Deep Abstract Q-Networks

Melrose Roderick, Chris Grimm, Stefanie Tellex
Humans to Robots Lab, Department of Computer Science, Brown University

Abstract

We examine the problem of learning and planning on high-dimensional domains with long horizons and sparse rewards. We combine recent techniques of deep reinforcement learning with existing model-based approaches using an expert-provided state abstraction.

Motivation

- Deep Q-Networks (DQN) are great at producing short-horizon policies
- DQN struggles on long-horizon domains
- These long-horizon domains are the closest analogues to real-world robotics tasks

Related Work

- Intrinsic motivation (IM) [Bellemare et al. 2016]
- Hierarchical methods [Kulkarni et al. 2016, Vezhnevets et al. 2017]

Some of these methods cannot learn to retrace their steps and opt to end their life to return to the start. These methods do not take advantage of the long-term planning benefits of model-based algorithms.

Results

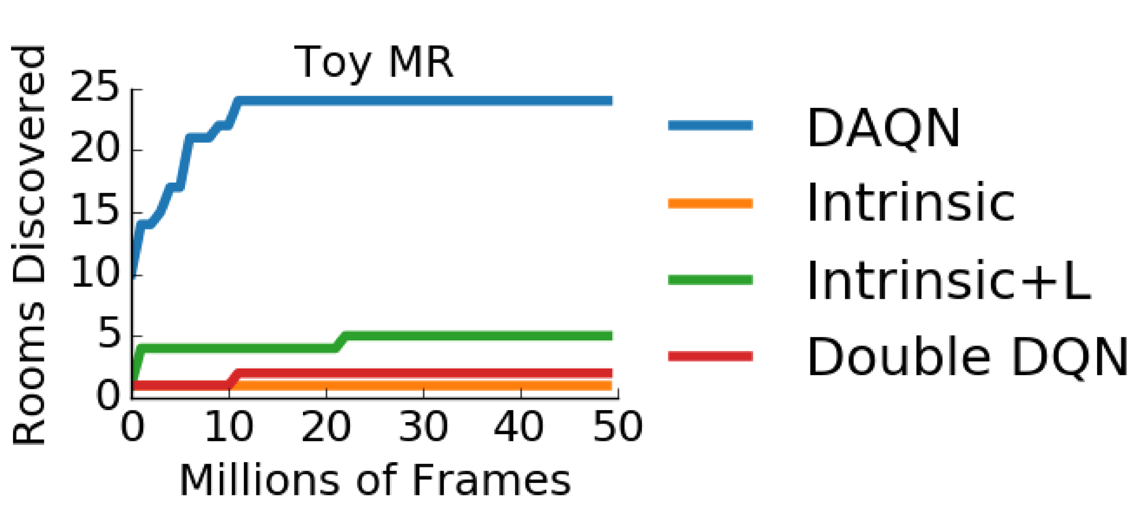
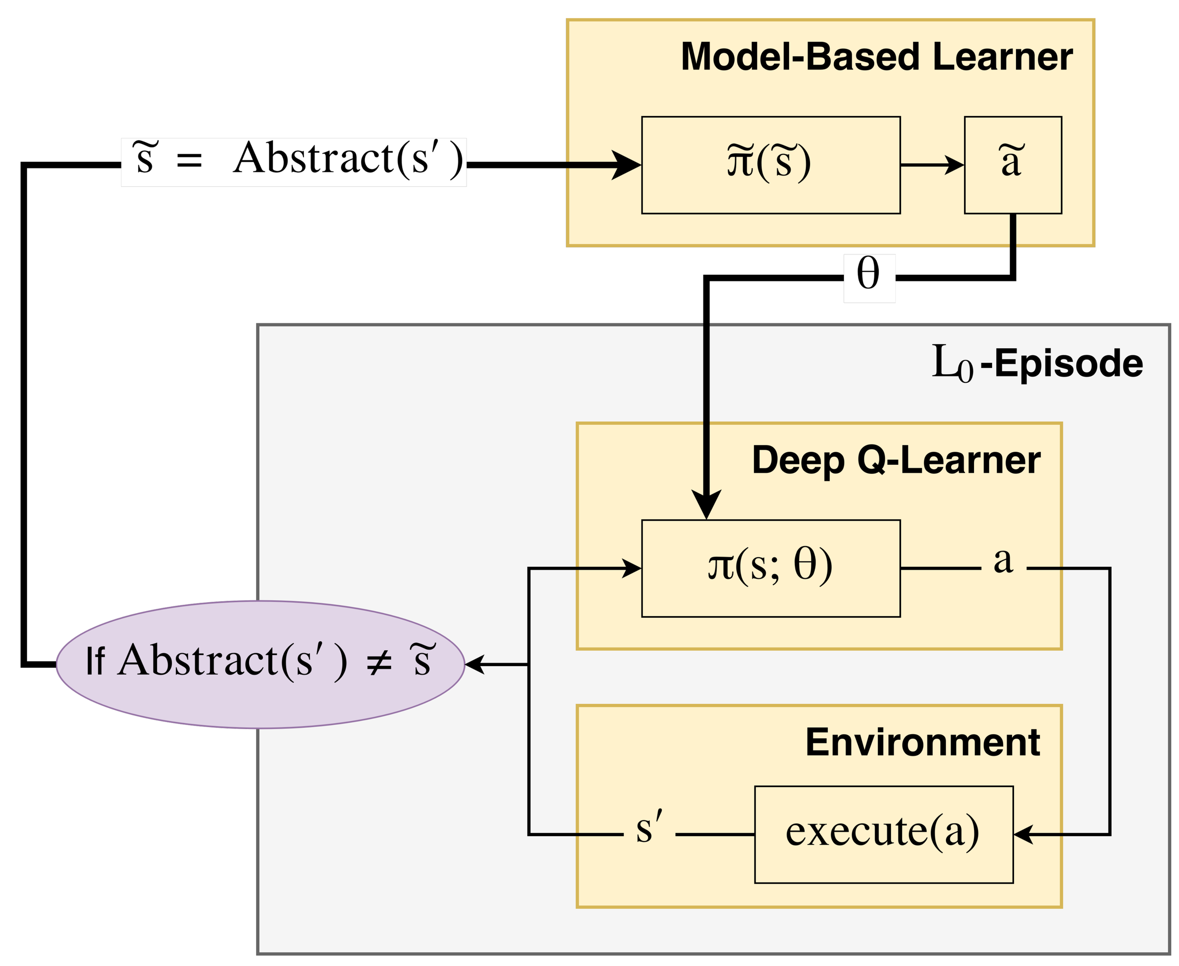


Figure: Rooms discovered in the Toy MR domain.

- Our method outperformed DQN in all experiments
- Our method outperforms Intrinsic Motivation on Toy MR and some of the Atari domains
- Our method fails to explore as far on Montezuma's Revenge because it cannot cross timed-based traps (vanishing deadly walls and disappearing bridges)

Model



- An annotated abstraction of the state-space is provided to the agent: $F : \mathcal{S} \to \tilde{\mathcal{S}}$.
- A model-based planner learns on this high-level, discrete state-space, $\tilde{\mathcal{S}}$
- Deep Q-Networks learn to transition between these high-level states
- The reward function of the high-level learner is simply a sum over observed rewards, $\Sigma\, r_i$
- We modify the terminal set and reward function of the low-level learner as follows:

$$\mathcal{E}_{\text{episode}} = \mathcal{E}_{\text{env}} \cup \{s \in \mathcal{S} : \mathcal{F}(s) \neq \tilde{s}_{\text{init}}\}$$

$$\mathcal{R}_{\text{episode}}(s, a, s') = \begin{cases} 1 & \text{if } \mathcal{F}(s') = \tilde{s}_{\text{goal}} \\ 0 & \text{else.} \end{cases}$$

Learning

The agent is only provided the state abstraction function so it must learn:

- The high-level hierarchy: what high-level states are connected
- The low-level policies: how to navigate these transitions (using images)
- The high-level policy: the plan over abstract states

To construct the hierarchy, the agent builds up the state and action sets, $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{A}}$, as transitions are discovered.
The low-level learners use the DQN algorithm.
The high-level learner uses the model-based R-Max algorithm.

Atari Domains

We test on the Atari 2600 games Montezuma's Revenge and Venture.

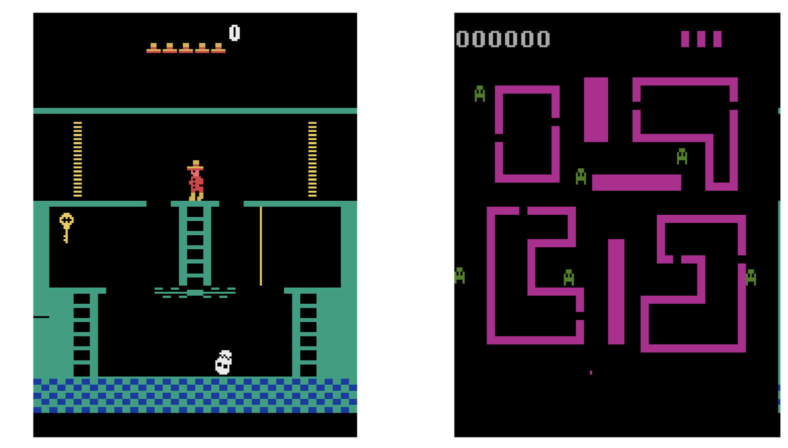


Figure: Example screens of Montezuma's Revenge and Venture.

Grid Domain: "Toy MR"

We created a domain with to parallel Montezuma's Revenge without the low-level complexities: time-based traps, jumps, and monsters.
Toy MR is a massive maze of rooms, where each room is a 11×11 grid.

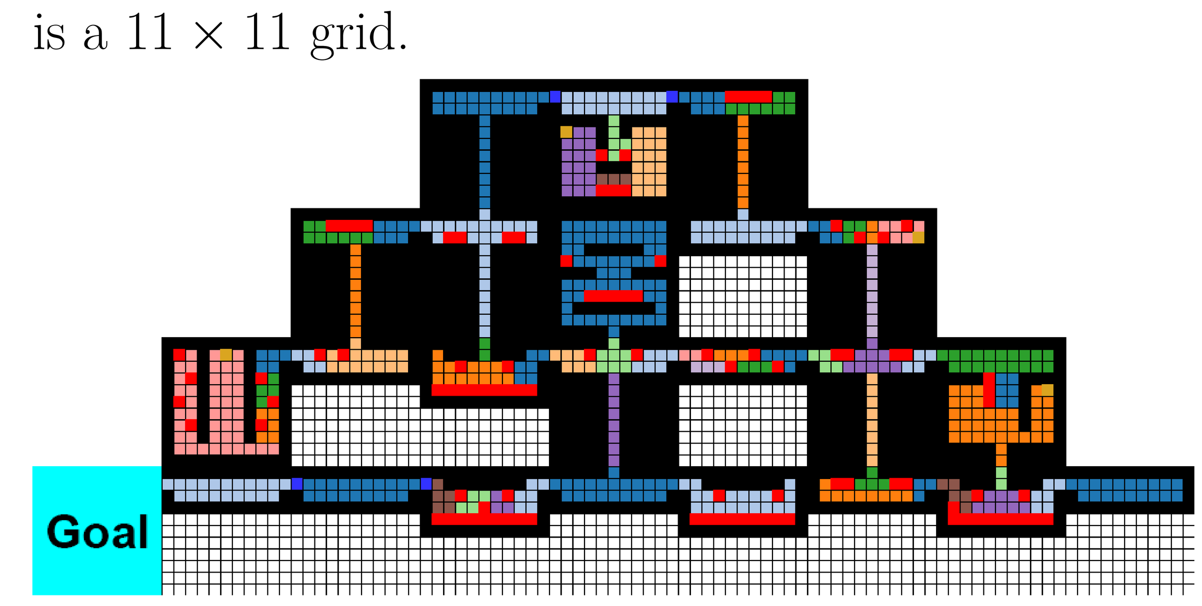


Figure: The map of all the rooms in Toy MR. The sectors provided to the agent in Toy MR are color-coded.

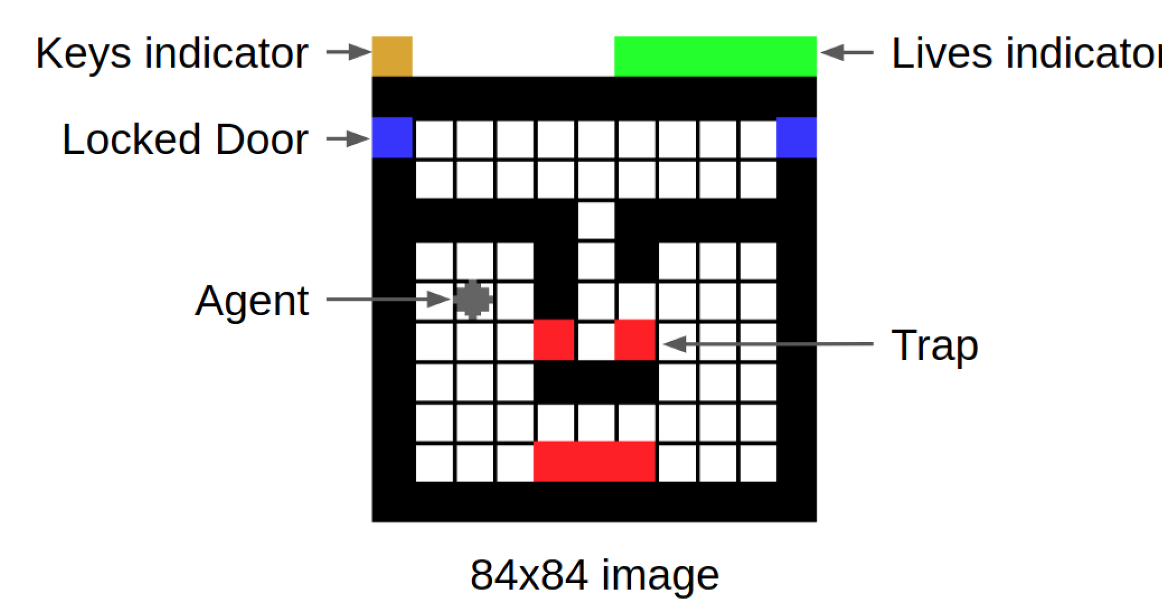


Figure: Example screen in Toy MR

Abstraction

The abstraction provided consists of the attributes:

- ⟨`Agent location`⟩ (room and sector)
- ⟨`Number of keys`⟩
- ⟨`i'th Key collected`⟩ (4 total in Toy MR)
- ⟨`j'th Door unlocked`⟩ (4 total in Toy MR)

Future Work

- Combine motivated exploration with DAQN
- Learn the state abstraction function

Experiment Results

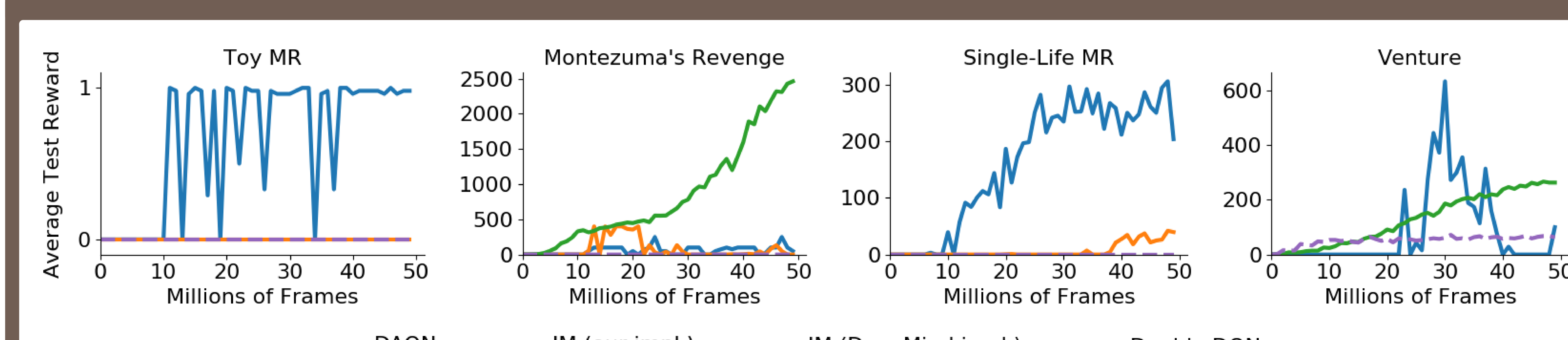


Figure: Average reward in the Toy MR, Montezuma's Revenge, Single-Life MR, and Venture domains using the following models: DAQN (blue), Our implementation of Intrinsic Motivation (orange), Deep Mind's implementation of Intrinsic Motivation (green), Double DQN (purple).

Acknowledgements

This material is based upon work supported by the National Science Foundation under grant numbers IIS-1426452, IIS-1652561, and IIS-1637614, DARPA under grant numbers W911NF-10-2-0016 and D15AP00102, and National Aeronautics and Space Administration under grant number NNX16AR61G.