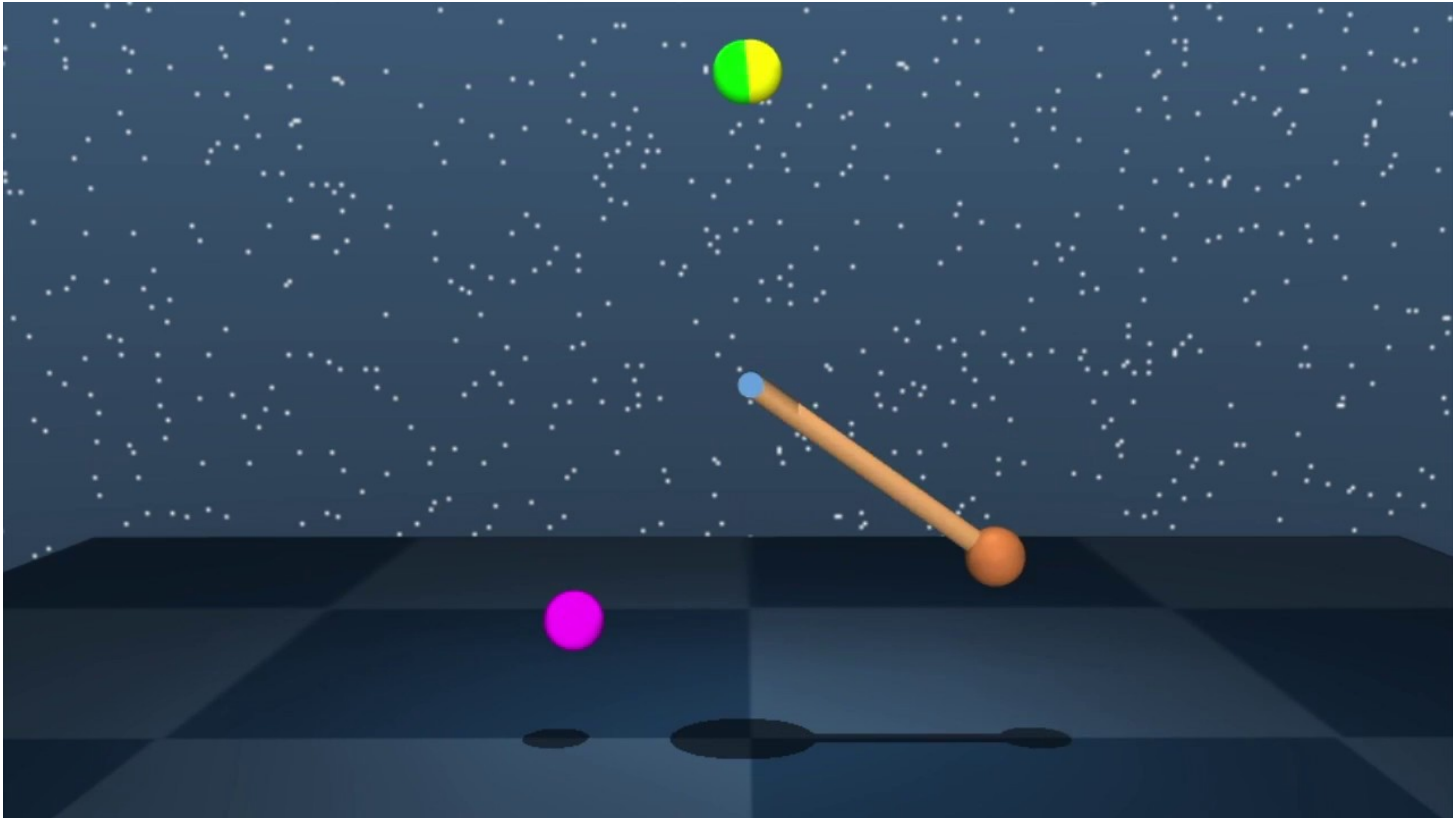
Master
Reinforcement Learning 2022
Lecture 8:
Hierarchical

Aske Plaat

Different Approaches

- Model-free
  - Value-based [2,3]
  - Policy-based [4]
- Model-based
  - Learned [5]
  - Perfect; Two-Agent [6]
- Multi-agent [7]
- Hierarchical Reinforcement Learning (Sub-goals) [8]
- Meta Learning [9]

Motivation



Overview

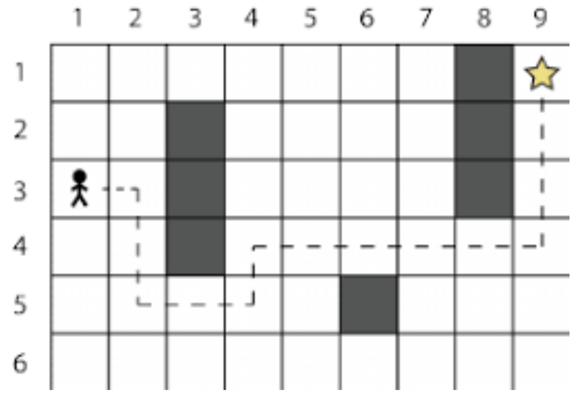
- Intuition
- Options Framework
- Tabular Algorithms
- Deep Algorithms
- Environments

Intuition

- How do you travel to a far-away friend?
  You walk to your bike, you cycle to the train station, take one or two train rides, and go to your friends house.

- Three levels of abstraction

Intuition



- How does RL do this?
  It takes a step, and another step, and another step, and another step, and another step, and another step, …

- One level

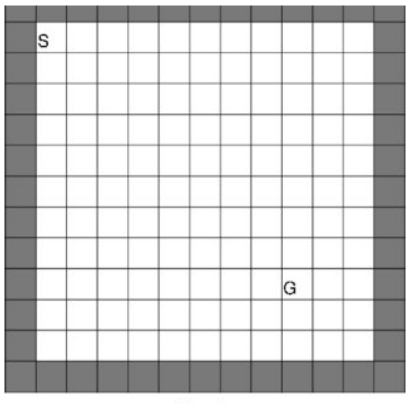
History

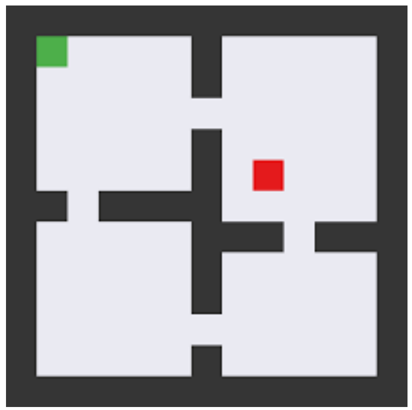
Intuition

- Hierarchical RL allows “temporal abstraction,” reasoning with actions at different time scales, short actions, and long actions, fine grain and coarse grain

Macros

- A primitive action is a regular, single-step, action
- A macro-action is any multi-step action (sub-policy), such as: go from door A to door B
- May be open-ended





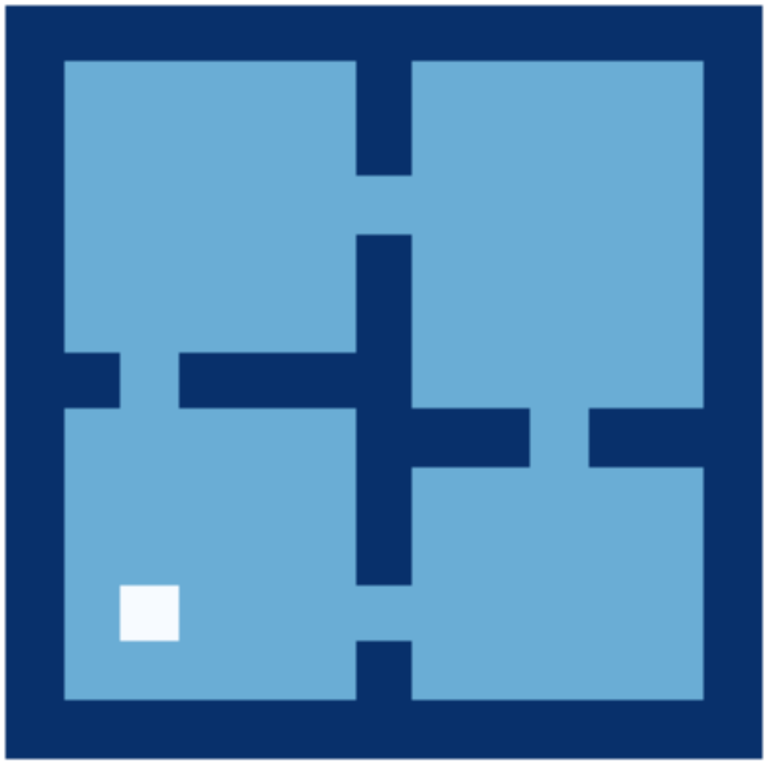
Optimality

- Note that the model-free small-step policy is likely better (more precise) than a policy incorporating a few large steps.

- HRL may be faster, but also coarser

Four Rooms

- (clip)



Options Framework

- 1999 [Sutton, Precup, Singh]
- Semi Markov Decision Process, allowing different times between actions, nested actions

- I_ω Initiation set of start states of the option
- π_ω Subpolicy of the option (the primitive actions that it consists of)
- β_ω Termination condition for each state if it terminates in that state

- All three must be provided by the programmer

Options and Actions

- Primitive Actions
- Ordinary States

- Options: **Subpolicies** of primitive actions
- **Initiation** states and **Termination** states (Goal States)
- Main Policy over Actions **and Options**

The Word "Goal"

- ***Goal*** can mean the Objective of RL
("Find the optimal policy that maximizes the expected cumulative future reward")

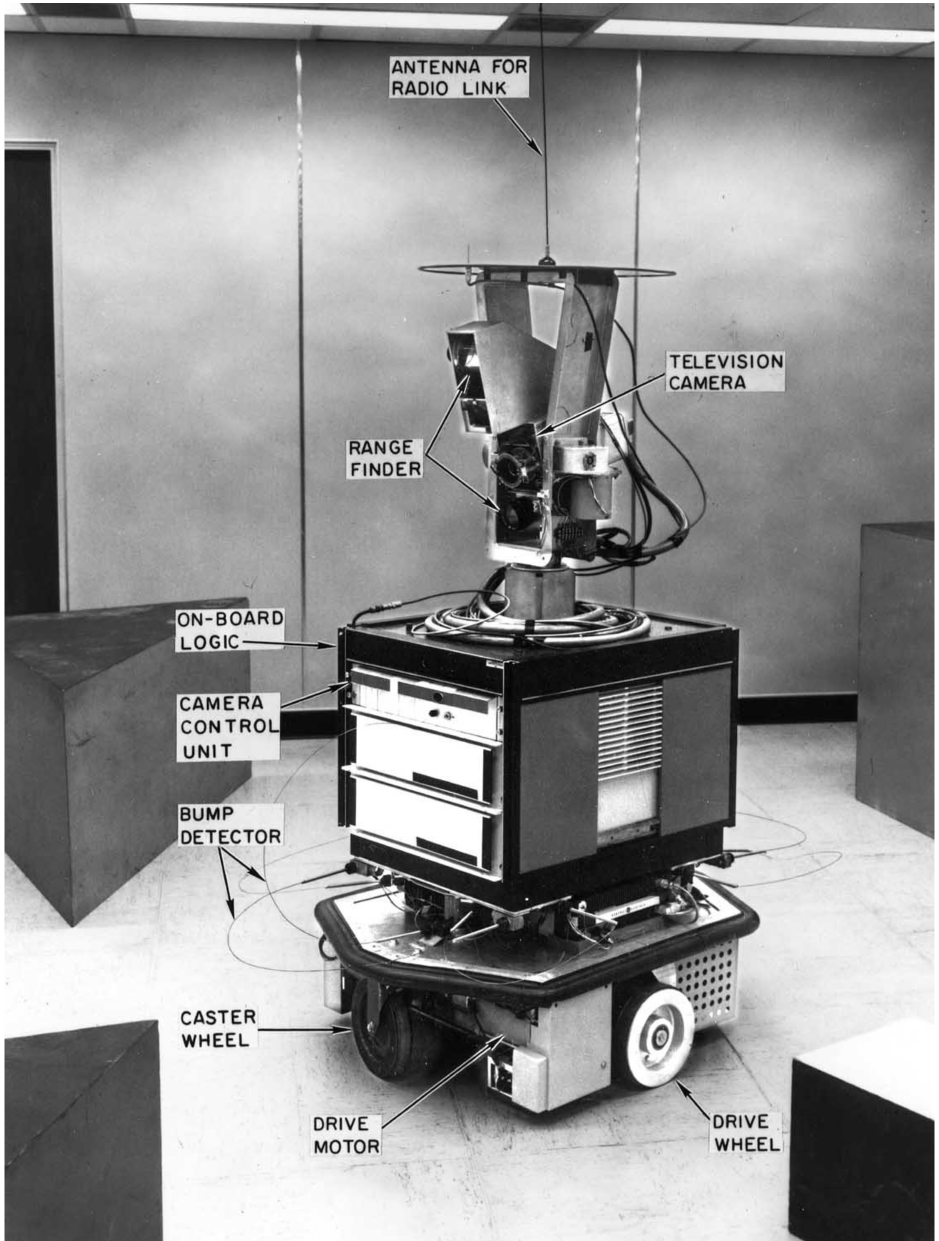
- ***Goal*** can also mean a certain State of the MDP to be reached
("The Dijkstra algorithm computes the shortest path from initial state to goal state")

Tabular Algorithms

- STRIPS
- HAM
- MAXQ
- Abstraction Hierarchies
- Relation with Planning, and thus with Model-based

STRIPS

- Stanford Research Institute Problem Solver
- [Fikes & Nilsson, 1971]
- Planning system that controlled SHAKEY, the robot
- Macros were used to create higher level subroutines
- User defines subgoals and subroutines

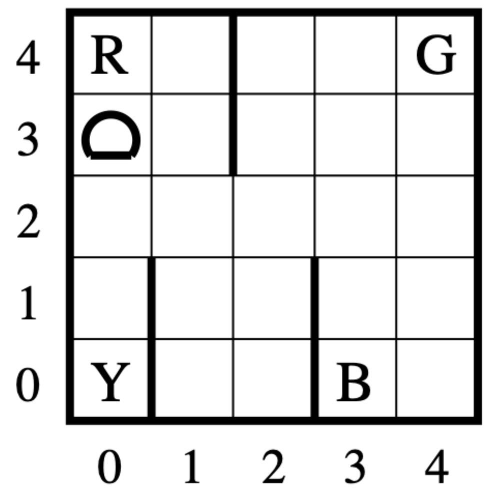


STRIPS Representation

- **States** are specified as conjunctions of predicates:
  - Start state: At(home) ∧ Sells(SM, Milk) ∧ Sells(SM, Bananas) ∧ Sells(HW, drill)
  - Goal state: At(home) ∧ Have(Milk) ∧ Have(Banana) ∧ Have(drill)
- **Actions** are described as preconditions and effects:
  - Go(x,y)
    - Precond: At(x)
    - Effect: ¬At(x) ∧ At(y)
  - Buy(x, store)
    - Precond: At(store) ∧ Sells(store, x)
    - Effect: Have(x)
- Planning as search

MAXQ

- [Thomas Dietterich, 1999]
- Hierarchical Decomposition of MDP and Value function
- Programmer defines subgoals and subpolicies
- MAXQ-Q-learning
- Introduced the Taxi domain



Abstraction Hierarchies

- [Knoblock 1994]: “Automatically generating abstractions for planning”
- Automated subgoal finding (smaller subproblems)
- However, [Backstrom & Jonsson 1995]: ”Planning with Abstraction Hierarchies can be exponentially less efficient”

Computational Problem

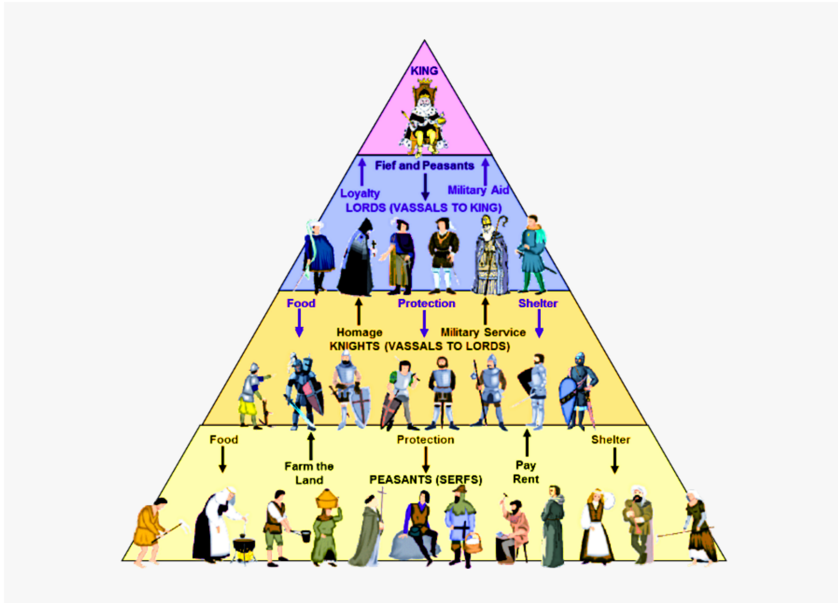
- Enumerating the state space of an MDP is exponential in the size of the problem
- Enumerating all possible subgoals, or all possible subpolicies, is also exponential in the size of the problem
- Using an exponential method to find speedups in an exponential problem may not be what you want
- Use domain knowledge [not general]
- Use deep learning

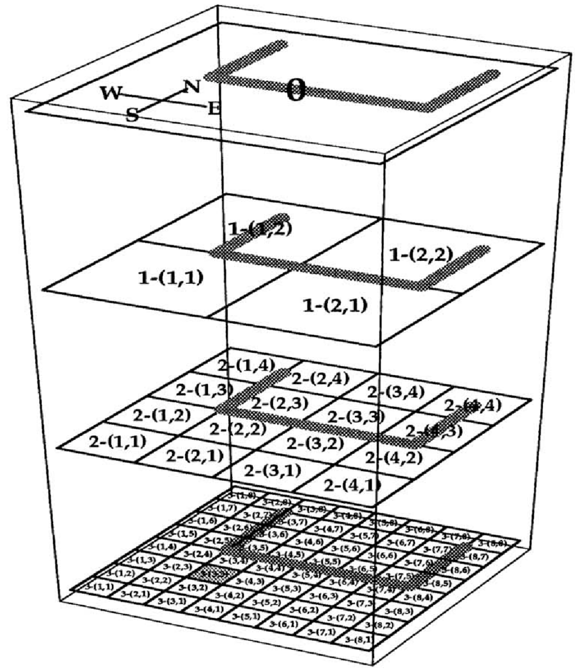
Deep Learning

- Feudal
- Option Critic
- STRAW
- HIRO
- HAC
- AMIGo
- Intrinsic IMGEP

Feudal

- [Dayan & Hinton 1993]
- Hierarchical Q-learning of sub-managers learning to satisfy demands by managers
- [Vezhnevets et al. 2017]
- FeUdal Networks, using decoupled manager and worker modules, working at different time scales



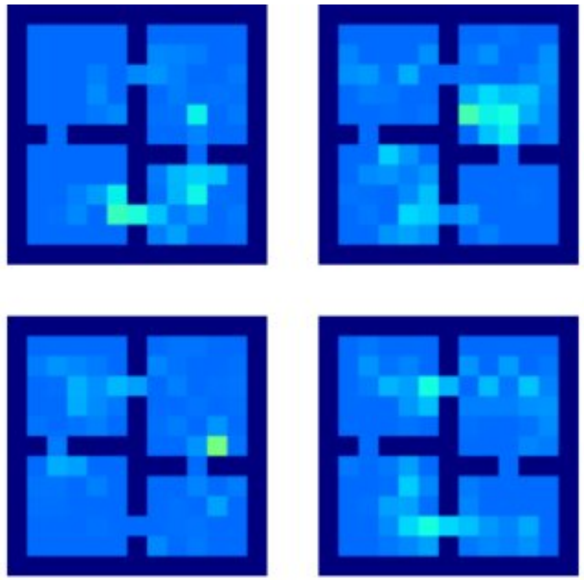


Feudal Networks

- [Vezhnevets et al. 2017]
- Manager computes latent state representation and goal vector
- LSTM
- Learning within the modules, to preserve local meaning
- Montezuma's Revenge, other ALE, DeepMind Lab
- Results show improvements over Flat RL (A3C)

Option Critic

- [Bacon et al. 2016] The Option-Critic Architecture
- Policy-gradient theorem for options, learn subpolicies and subgoals automatically
- Number of options is hyperparameter
- Good results on some ALE



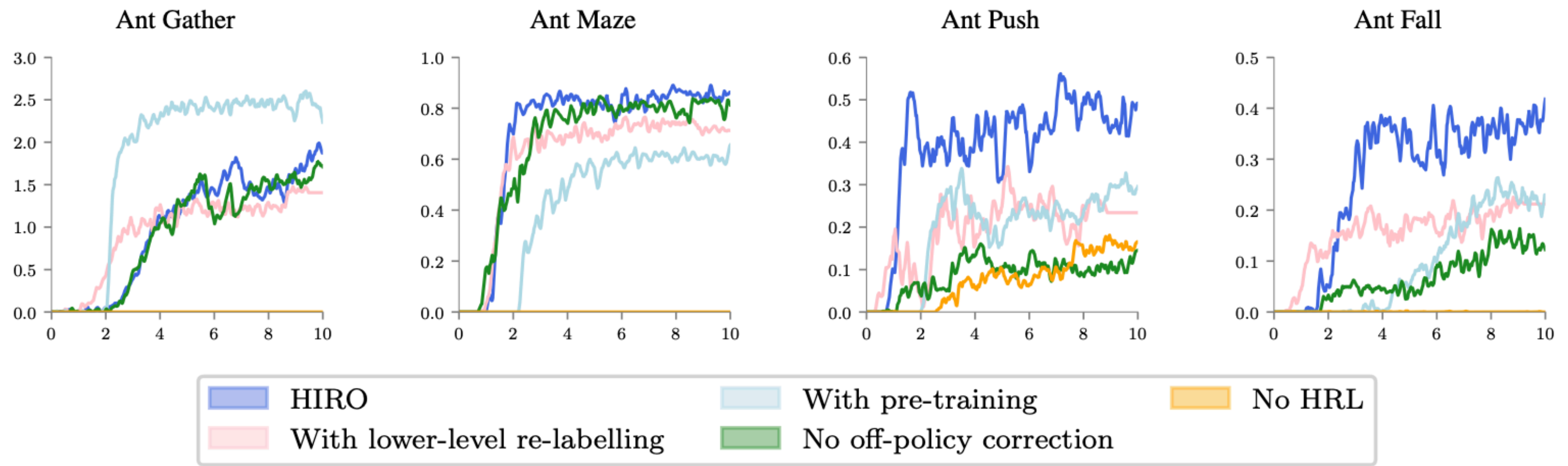
Termination Probabilities
4 options

STRAW

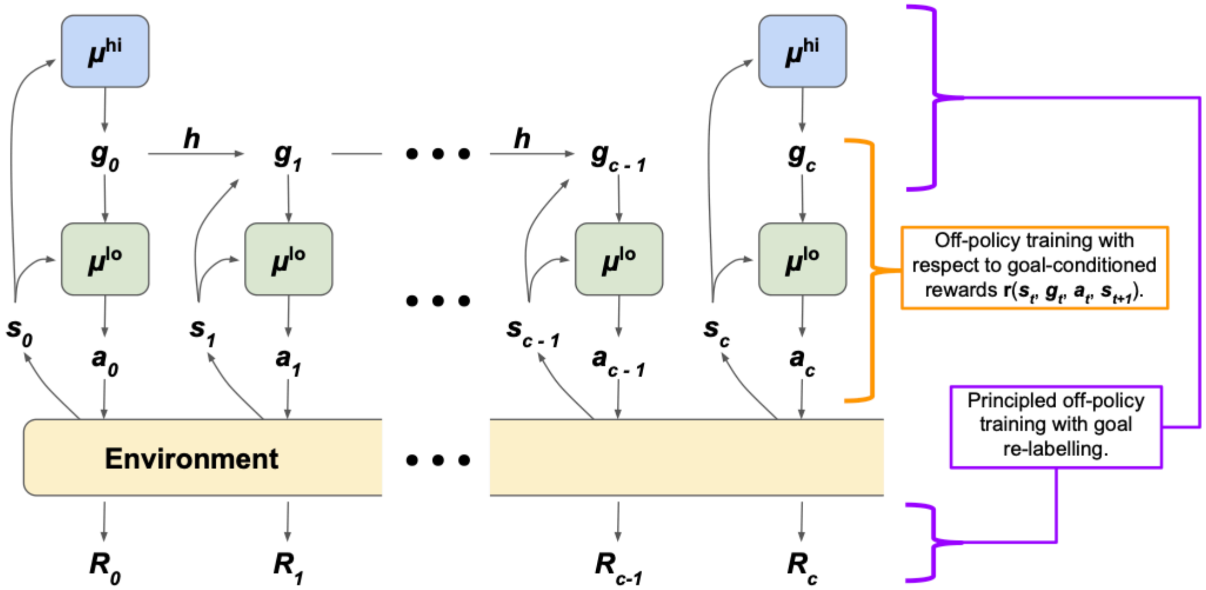
- [Vezhnevets et al. 2016] Strategic Attentive Writer for Learning Macro-Actions
- Learn implicit plans from environment
- PacMan, Frostbite
- Text problems

Hiro

- [Nachum et al. 2018] Data Efficient Hierarchical Reinforcement Learning
- Sample efficient, find subgoals and subpolicies
- Compute upper and goal-conditioned lower levels in parallel
- Off-policy (unstable lower levels)
- MuJoCo



Hiro



1. Collect experience $s_t, g_t, a_t, R_t, \ldots$.
2. Train μ^{lo} with experience transitions $(s_t, g_t, a_t, r_t, s_{t+1}, g_{t+1})$ using g_t as additional state observation and reward given by goal-conditioned function $r_t = r(s_t, g_t, a_t, s_{t+1}) = -||s_t + g_t - s_{t+1}||_2$.
3. Train μ^{hi} on temporally-extended experience $(s_t, \tilde{g}_t, \sum R_{t:t+c-1}, s_{t+c})$, where $\tilde{g}_t$ is re-labelled high-level action to maximize probability of past low-level actions $a_{t:t+c-1}$.
4. Repeat.

Figure 2: The design and basic training of HIRO. The lower-level policy interacts directly with the environment. The higher-level policy instructs the lower-level policy via high-level actions, or goals, $g_t \in \mathbb{R}^{d_s}$ which it samples anew every c steps. On intermediate steps, a fixed goal transition function h determines the next step's goal. The goal simply instructs the lower-level policy to reach specific states, which allows the lower-level policy to easily learn from prior off-policy experience.

HAC

- [Levy et al. 2017] Learning multi-level hierarchies with hindsight
- Overcomes instability of joint learning of upper and lower levels

HAC

$\underline{\Pi_i : \text{State, Goal State} \rightarrow \text{Action}}$

$\Pi_2 : \theta, \dot{\theta},$ $\rightarrow$

$\Pi_1 : \theta, \dot{\theta},$ $\rightarrow$

$\Pi_0 : \theta, \dot{\theta},$ $\rightarrow$ Joint Torques

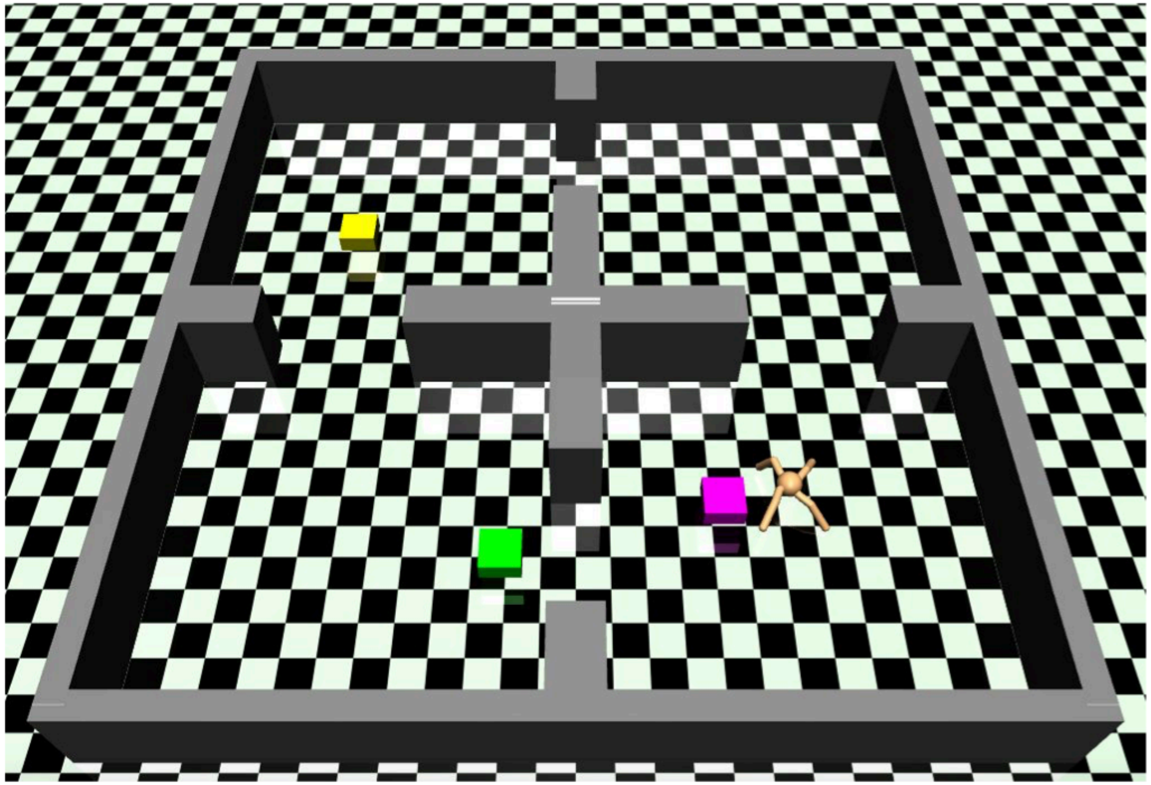


Figure 1: An ant agent uses a 3-level hierarchy to traverse though rooms to reach its goal, represented by the yellow cube. Π_2 uses as input the current state (joint positions θ and velocities $\dot{\theta}$) and goal state (yellow box) and outputs a subgoal state (green box) for Π_1 to achieve. Π_1 takes in the current state and its goal state (green box) and outputs a subgoal state (purple box) for Π_0 to achieve. Π_0 takes in the current state and goal state (purple box) and outputs a vector of joint torques.

HAC

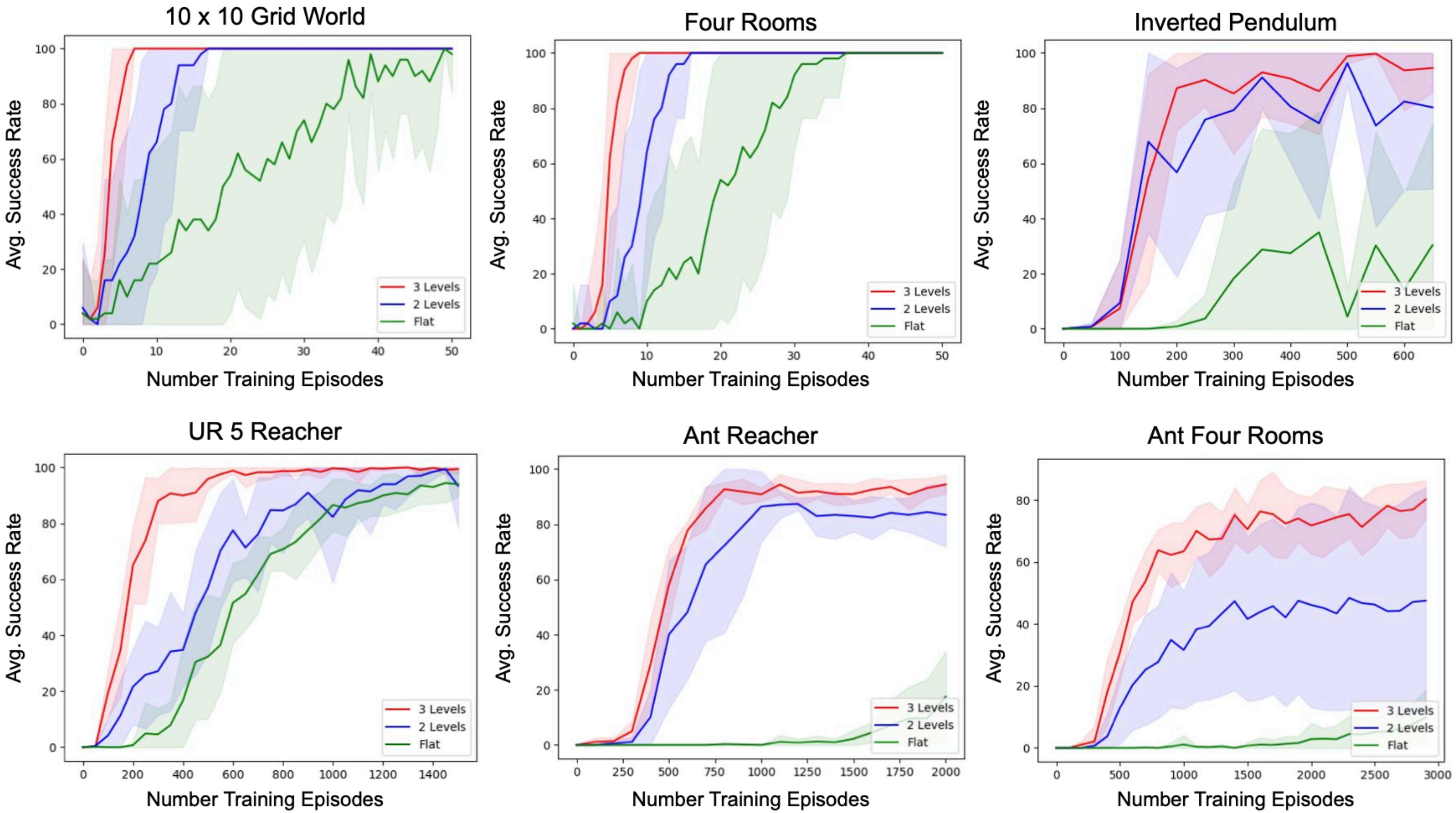


Figure 4: Average success rates for 3-level (red), 2-level agent (blue), and flat (green) agents in each task. The error bars show 1 standard deviation.

HAC/Hiro

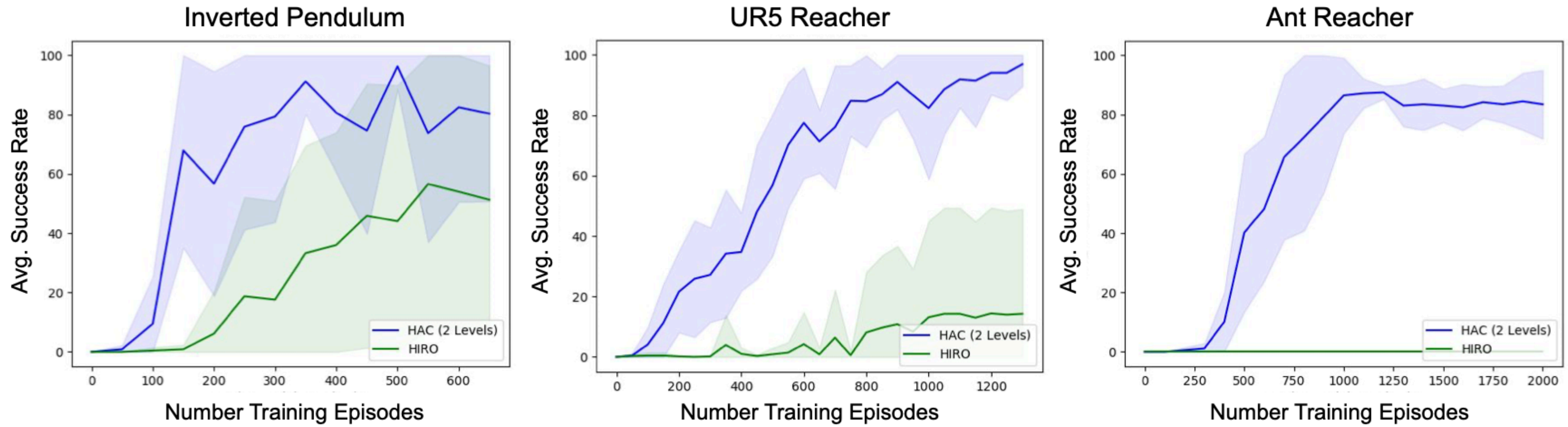


Figure 5: Figure compares the performance of HAC (2 Levels) and HIRO. The charts show the average success rate and 1 standard deviation.

AMIGo

- [Campero et al. 2021] Adversarially motivated intrinsic goals

- Teacher should learn appropriate tasks for student, not too hard, not too easy

AMIGo

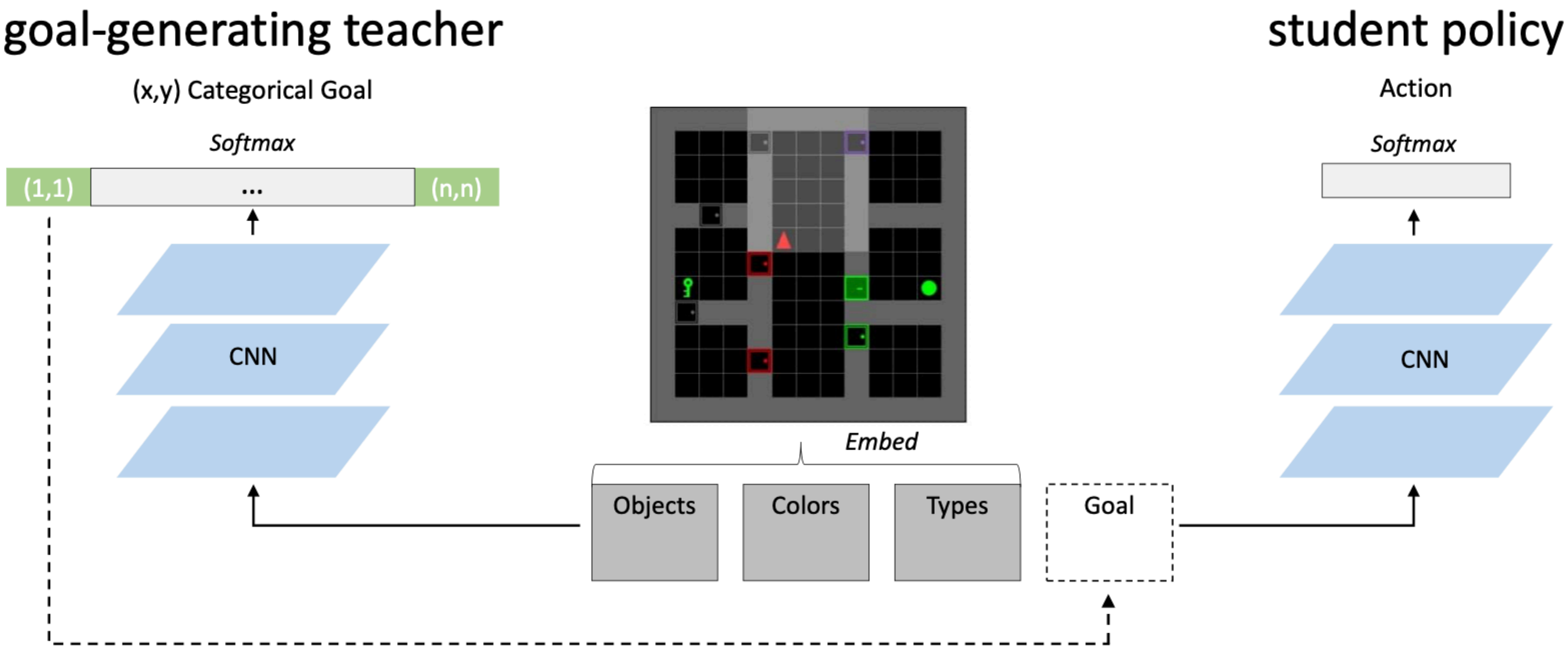


Figure 1: Training with AMIGo consists of combining two modules: a goal-generating teacher and a goal-conditioned student policy, whereby the teacher provides intrinsic goals to supplement the extrinsic goals from the environment. In our experimental set-up, the teacher is a dimensionality-preserving convolutional network which, at the beginning of an episode, outputs a location in absolute (x, y) coordinates. These are provided as a one-hot indicator in an extra channel of the student's convolutional neural network, which in turn outputs the agent's actions.

Universal Value Function

- Goal-conditioned
- Universal Value Function [Schaul et al. 2015]
- Value functions that approximate not just on state but also on goal $V_\theta(s, g)$
- Related to Multi-task learning

Intrinsic Motivation

	With *feedback*	Without *feedback*
Active	Reinforcement	Intrinsic motivation
Passive	Supervised	Unsupervised

- [Aubret et al. 2019] goal parameterization, entropy, curriculum
- [Singh et al. 2005] novelty, curiosity

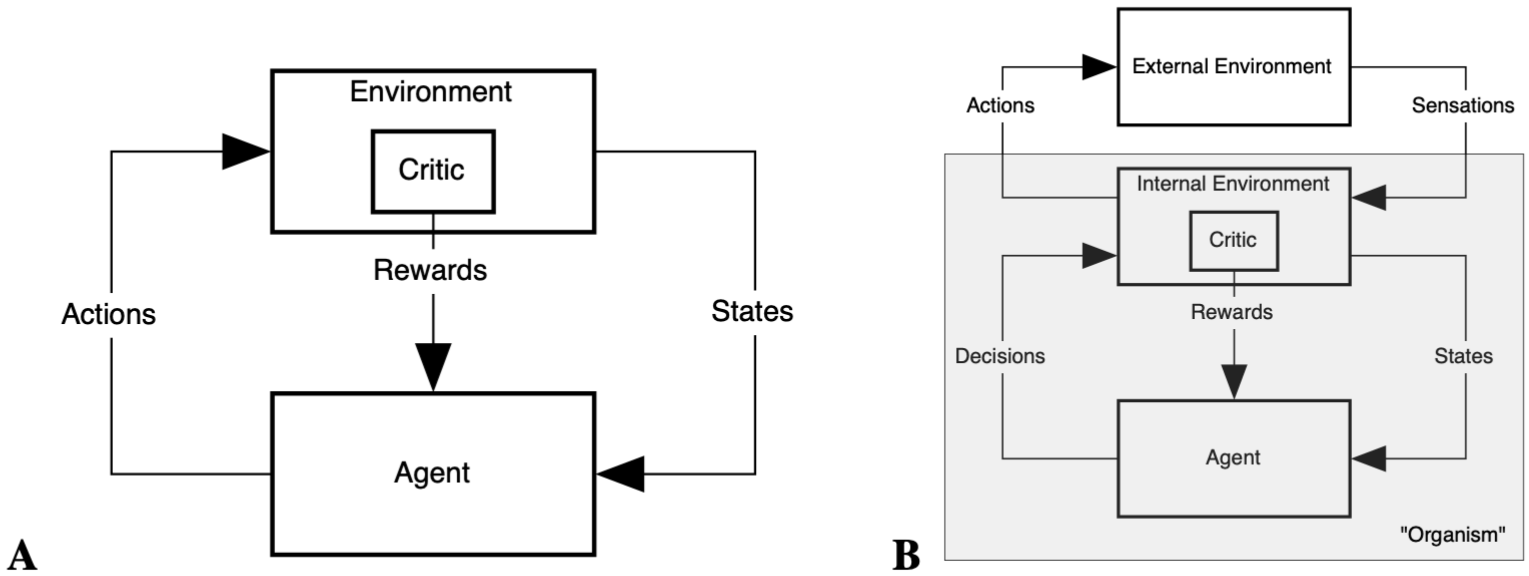


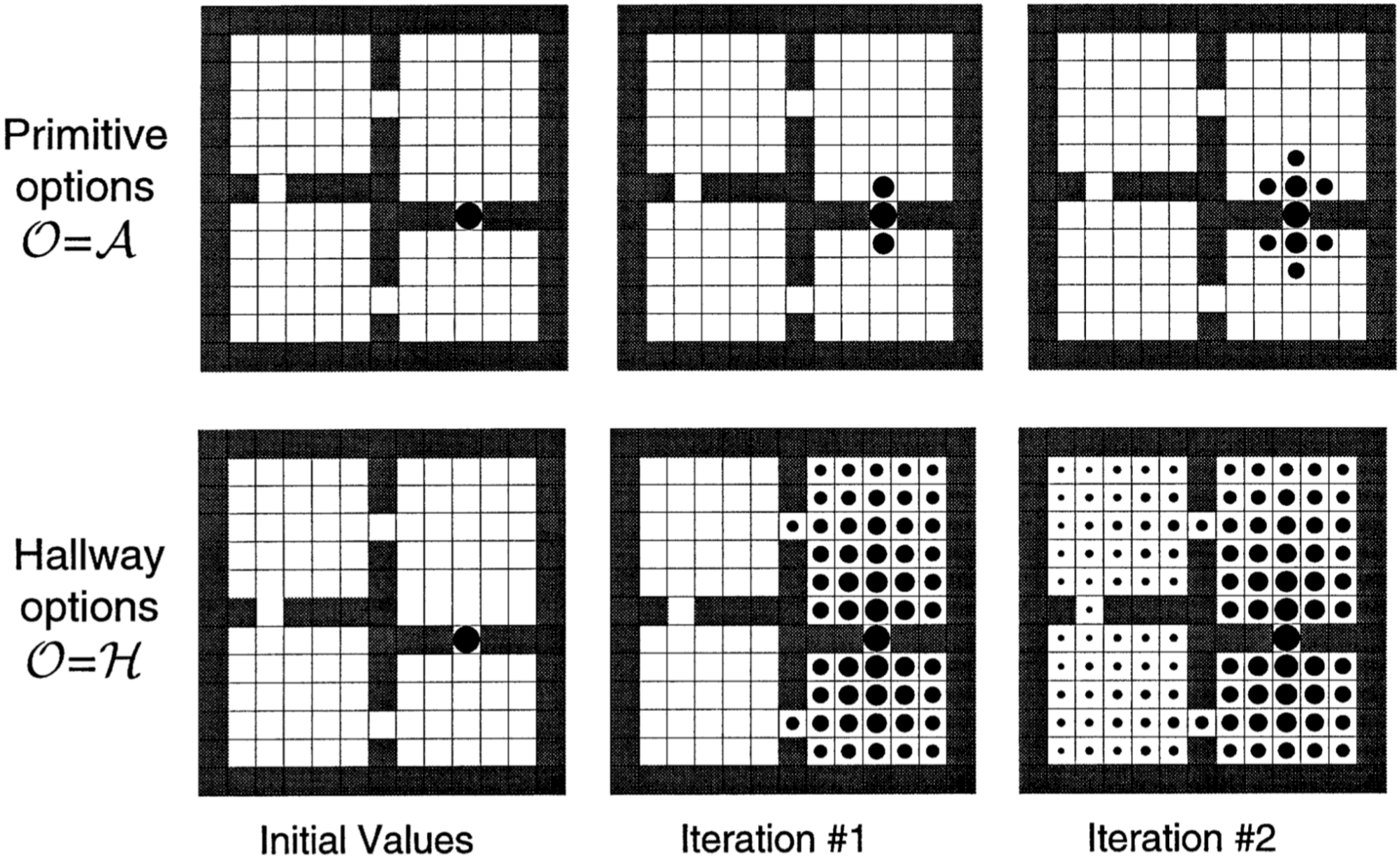
Figure 1: *Agent-Environment Interaction in RL.* **A**: *The usual view.* **B**: *An elaboration.*

What do you think?

Conclusion

- HRL follows human problem solving intuition
- Works well if subgoals and subpolicies can be easily specified with domain knowledge (domain-specific)
- Classical Tabular methods suffer from combinatorial explosion of states/subgoals and actions/subpolicies for general methods
- Some Deep Learning methods work; few epochs (expensive)
- Good match with Team/Multi-agent concepts
- Active field of research

Four Rooms



Primitive options $\mathcal{O}=\mathcal{A}$
Hallway options $\mathcal{O}=\mathcal{H}$
Initial Values
Iteration #1
Iteration #2

Montezuma's Revenge

StarCraft

Questions?