Learning Symbolic Representations for Reinforcement Learning of Non-Markovian Behavior

Phillip Christoffersen Andrew C. Li Rodrigo Toro Icarte Sheila A. McIlraith†

Department of Computer Science, University of Toronto, Toronto, Canada
Vector Institute for Artificial Intelligence, Toronto, Canada
†Schwartz Reisman Institute for Technology and Society, Toronto, Canada

Overview

Many real-world RL tasks require temporally extended behavior.
Such reward-worthy behavior requires the execution of a pattern of actions that yields reward only upon completion of the pattern.

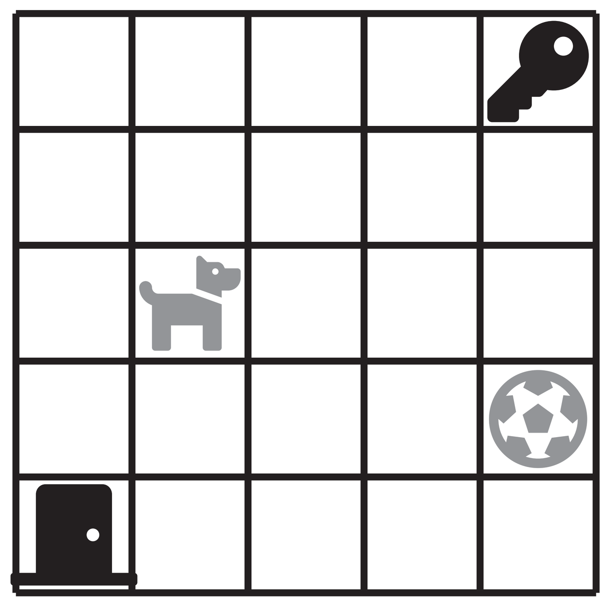
Standard deep RL solution:
- Use a recurrent neural network (RNN) — hidden state summarizes state-action history.
- Disadvantage: takes large number of samples to train.

Standard KR-based solution:
- Use a pre-determined abstraction of the state space to define reward-relevant features, realized via a *labelling function*.
- Disadvantage: requires *a priori* domain knowledge.

Our solution:
- Automatically learn reward-relevant features from the state-action history!
- Use the learned features to accelerate learning.

In our experiments with non-Markovian goals, we outperform state-of-the-art RL based on RNNs.

Example

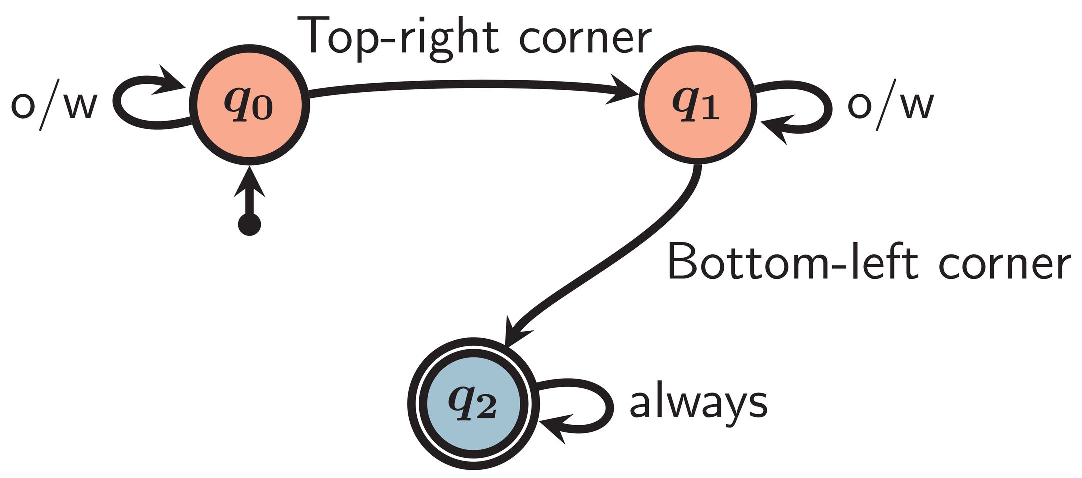


- **Left:** A 5x5 gridworld where the agent must **first reach the key** and only then **arrive at the door** to solve the task.
- KR-based solution: augment the RL agent with an observation **have_keys**.
- Optimal behavior:
  - When **have_keys** is false, move towards the key.
  - When **have_keys** is true, move towards the door.
- **Note:** with the **have_keys** proposition, the environment becomes Markovian.
- We consider **non-Markovian goals**: reward is 1 or 0 based on whether the goal is achieved.

Method

Algorithm 1: AutRL

dfa ⟵ empty_automata;
π ⟵ uniform_random_policy;
traces ⟵ $\emptyset$;
while *true* **do**
  sample_traces ⟵ sample(π, N) ;
  append traces with sample_traces;
  if *sample_traces inconsistent with dfa* **then**
    dfa ⟵ aut_learn(traces);
  end
  π ⟵ markov_learn(sample_traces $\times$ dfa);
end



A DFA learned by `AutRL` *for the example.*

Main idea:

- Train a DFA to predict whether a state-action sequence receives reward 0 or 1.
- Augment the agent with the DFA state and use **any standard RL algorithm** to learn policies which are non-Markovian in the original problem.
- If a learned DFA can predict this reward perfectly, then the problem **becomes completely Markovian**
- Alternate between DFA learning and Markovian learning, until a consistency condition is met for the learned DFA.

Details:

- Recent advances in automata learning (Shvo et al, 2020) allow us to **efficiently** learn small DFAs that are **robust to noise**.
- In practice, the DFA need not perfectly classify the reward. For example, it is enough to be able to partition the DFA states into ones where **have_keys** is true, and where **have_keys** is false.
- **We prove** `AutRL` **optimally converges** under mild assumptions on the Markovian learning policy, exploratory policy, and on the structure of the goal-based reward.

Experimental Evaluation

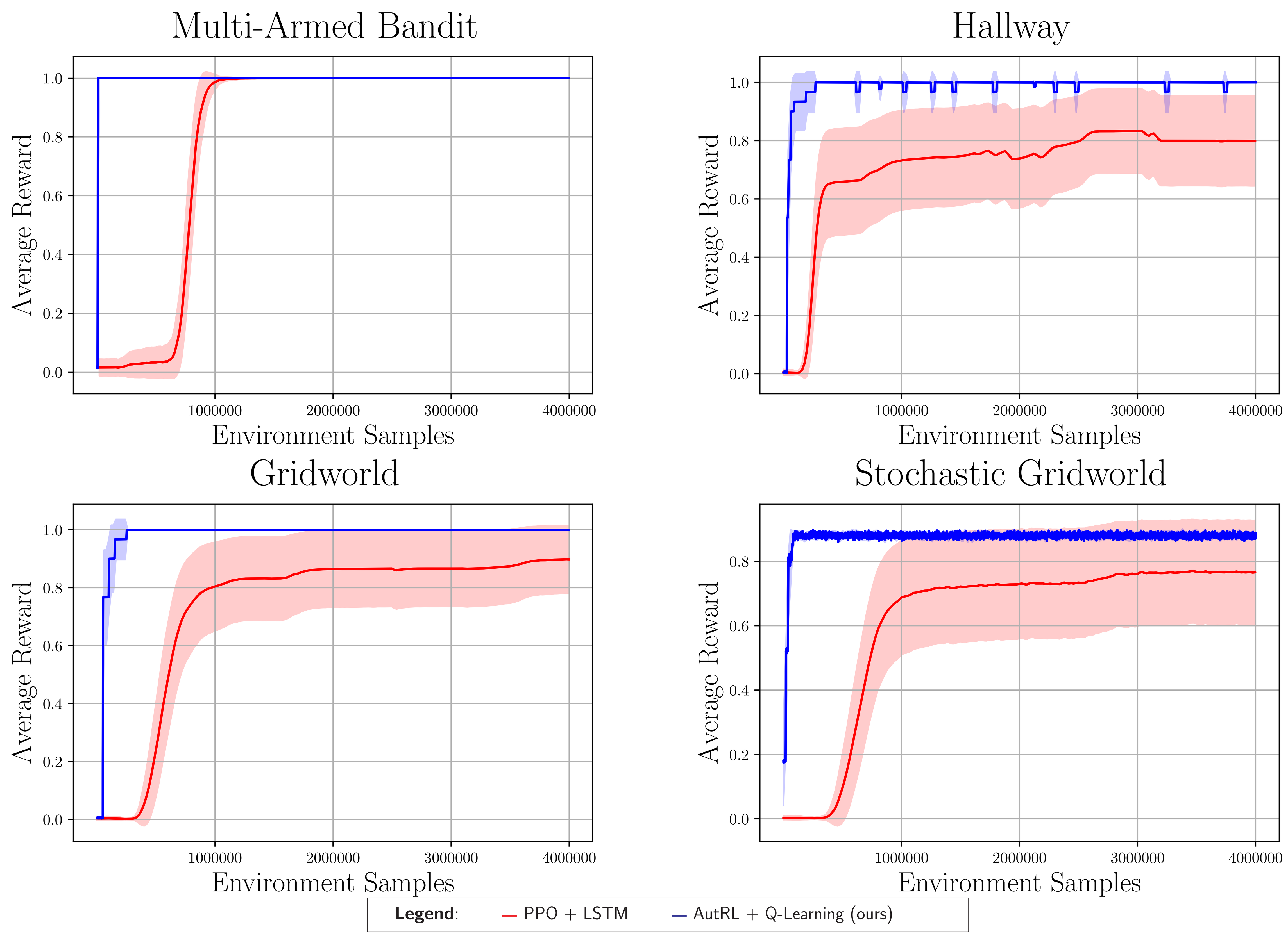


Figure: The results from the conducted experiments. The error bars are 95% confidence intervals over 30 runs.

Discussion

Analysis of Results:

- **AutRL is, at most, over an order of magnitude more sample-efficient than Recurrent-PPO**, to 95% confidence.
- AutRL exhibited much more consistent and stable learning than the Recurrent-PPO.
- In practice, checking for high performance of Markov learning given a DFA, instead of perfect reward classification, yields better stability and sample efficiency.

Selected Related Work:

- Prior works also attempt to learn reward structure, but are sensitive to noise (e.g. Toro Icarte et al., 2019; Xu et al., 2020).
- (Gaon & Brafman, 2020) shares our high-level idea but we leverage advances in automata learning and show this can outperform standard state-of-the-art RL.

Limitations and Future Work:

- This method does not perform well if the goal histories $\mathcal{G}$ do not form a regular language in $S \times A$ (e.g. counting-related tasks).
- We focus on non-Markovian reward, future work can consider partial observability.
- DFA learning method requires the state-action space to be small and discrete: future work should focus on continuous states and scalability.
- We only support non-Markovian goals rather than general reward functions.

Bibliography:

- Maor Gaon and Ronen Brafman. Reinforcement learning with non-markovian rewards. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3980–3987, 2020.
- Rodrigo Toro Icarte, Ethan Waldie, Toryn Klassen, Rick Valenzano, Margarita Castro, and Sheila McIlraith. Learning reward machines for partially observable reinforcement learning. In *Proceedings of the 32nd Conference on Advances in Neural Information Processing Systems (NeurIPS)*, pages 15523–15534, 2019.
- Maayan Shvo, Andrew C Li, Rodrigo Toro Icarte, and Sheila A McIlraith. Interpretable Sequence Classification via Discrete Optimization. *arXiv preprint arXiv:2010.02819*, 2020.
- Zhe Xu, Ivan Gavran, Yousef Ahmad, Rupak Majumdar, Daniel Neider, Ufuk Topcu, and Bo Wu. Joint inference of reward machines and policies for reinforcement learning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 590–598, 2020.