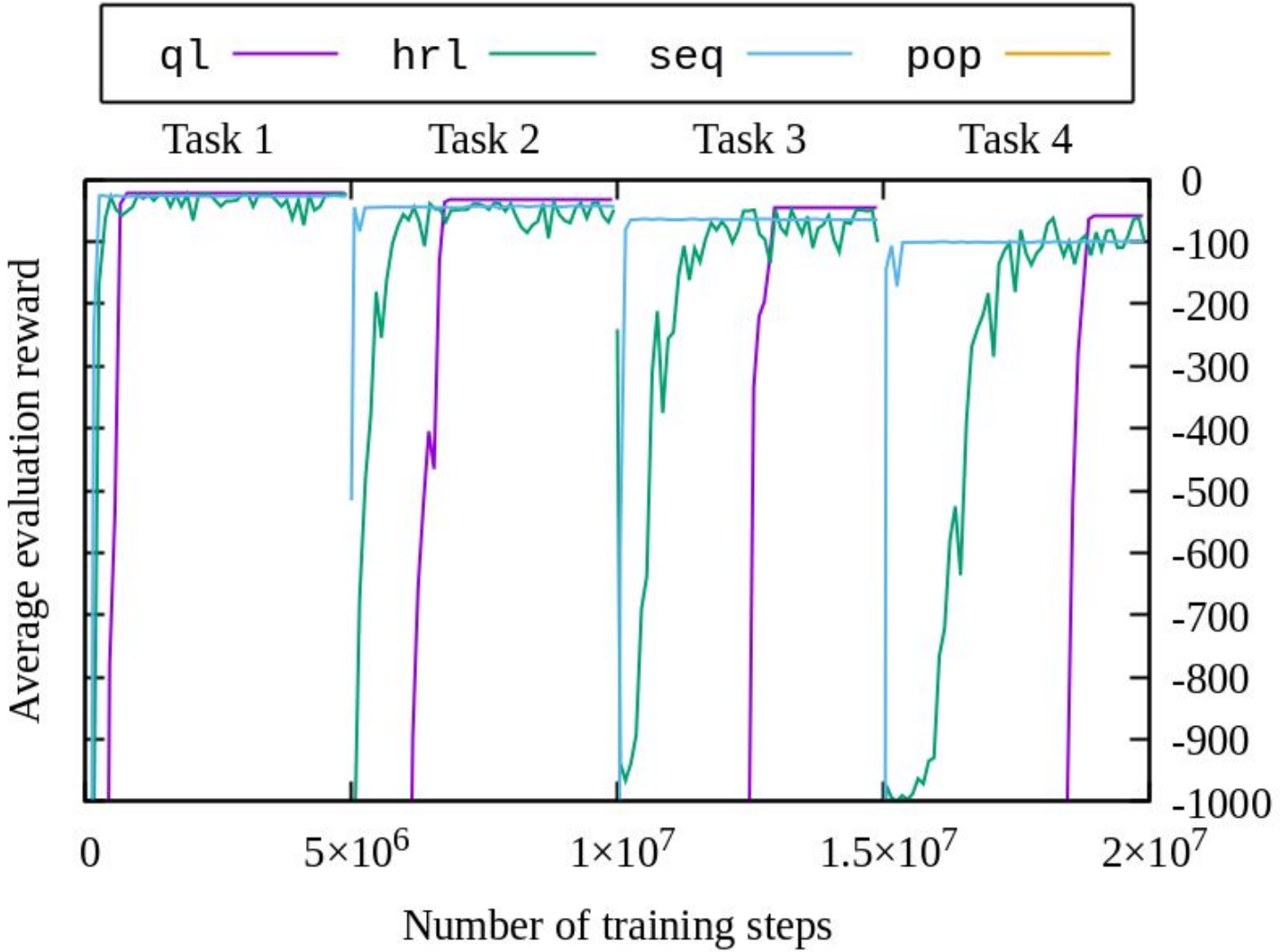
- Issues:
  - Suboptimality
    - Dealt with by partial-order planning
  - Unexpected outcomes (bad models, bad policies, etc.)
    - Execution monitoring

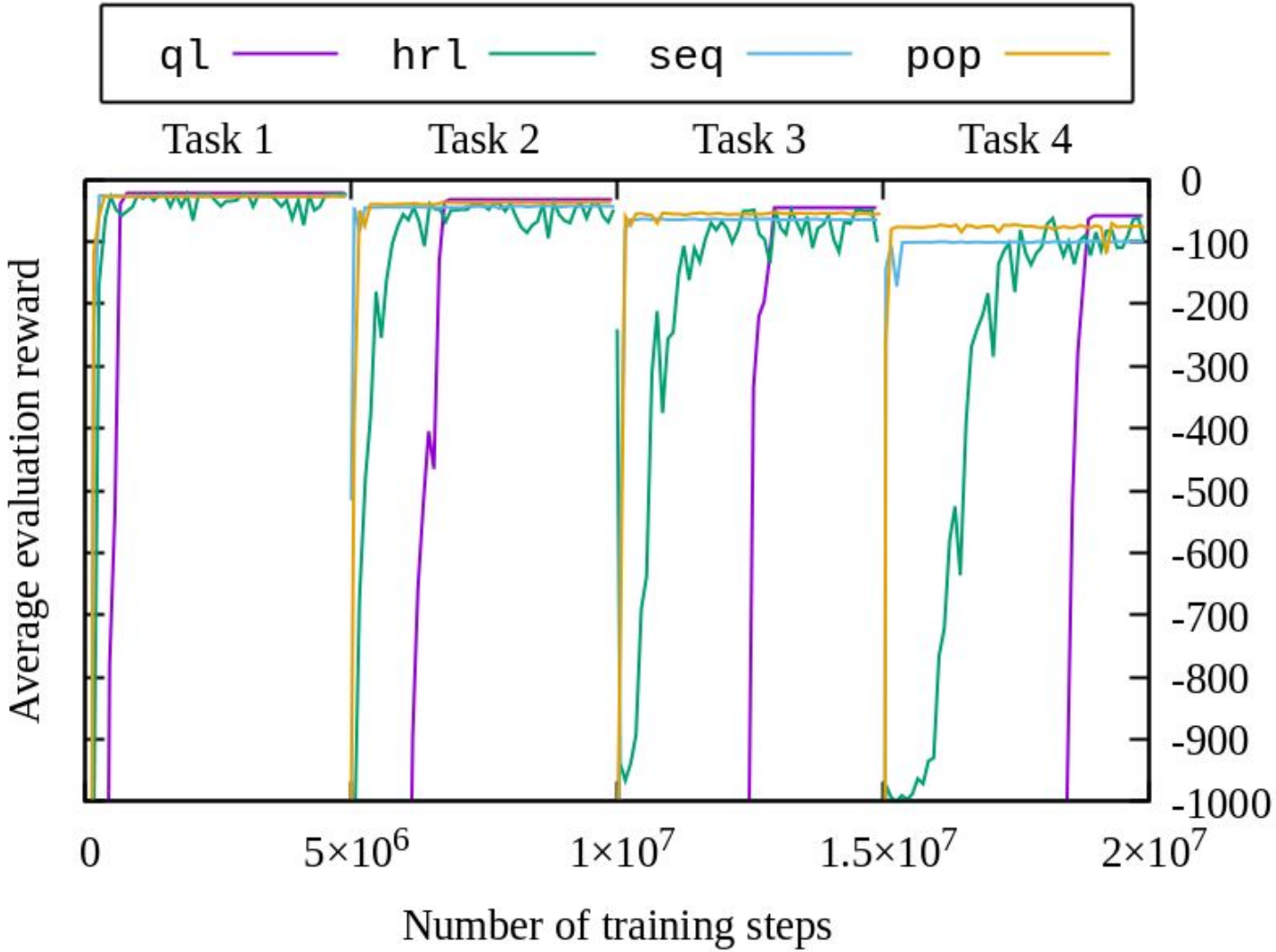
Experiments and results - The Office World



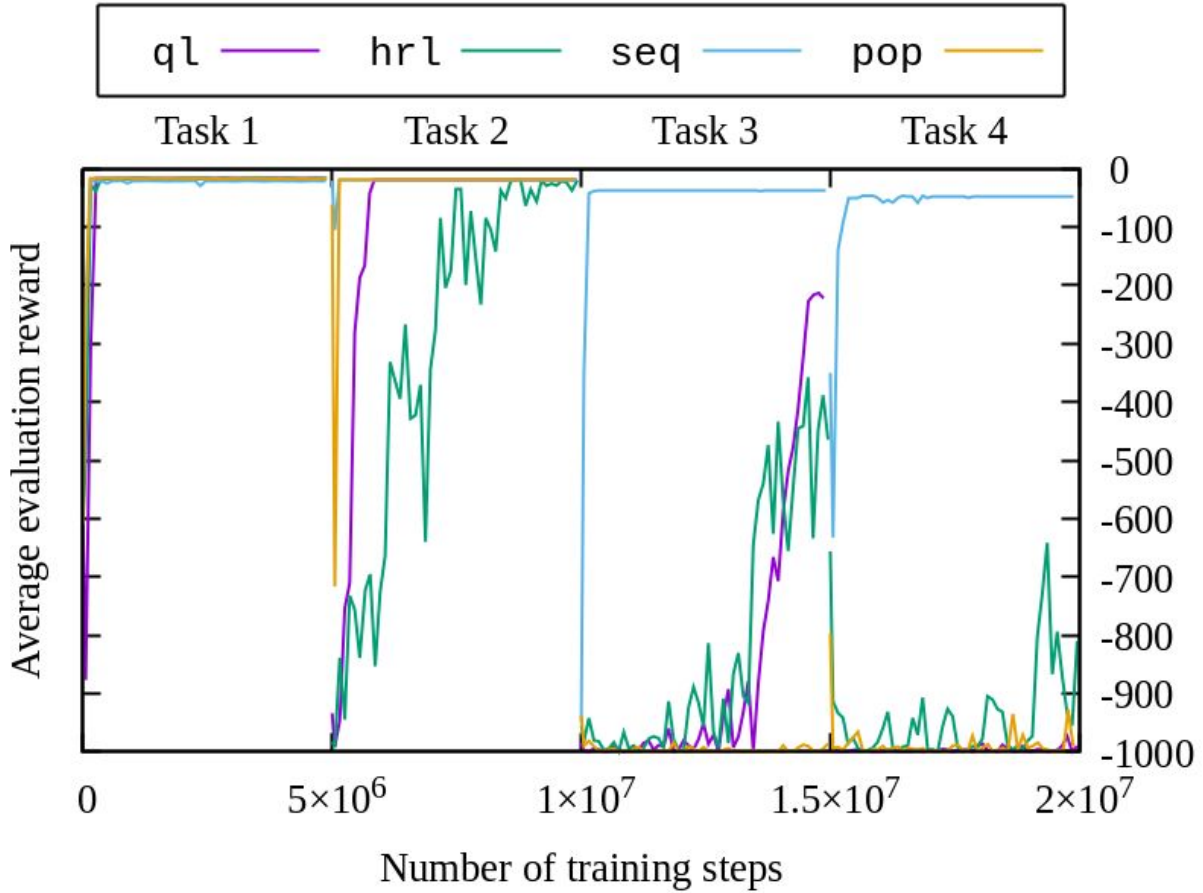

ql
hrl
seq
pop
Task 1
Task 2
Task 3
Task 4
Average evaluation reward
0
-100
-200
-300
-400
-500
-600
-700
-800
-900
-1000
0
5×10^6
1×10^7
1.5×10^7
2×10^7
Number of training steps


ql
hrl
seq
pop
Task 1
Task 2
Task 3
Task 4
Average evaluation reward
0
-100
-200
-300
-400
-500
-600
-700
-800
-900
-1000
0
5×10^6
1×10^7
1.5×10^7
2×10^7
Number of training steps


ql
hrl
seq
pop
Task 1
Task 2
Task 3
Task 4
Average evaluation reward
0
-100
-200
-300
-400
-500
-600
-700
-800
-900
-1000
0
5×10^6
1×10^7
1.5×10^7
2×10^7
Number of training steps


ql
hrl
seq
pop
Task 1
Task 2
Task 3
Task 4
Average evaluation reward
0
-100
-200
-300
-400
-500
-600
-700
-800
-900
-1000
0
5×10^6
1×10^7
1.5×10^7
2×10^7
Number of training steps

Other experiments - The Minecraft World



Summary

- Defined **Taskable RL**, a new type of RL problem
- Built a system that leverage symbolic models
- Showed that the approach is sound and effective