Design and Implementation of General Purpose Reinforcement Learning Agents

Tyler Streeter

November 17, 2005

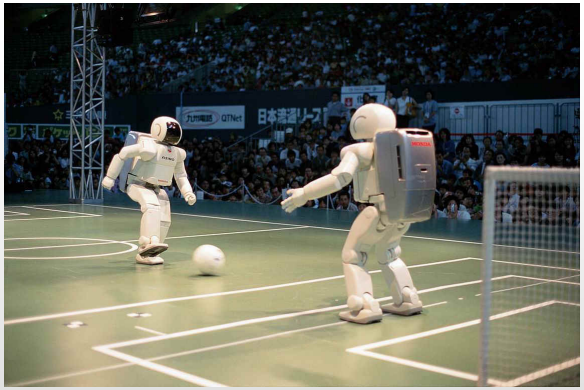
Human Computer Interaction

Iowa State University
of Science and Technology

Virtual Reality Applications Center

Motivation

Intelligent agents are becoming increasingly important.

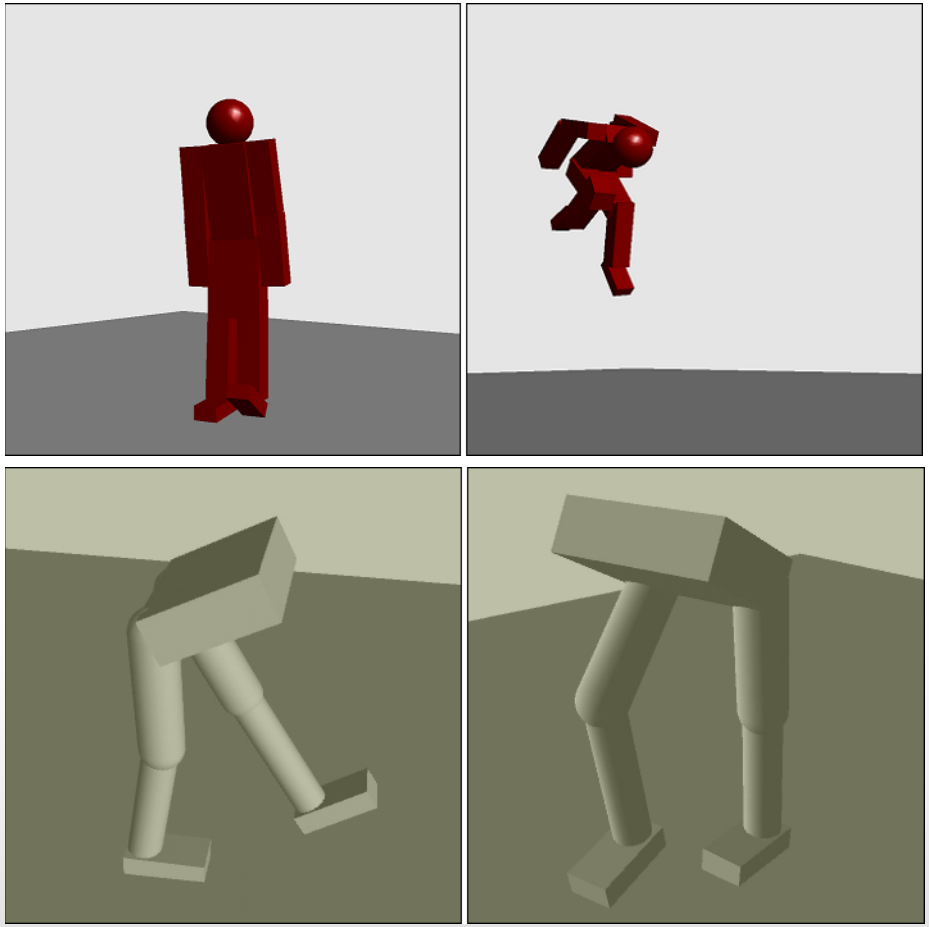


Motivation

- Most intelligent agents today are designed for very specific tasks.
- Ideally, agents could be reused for many tasks.
- Goal: provide a general purpose agent implementation.
- Reinforcement learning provides practical algorithms.

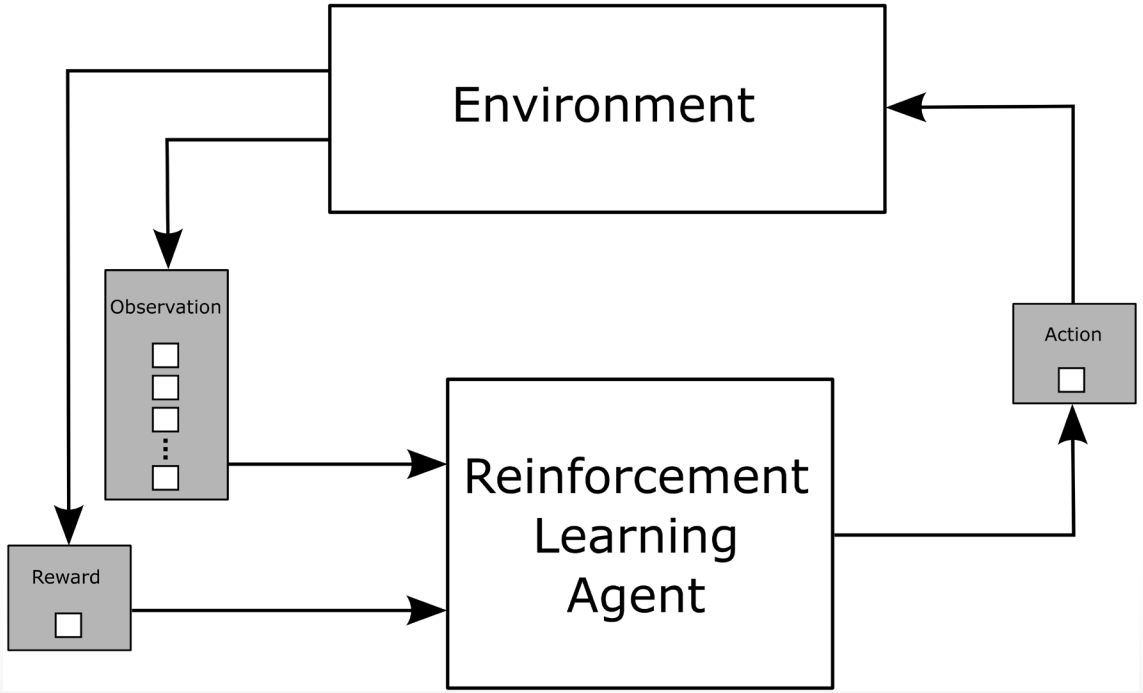
Initial Work

Humanoid motor control through neuroevolution (i.e. controlling physically simulated characters with neural networks optimized with genetic algorithms) (videos)



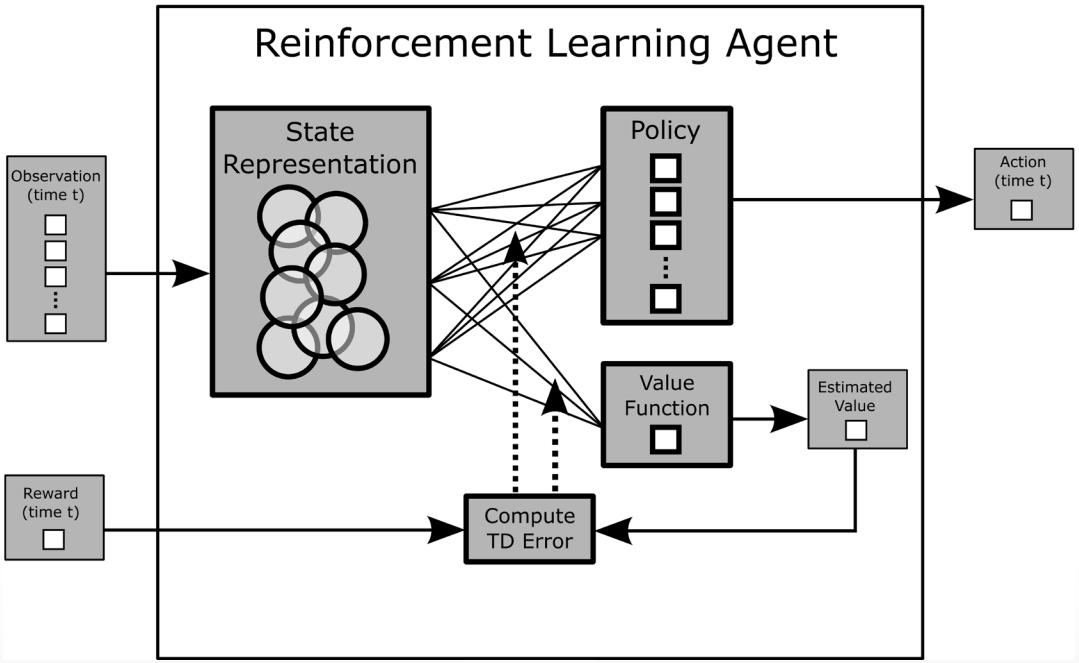
What is Reinforcement Learning?

- Learning how to behave in order to maximize a numerical reward signal
- Very general: almost any problem can be formulated as a reinforcement learning problem

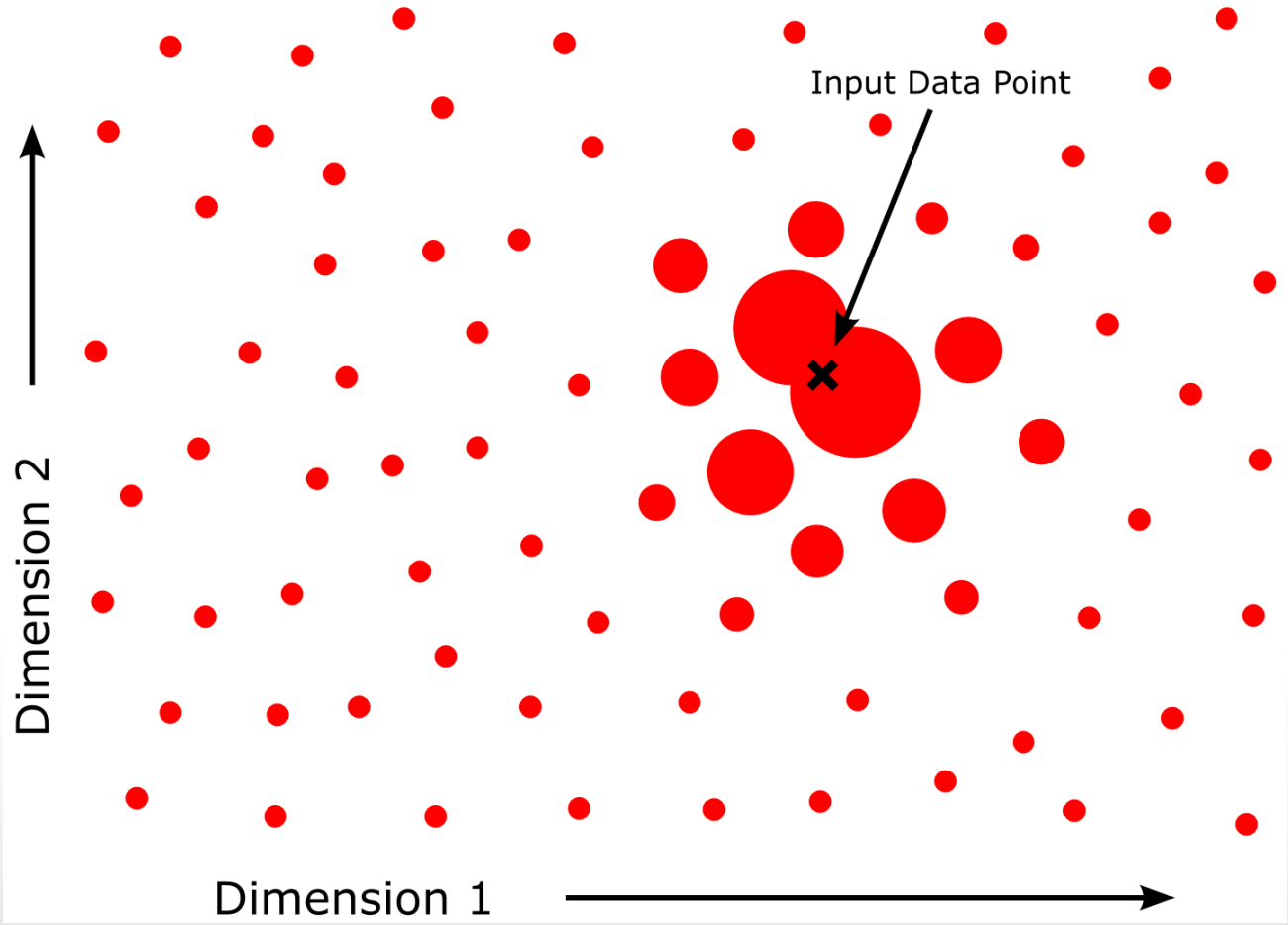


Basic RL Agent Implementation

- Main components:
  - Value function: maps states to “values”
  - Policy: maps states to actions
- State representation converts observations to features (allows linear function approximation methods for value function and policy)
- Temporal difference (TD) prediction errors train value function and policy

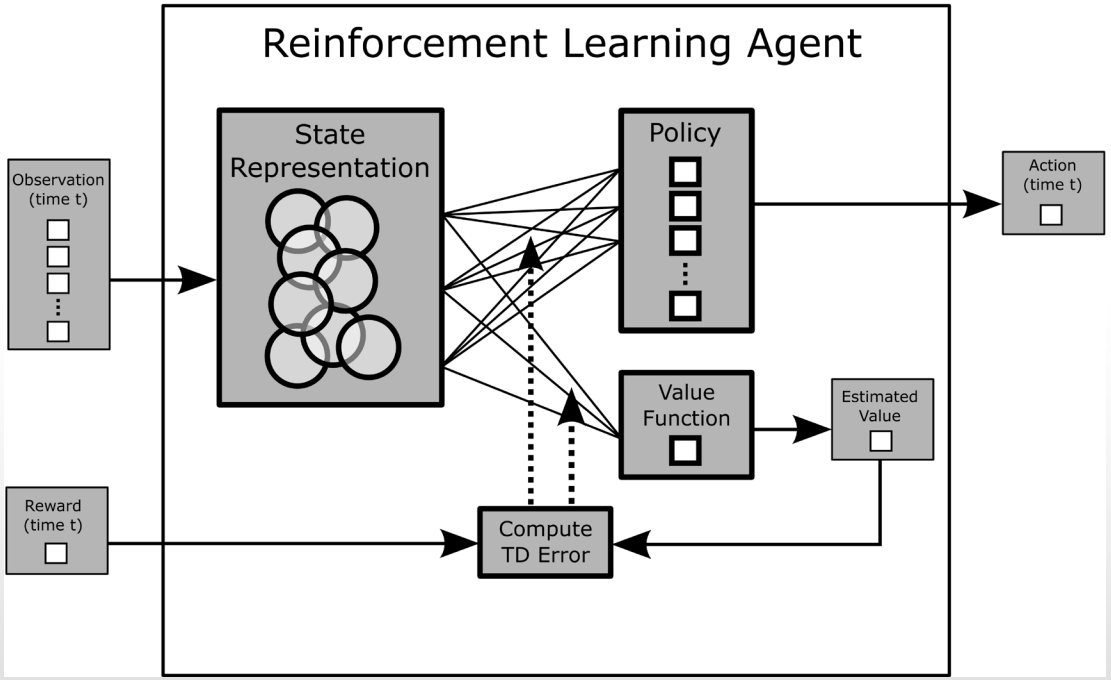


RBF State Representation



Temporal Difference Learning

- Learning to predict the difference in value between successive time steps.
- Compute TD error: $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$
- Train value function: $V(s_t) \leftarrow V(s_t) + \eta\delta_t$
- Train policy by adjusting action probabilities



Biological Inspiration

- Biological brains: the proof of concept that intelligence actually works.
- Midbrain dopamine neuron activity is very similar to temporal difference errors.

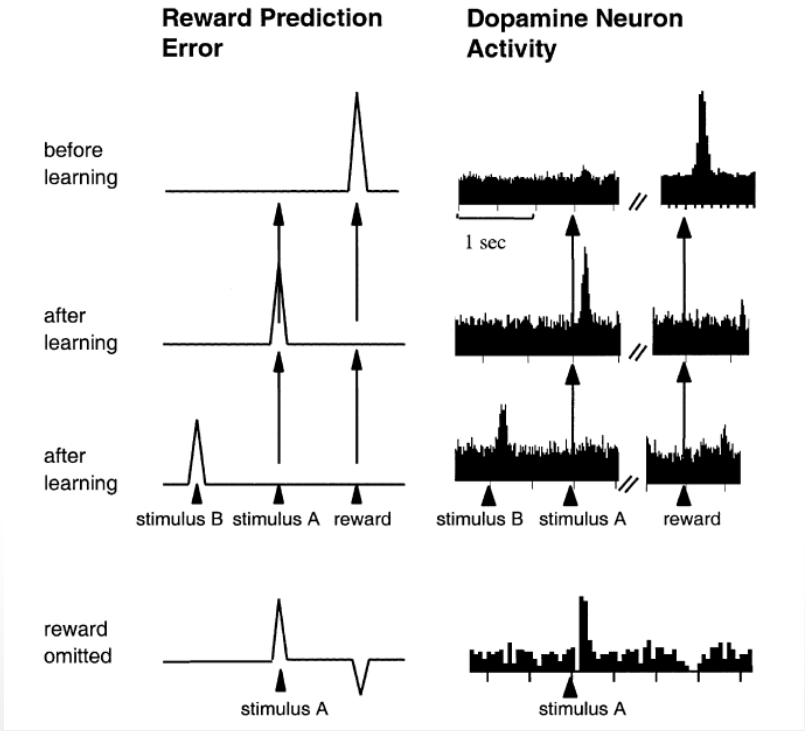
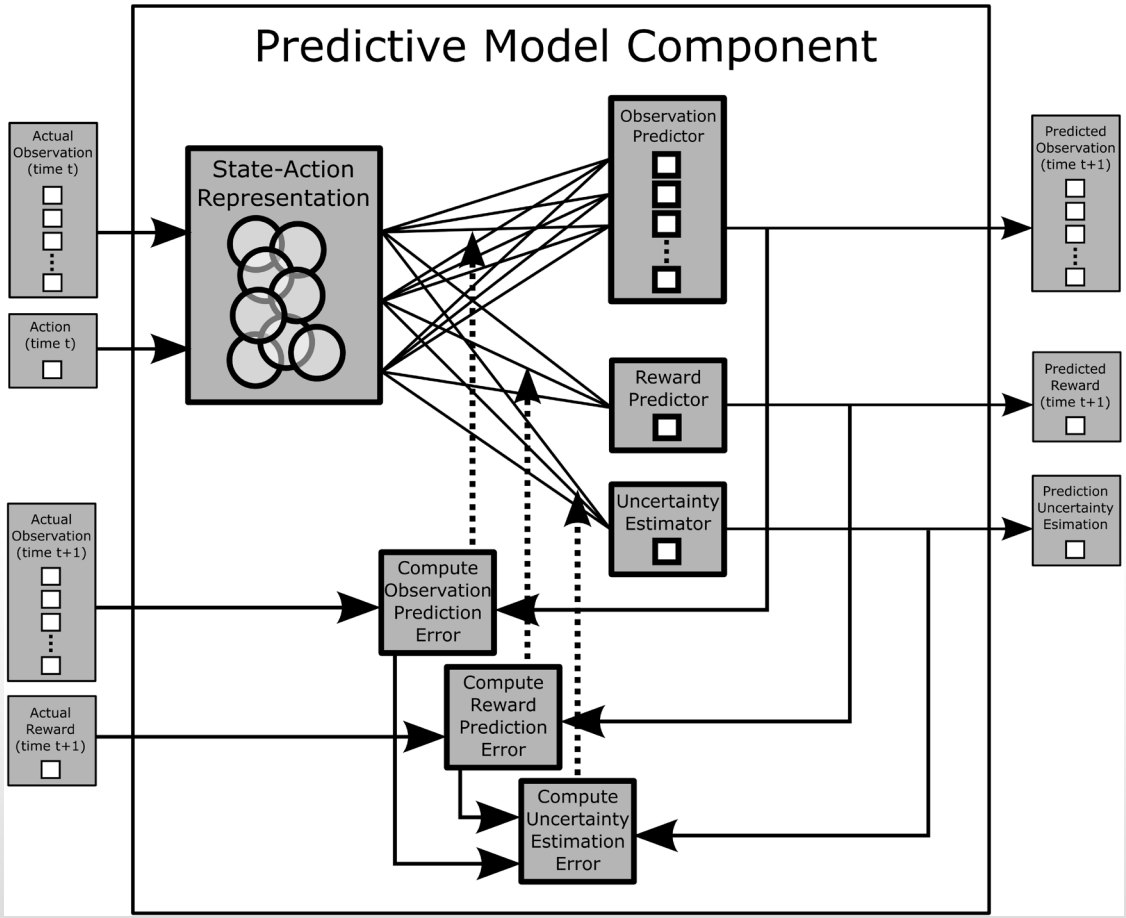


Figure from Suri, R.E. (2002). TD Models of Reward Predictive Responses in Dopamine Neurons. *Neural Networks*, 15:523-533.

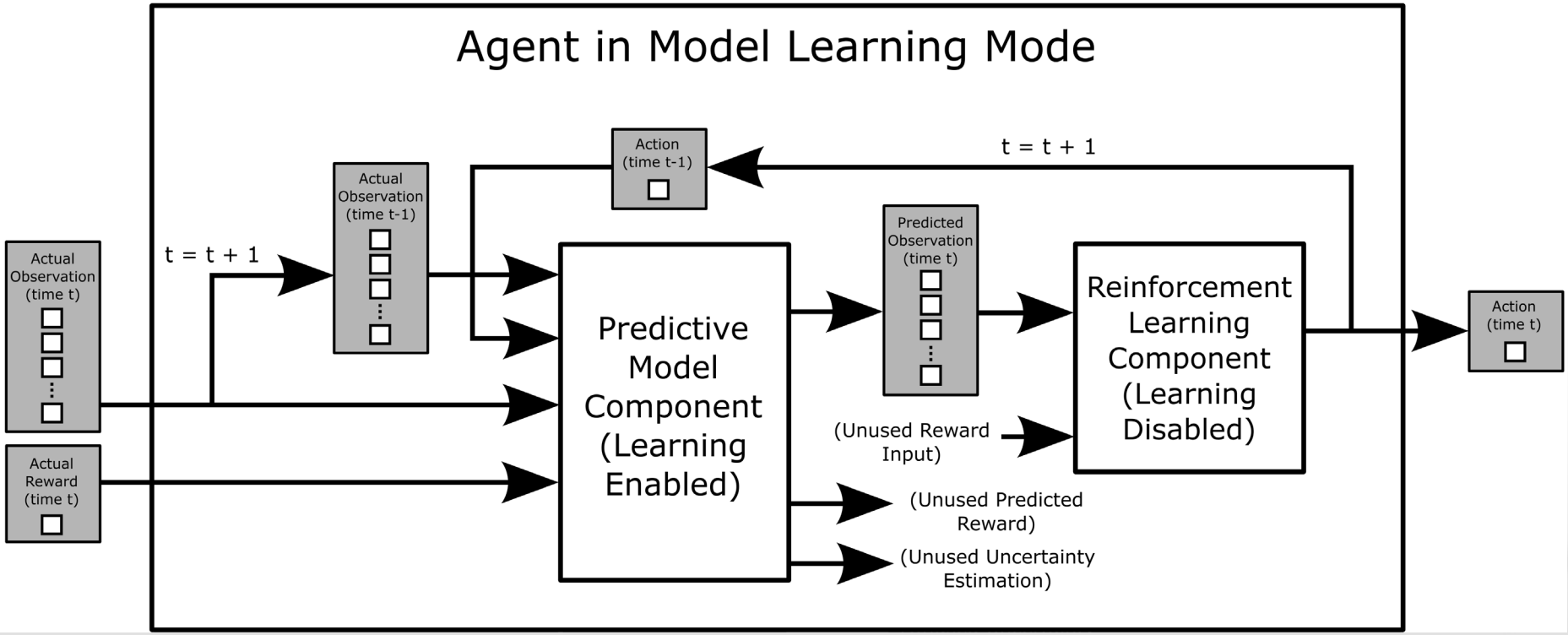
Internal Predictive Model

An accurate predictive model can temporarily replace actual experience from the environment.



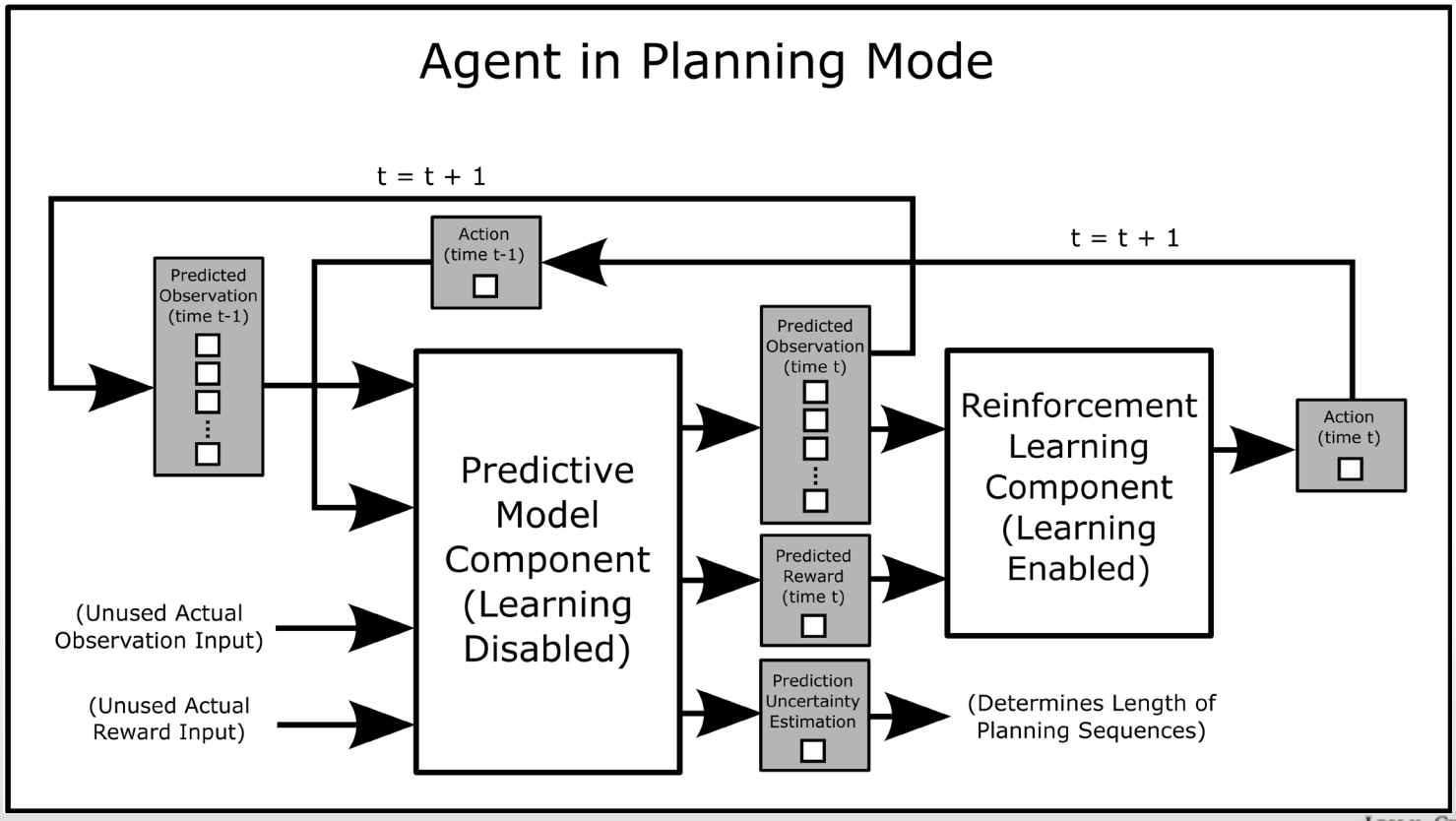
Training a Predictive Model

- Given the previous observation and action, the predictive model tries to predict the current observation and reward.
- Training signals are computed from the error between actual and predicted information.



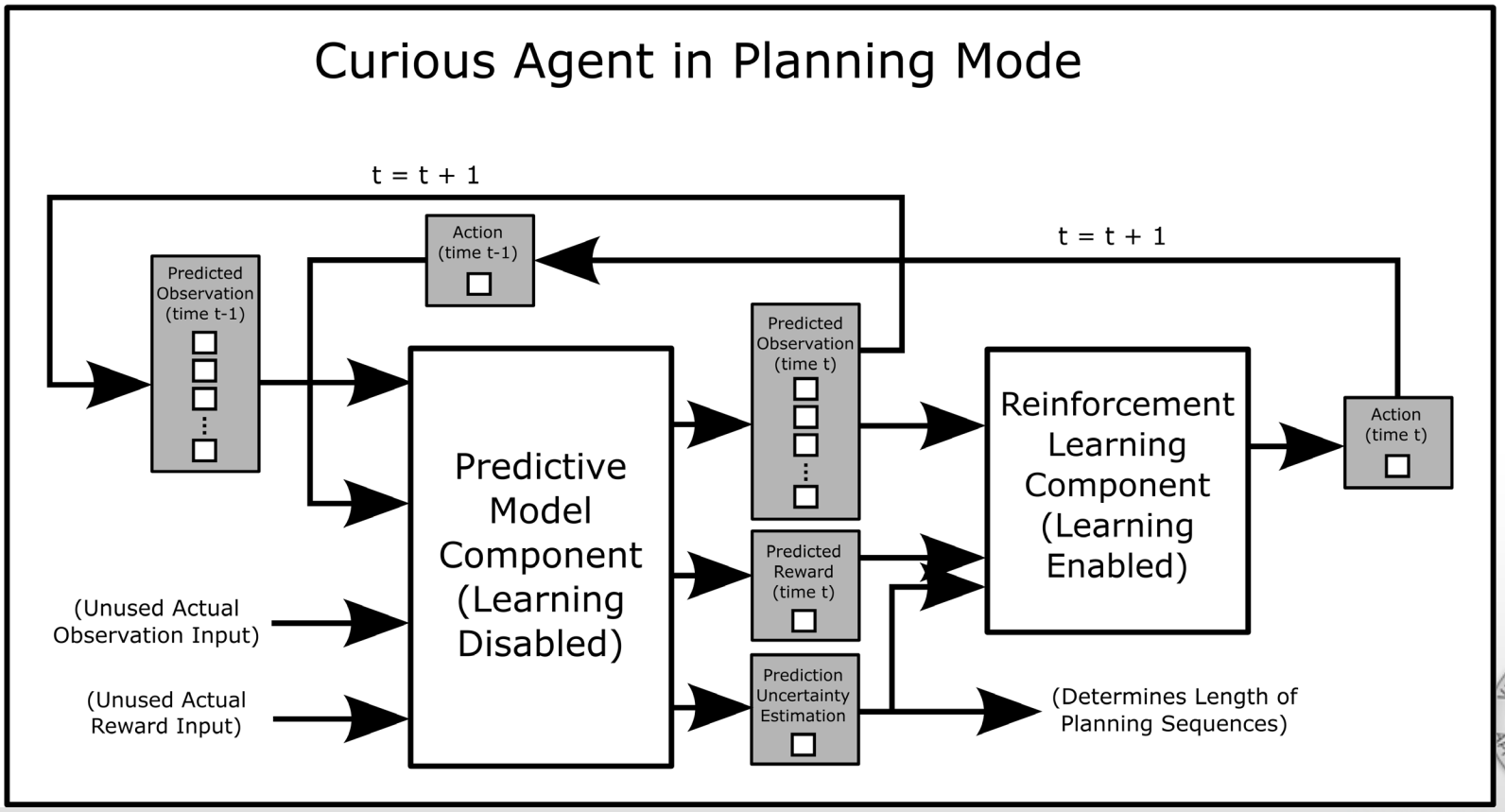
Planning

- Reinforcement learning from simulated experiences.
- Planning sequences end when uncertainty is too high.



Curiosity Rewards

- Intrinsic drive to explore unfamiliar states.
- Provide extra rewards proportional to uncertainty.

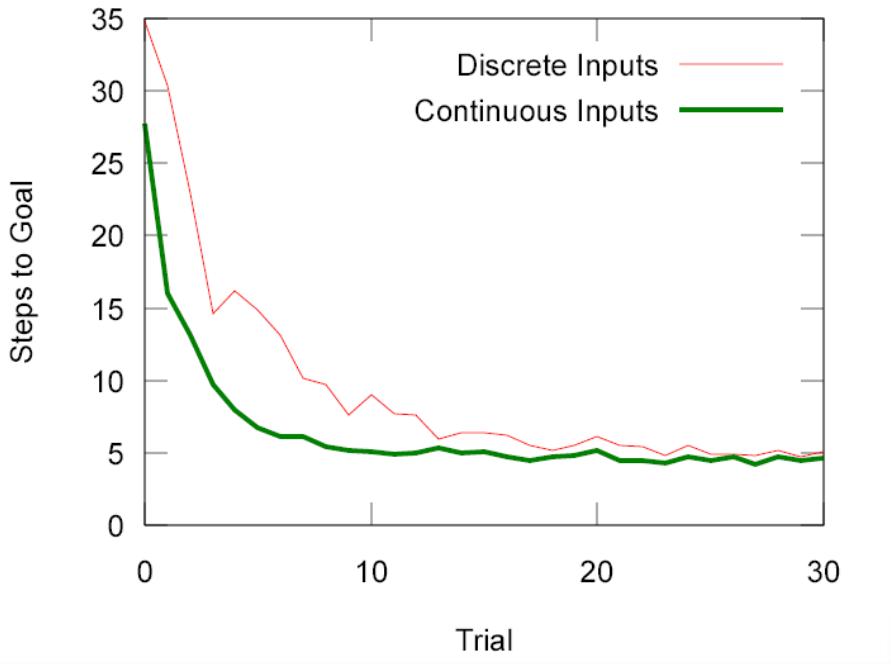


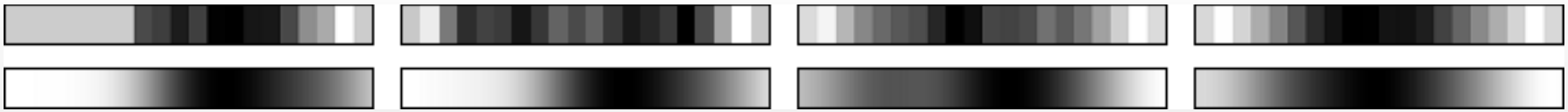
Verve

- Verve is an Open Source implementation of curious, planning reinforcement learning agents.
- Intended to be an out-of-the-box solution, e.g. for game development or robotics.
- Distributed as a cross-platform library written in C++.
- Agents can be saved to and loaded from XML files.
- Includes Python bindings.

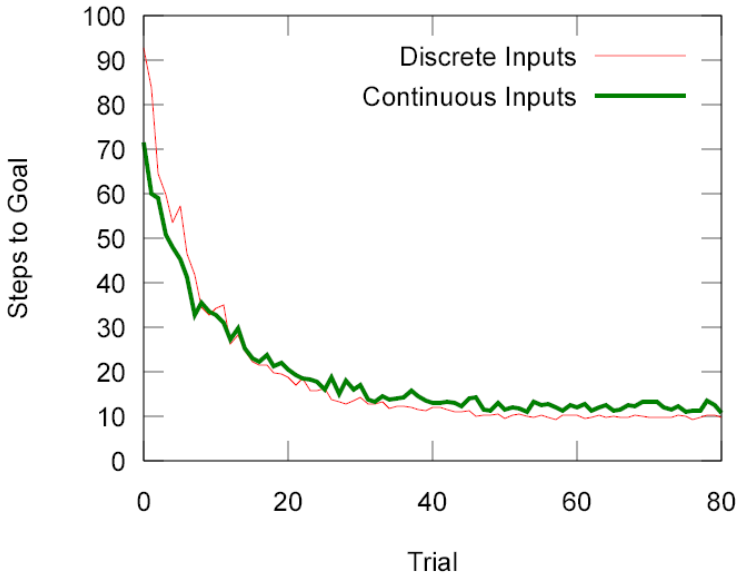
http://verve-agents.sourceforge.net

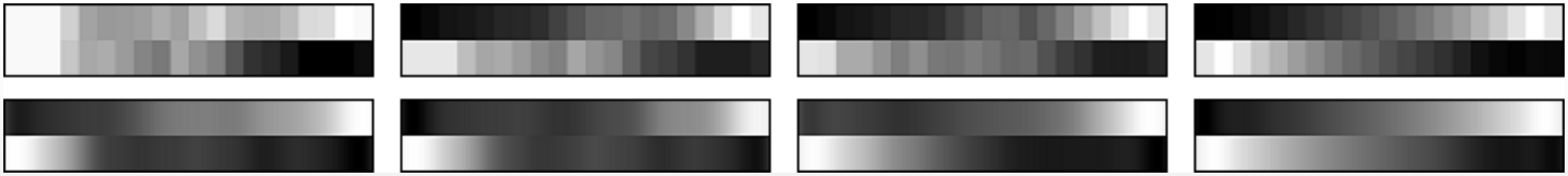
1D Hot Plate Task



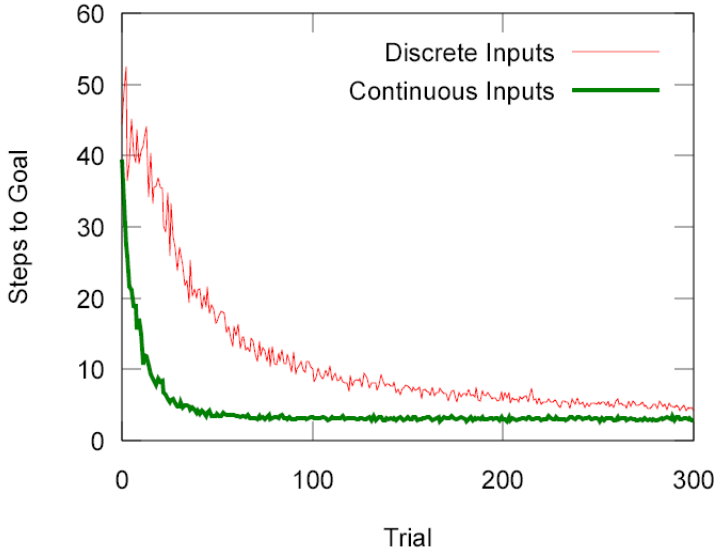


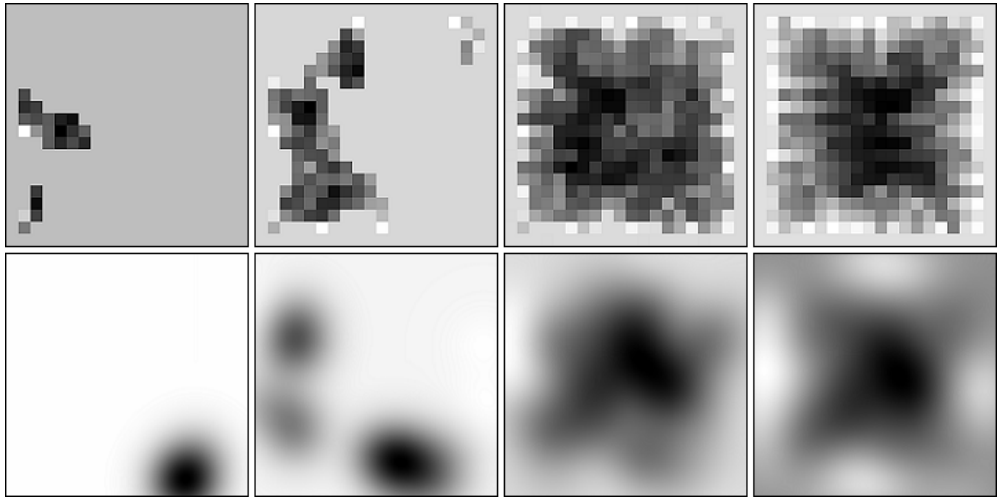
1D Signaled Hot Plate Task

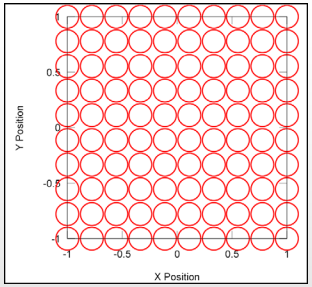


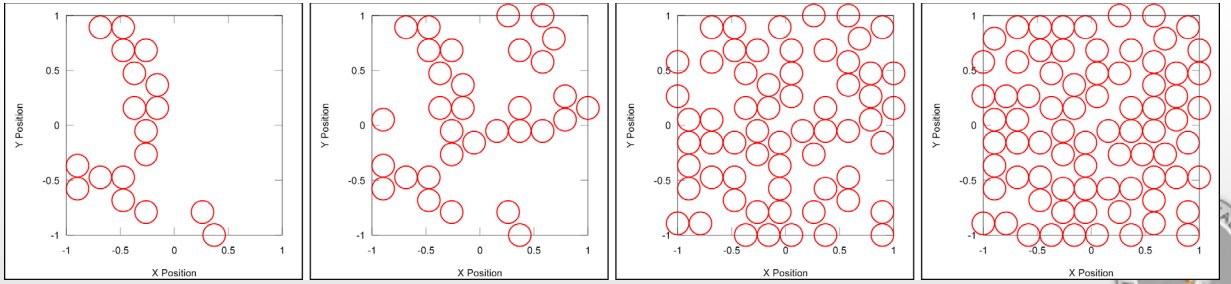


2D Hot Plate Task

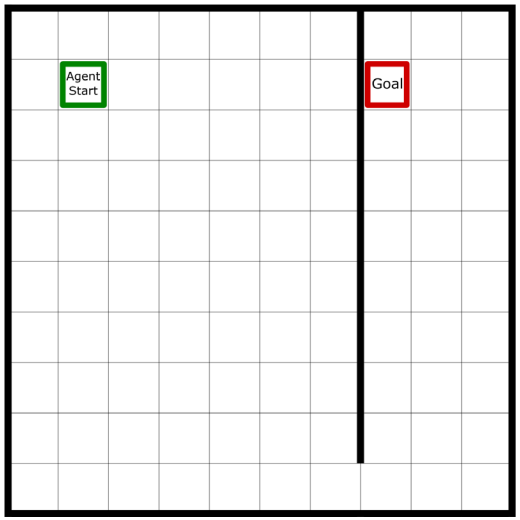


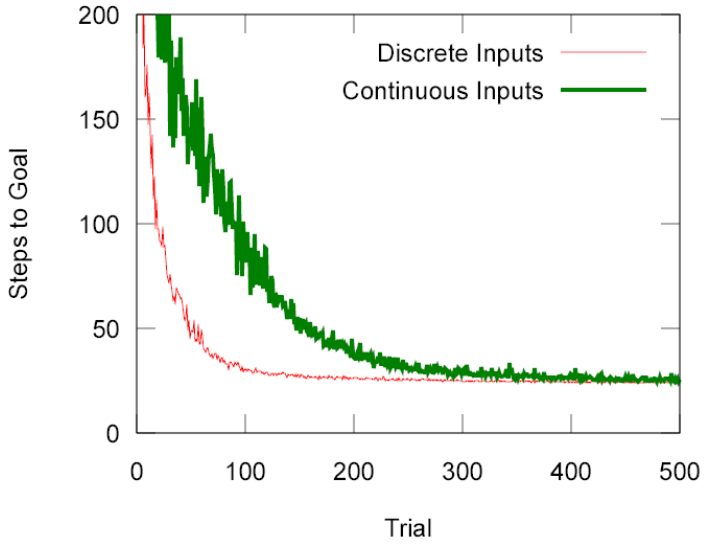


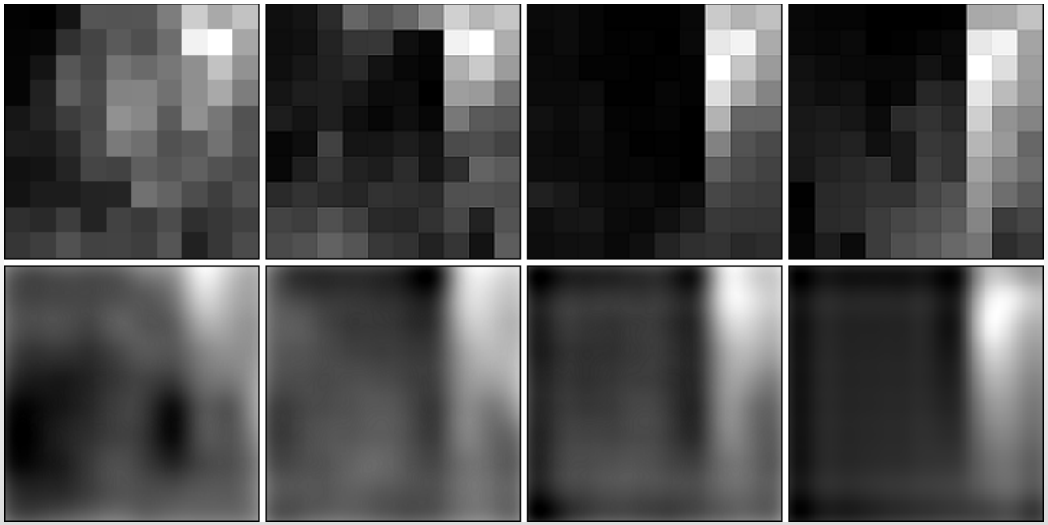




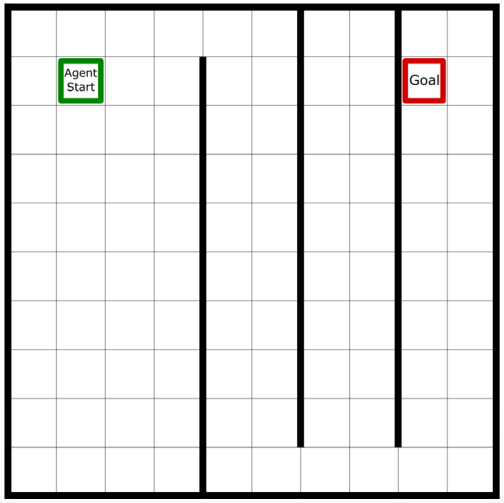
2D Maze Task #1

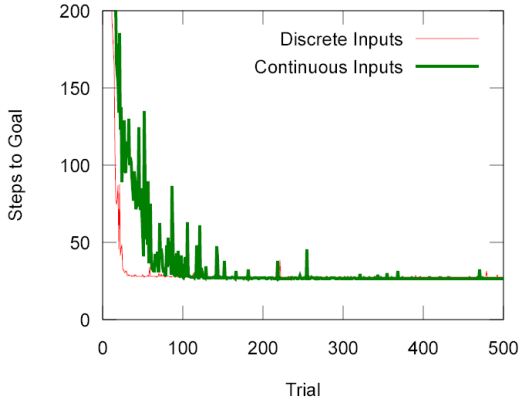


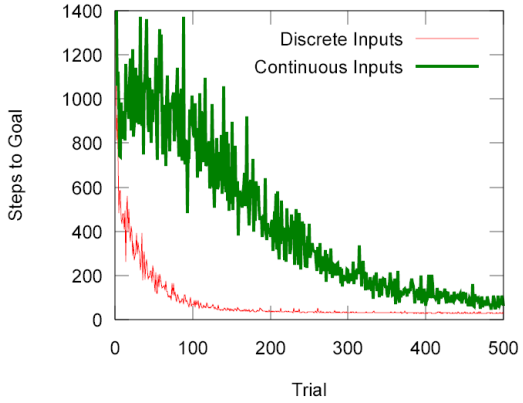


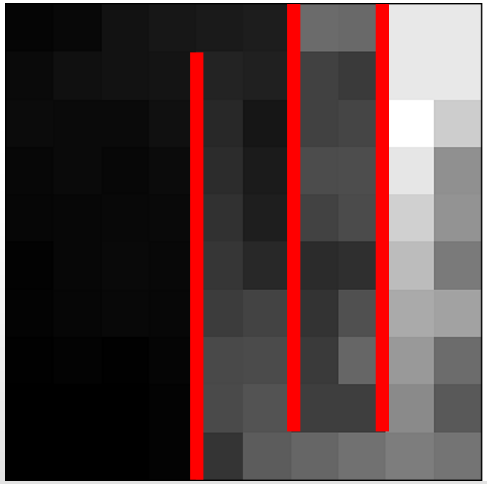


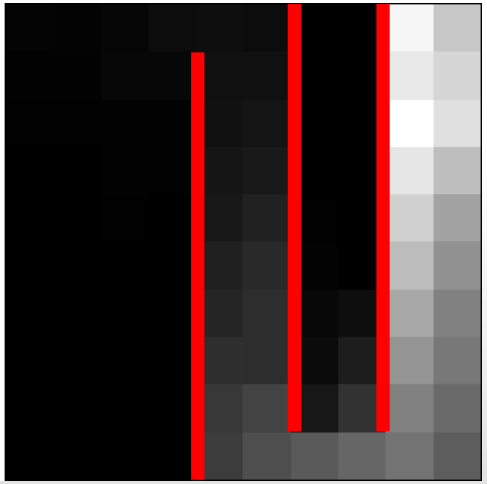
2D Maze Task #2



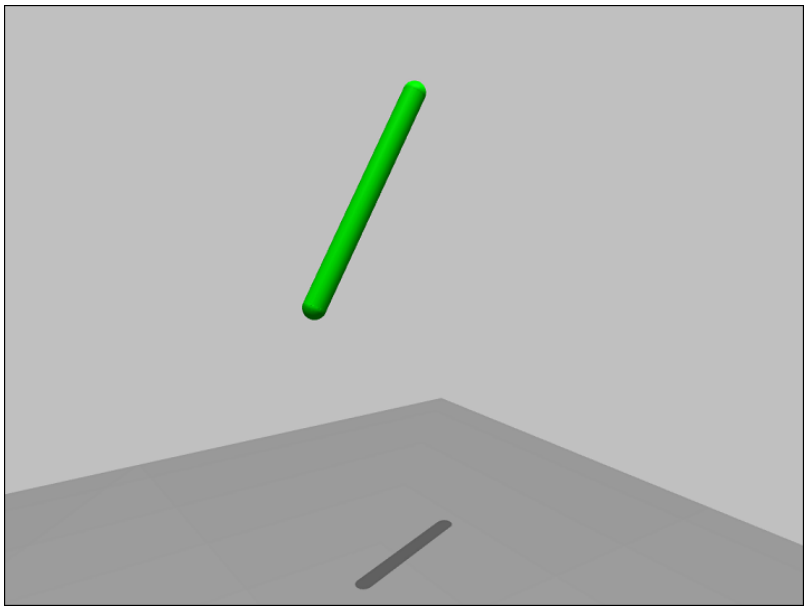


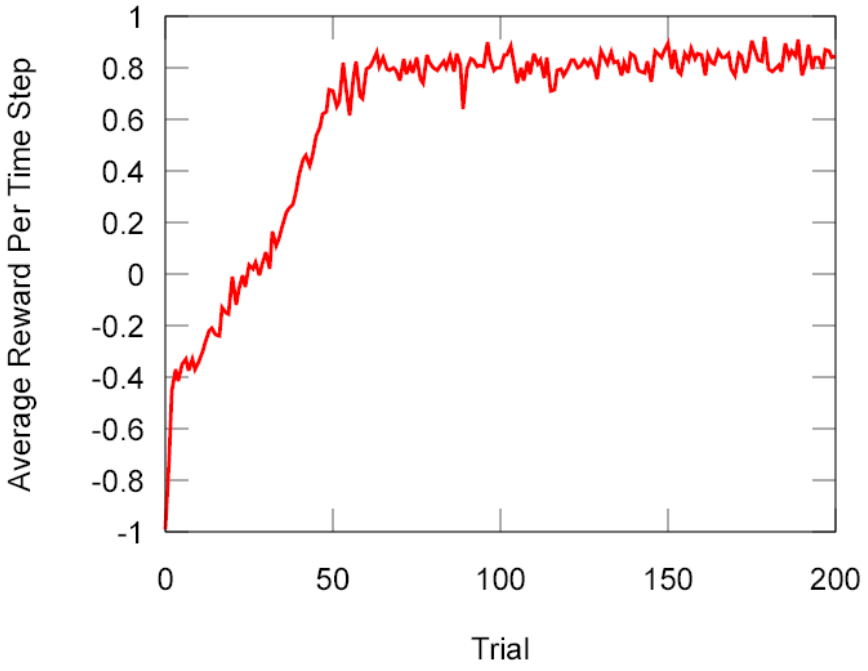


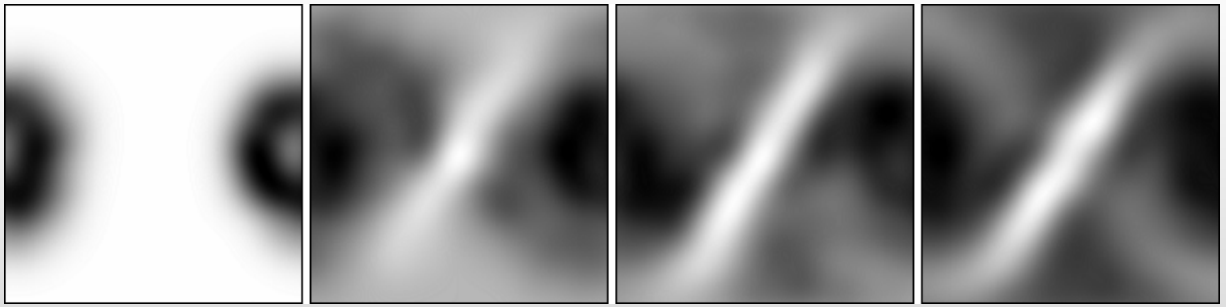




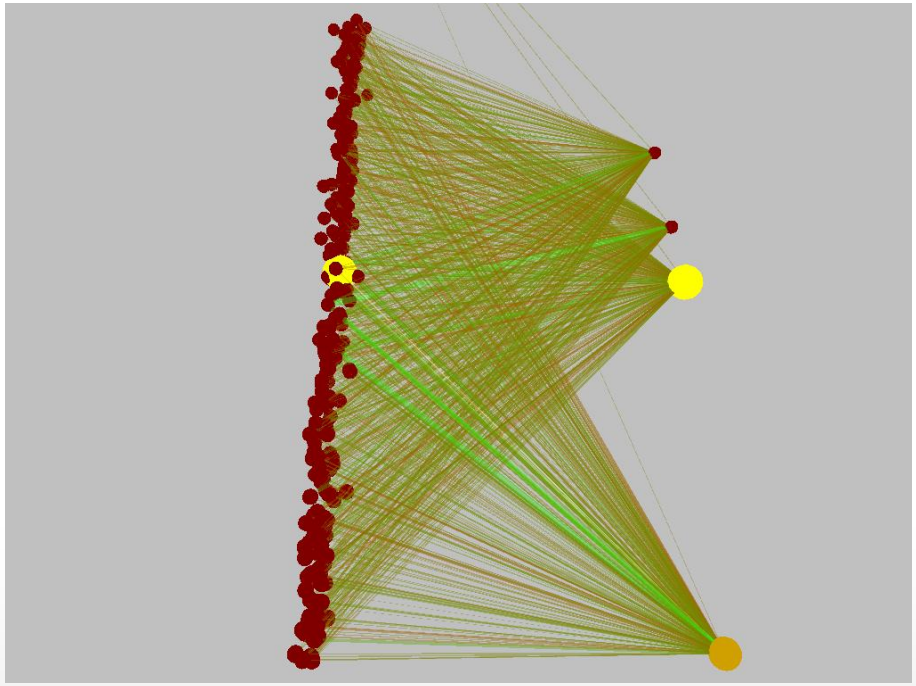
Pendulum Swing-Up Task

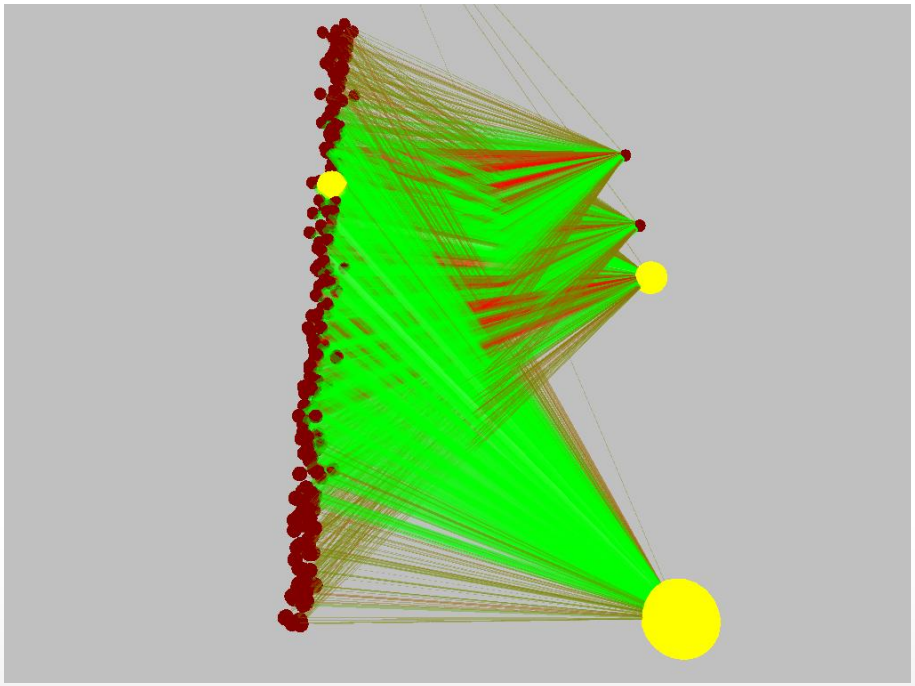




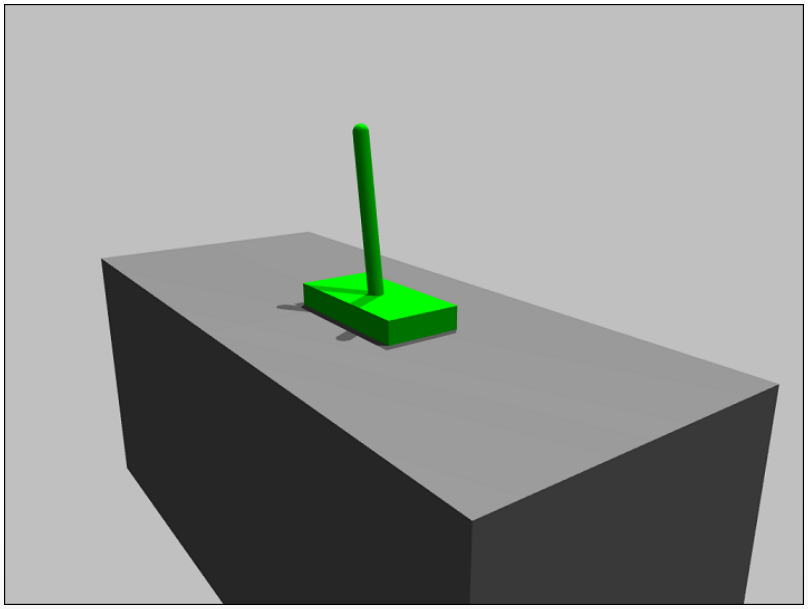


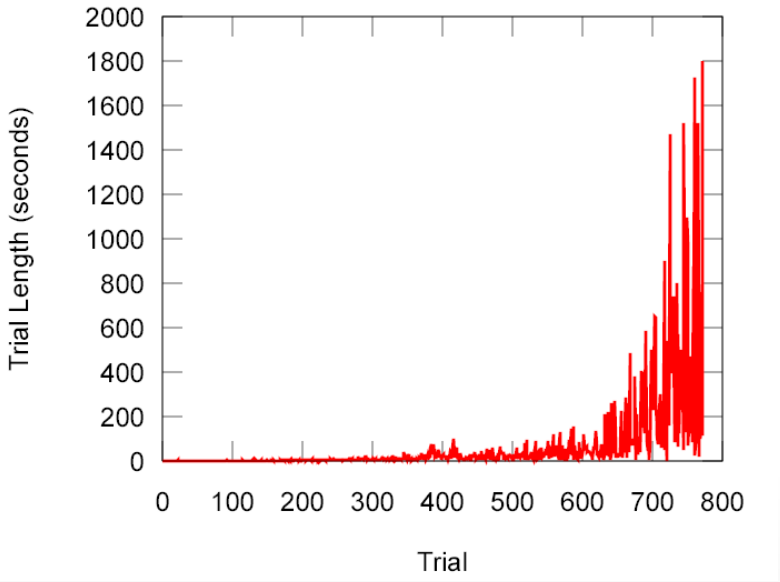
Pendulum Neural Networks



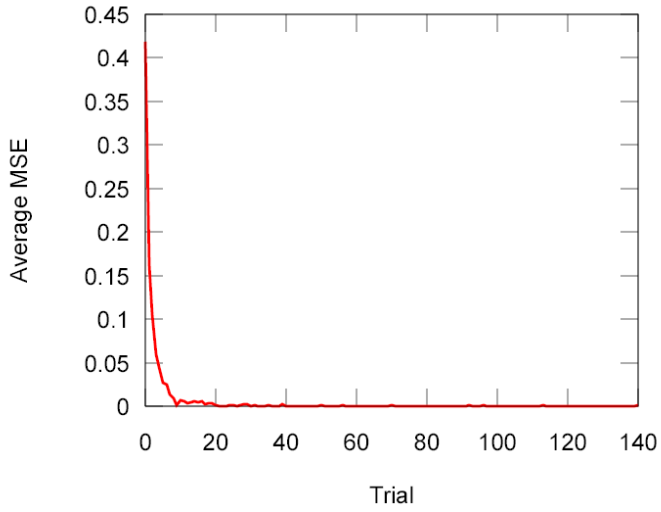


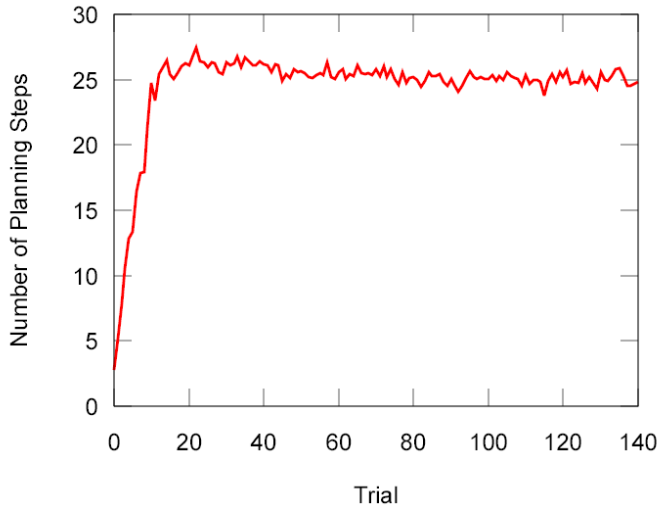
Cart-Pole/Inverted Pendulum Task

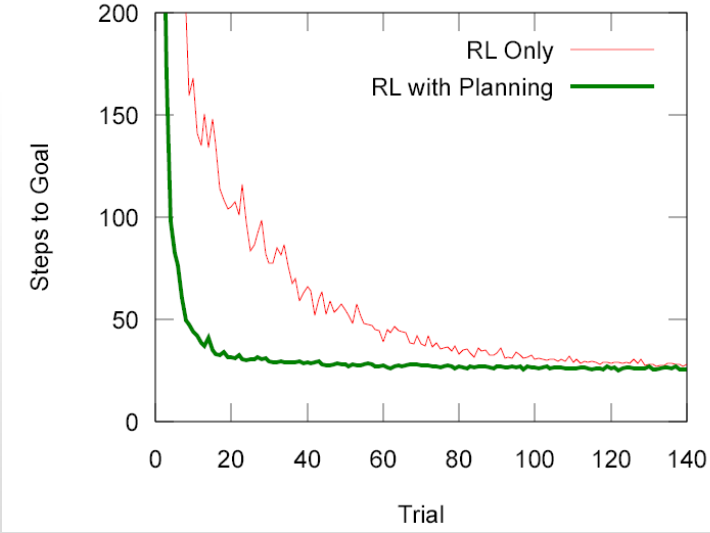




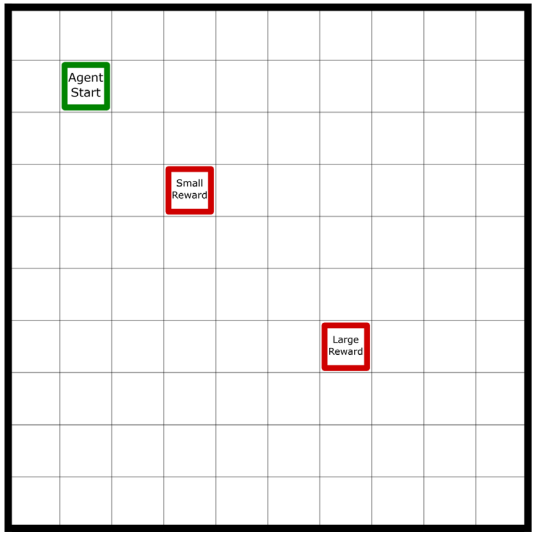
Planning in Maze #2

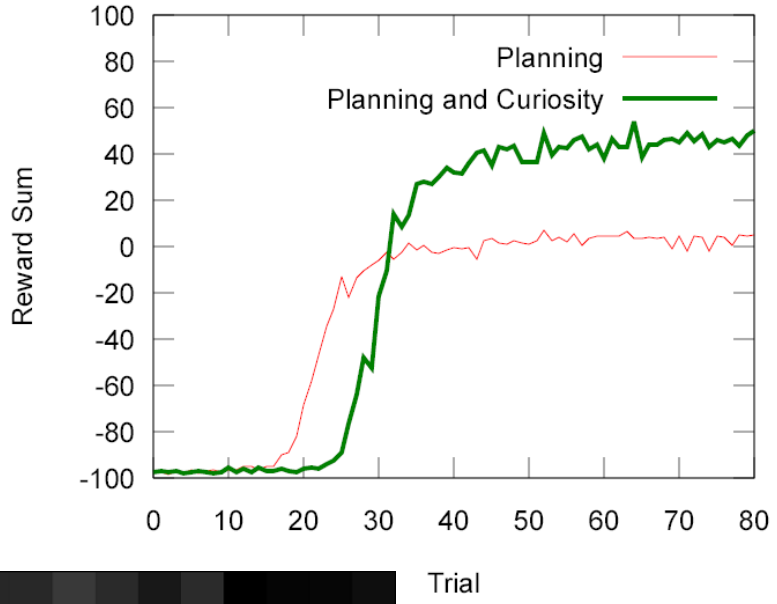


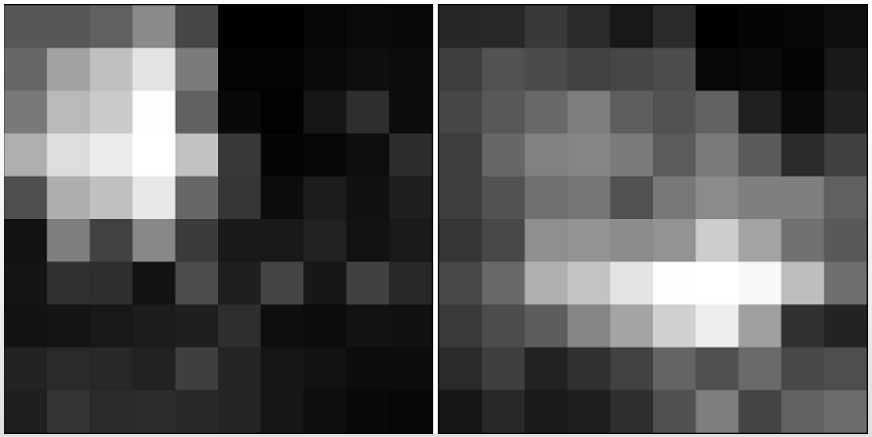




Curiosity Task







Future Work

- More applications (real robots, game development, interactive training)
- Hierarchies of motor programs
  - Constructed from low-level primitive actions
  - High-level planning
  - High-level exploration