Unsupervised Methods For Subgoal Discovery During Intrinsic Motivation in Model-Free Hierarchical Reinforcement Learning

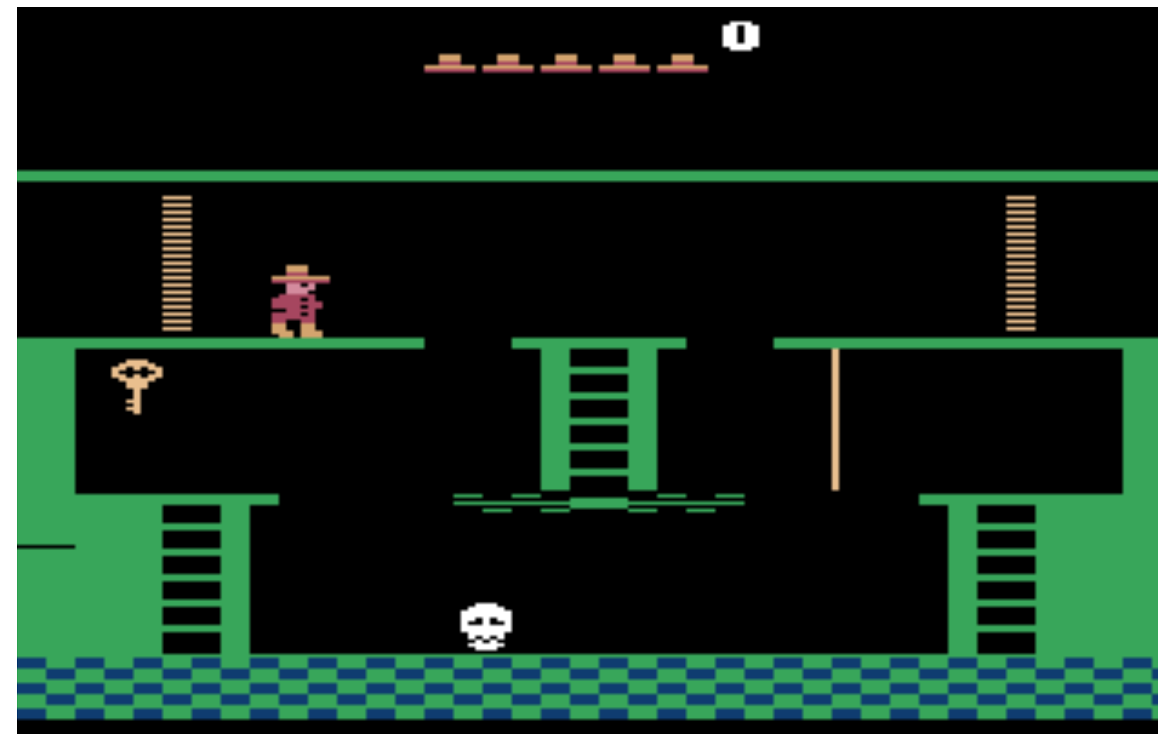
Jacob Rafati
http://rafati.net

Co-authored with: **David C. Noelle**

Ph.D. Candidate
Electrical Engineering and Computer Science
Computational Cognitive Neuroscience Laboratory
University of California, Merced

Computational Cognitive Neuroscience Laboratory
University of California, Merced

Games



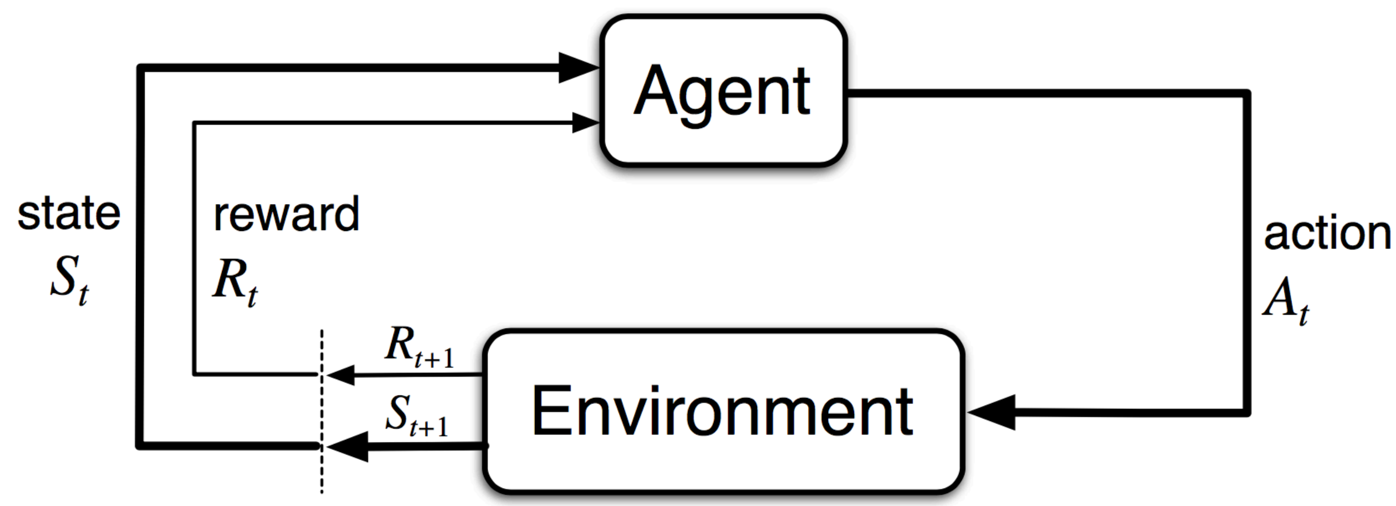
Goals & Rules

- “Key components of games are **goals**, **rules**, **challenge**, and **interaction**. Games generally involve mental or physical stimulation, and often both.”

https://en.wikipedia.org/wiki/Game

Reinforcement Learning

Reinforcement learning (RL) is learning how to map **situations (*state*)** to ***actions*** so as to maximize numerical ***reward*** signals received during the **experiences** that an artificial **agent** has as it interacts with its **environment**.

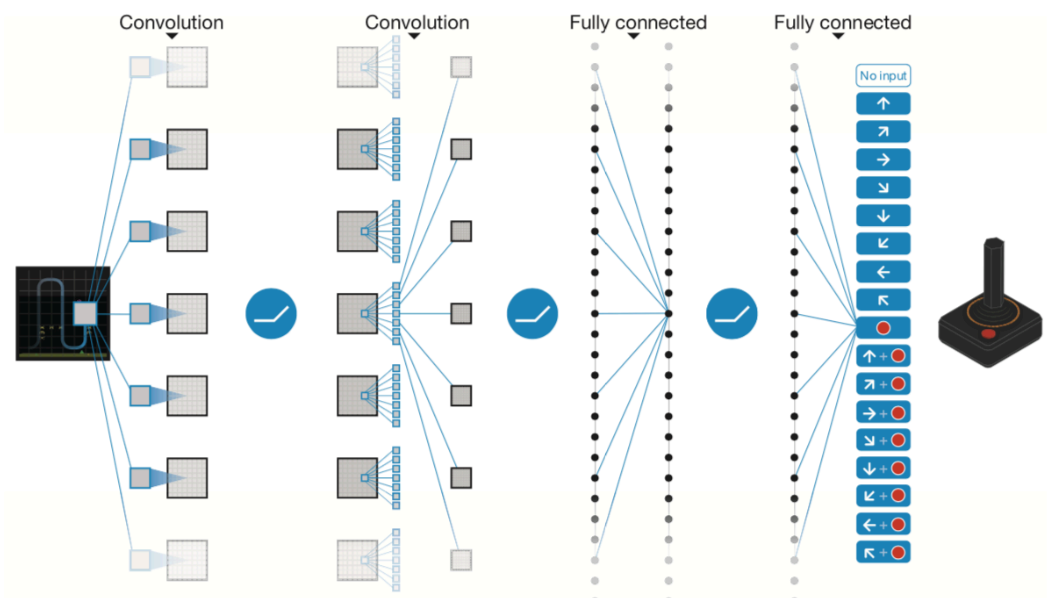


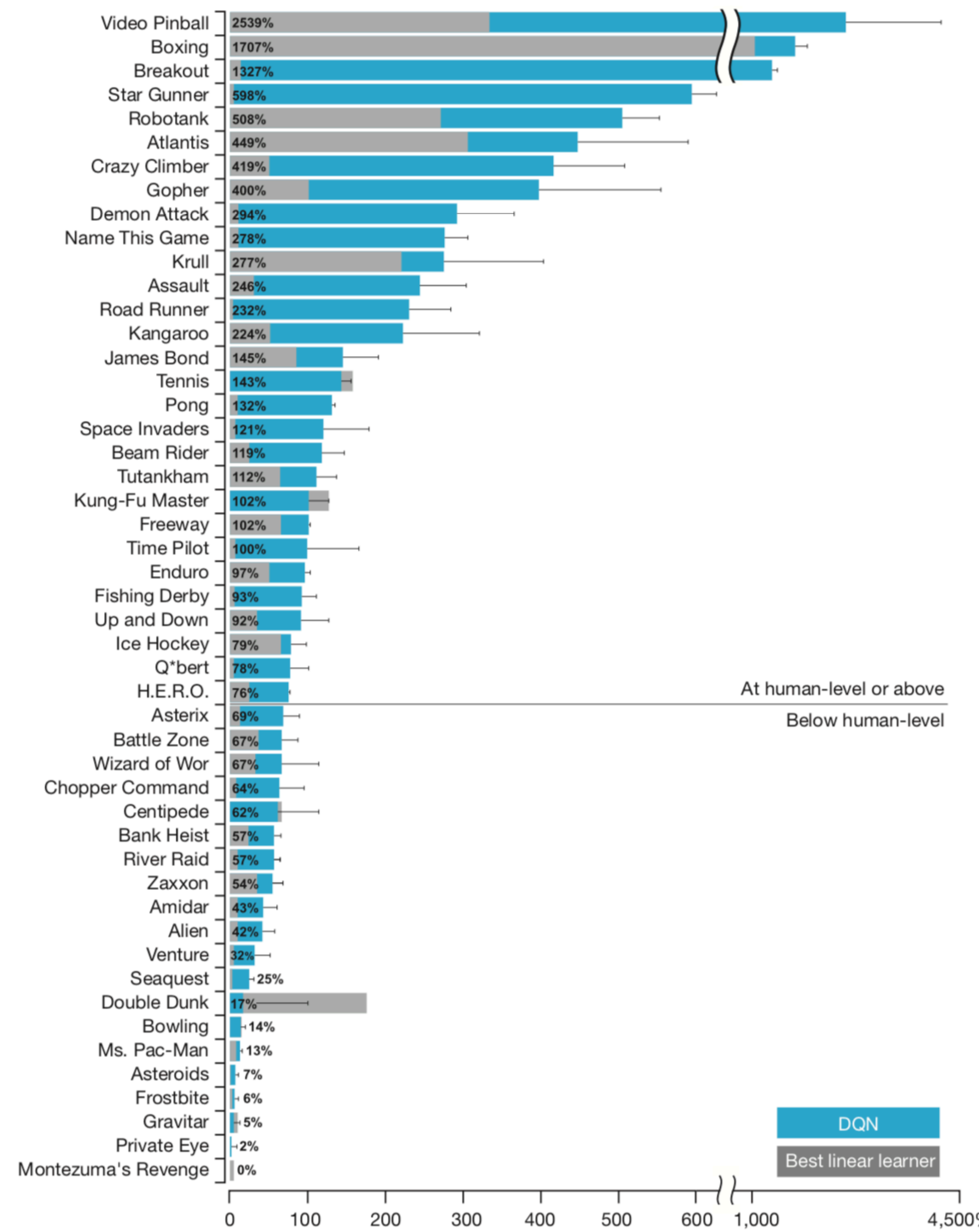
experience: $e_t = \{s_t, a_t, s_{t+1}, r_{t+1}\}$

Objective: Learn $\pi : \mathcal{S} \rightarrow \mathcal{A}$ to maximize cumulative rewards

(Sutton and Barto, 2017)

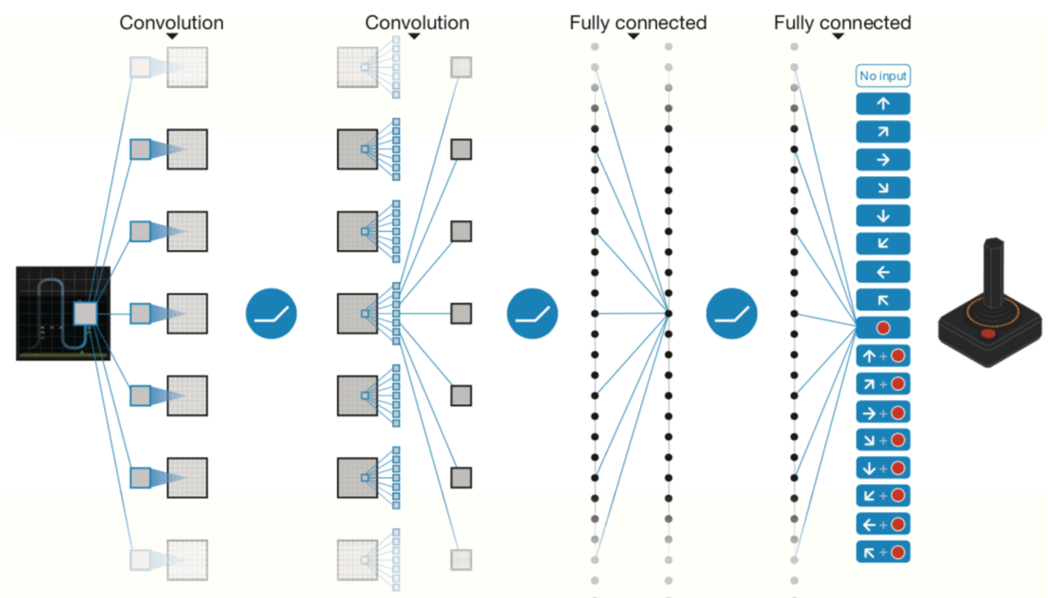
Super-Human Success

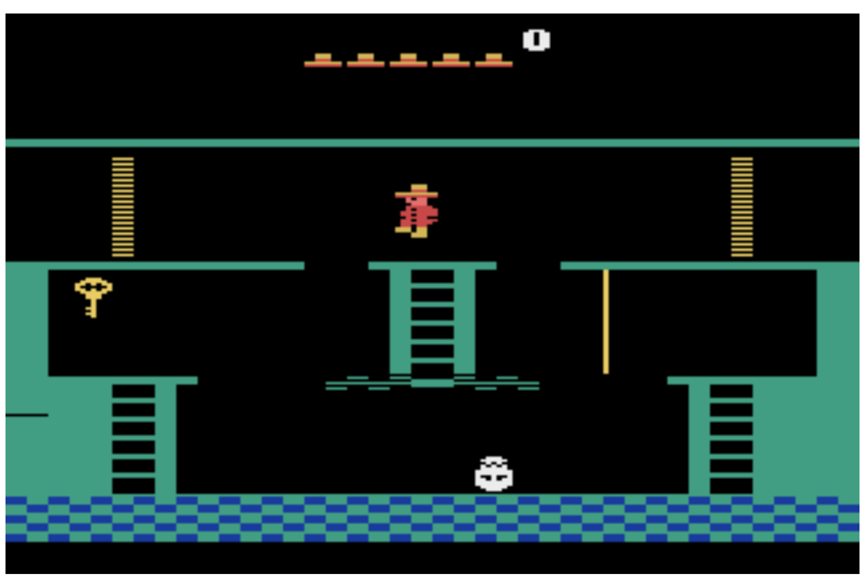


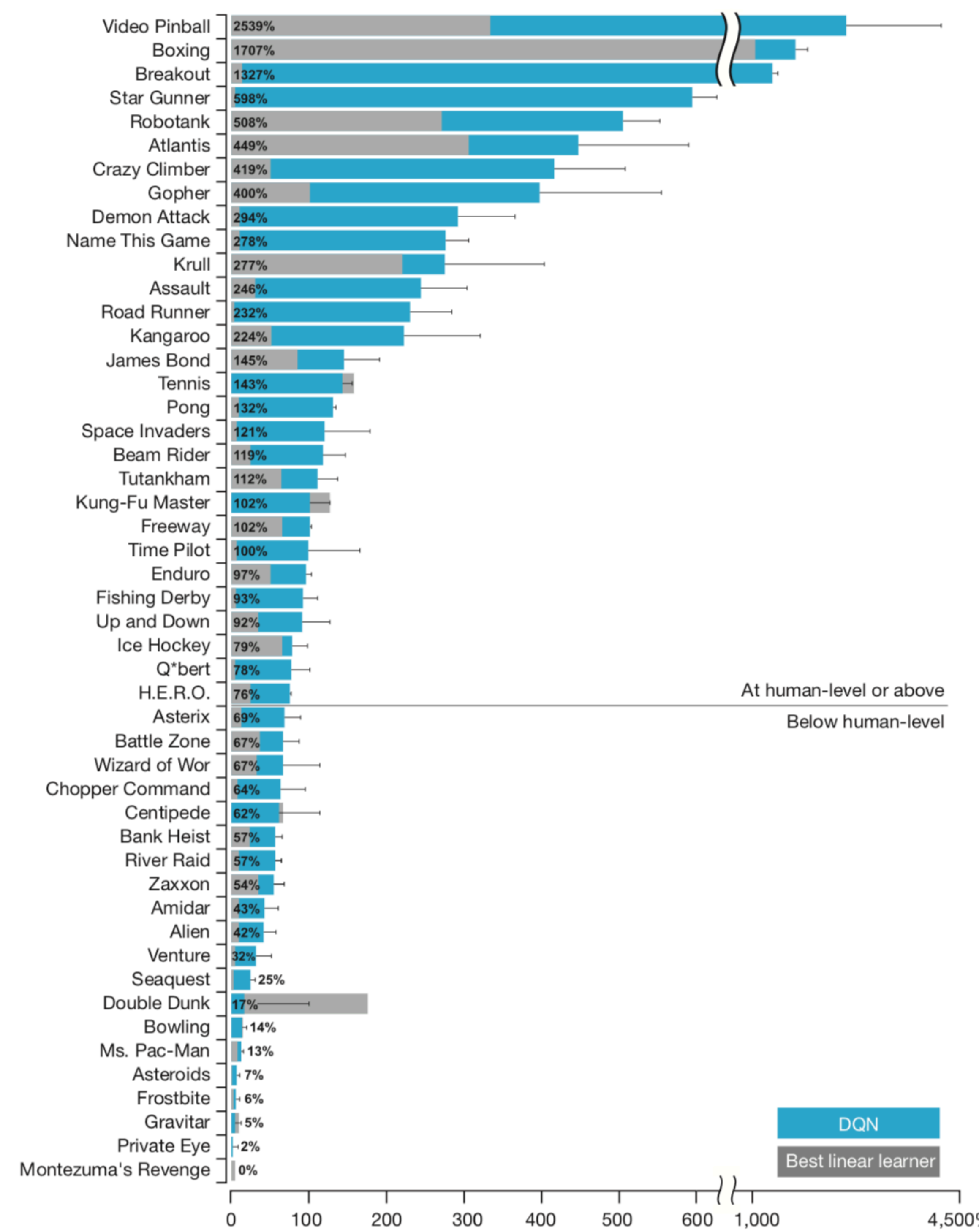


(Mnih. et. al., 2015)

Failure in a complex task







(Mnih. et. al., 2015)

Learning Representations in Hierarchical Reinforcement Learning

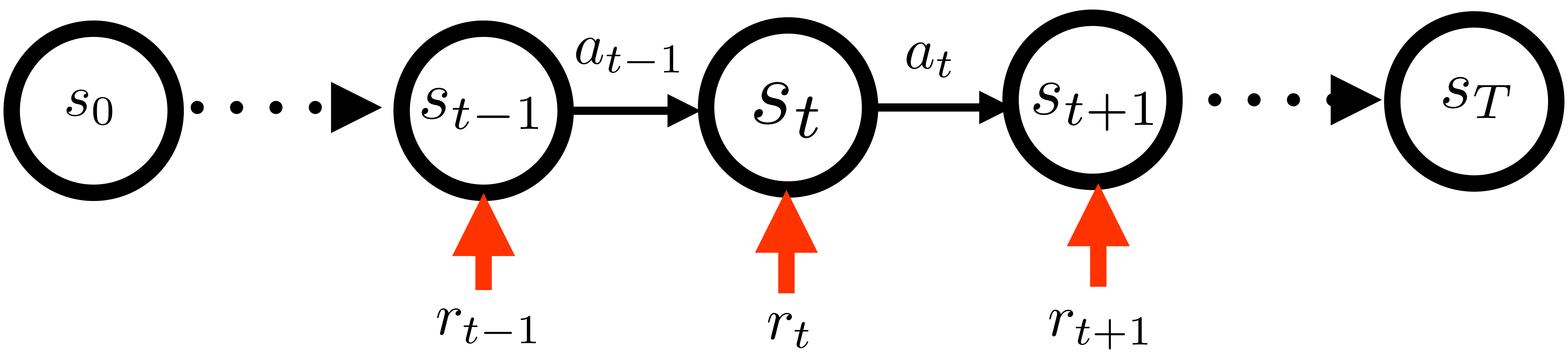
- Trade-off between ***exploration and exploitation*** in an environment with ***sparse feedback*** is a major challenge.
- Learning to operate over different levels of ***temporal abstraction*** is an important open problem in reinforcement learning.
- Exploring the state-space while learning reusable skills through ***intrinsic motivation***.
- Discovering useful ***subgoals*** in large-scale hierarchical reinforcement learning is a major open problem.

Return

Return is the cumulative sum of a received reward:

$$G_t = \sum_{t'=t+1}^{T} \gamma^{t'-t-1} r_{t'}$$

$\gamma \in [0, 1]$ is the discount factor



Policy Function

- Policy Function: At each time step agent implements a mapping from states to possible actions

$$\pi : \mathcal{S} \to \mathcal{A}$$

- **Objective**: Finding an **optimal policy** that maximizes the cumulated rewards

$$\pi^* = \arg\max_{\pi} \mathbb{E}\big[G_t | S_t = s\big], \quad \forall s \in \mathcal{S}$$

Q-Function

- State-Action Value Function is the expected return when starting from *(s,a)* and following *a policy* thereafter

$$Q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$$

$$Q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

Temporal Difference

- **Model-free reinforcement learning algorithm.**
- State-transition probabilities or reward function are not available
- A powerful computational cognitive neuroscience model of learning in brain
- A combination of Monte Carlo method and Dynamic Programming

Q-learning

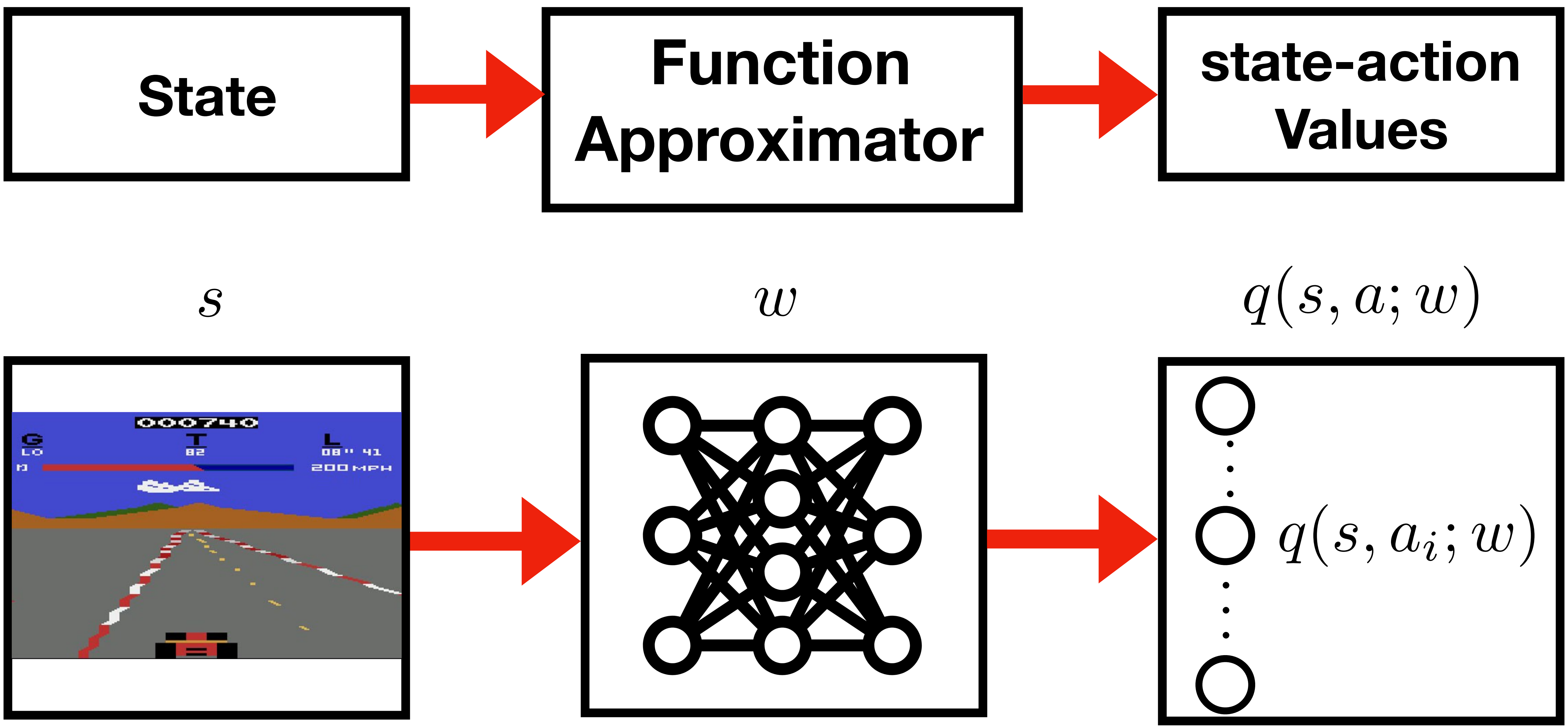
$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

$$Q(s,a) \rightarrow \text{ prediction of return}$$

$$r + \gamma \max_{a'} Q(s',a') \;\rightarrow\; \text{target value}$$

Generalization

$$Q(s,a) \approx q(s,a;w)$$



Deep RL

$$\min_w L(w)$$

$$w = \arg\min_w L(w)$$

$$L(w) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(r + \max_{a'} q(s', a'; w^-) - q(s, a; w)\right)^2\right]$$

$$\mathcal{D} = \{e_t | t = 0, \ldots, T\} \rightarrow \text{ Experience replay memory}$$

Stochastic Gradient Decent method

$$w \leftarrow w - \nabla_w L(w)$$

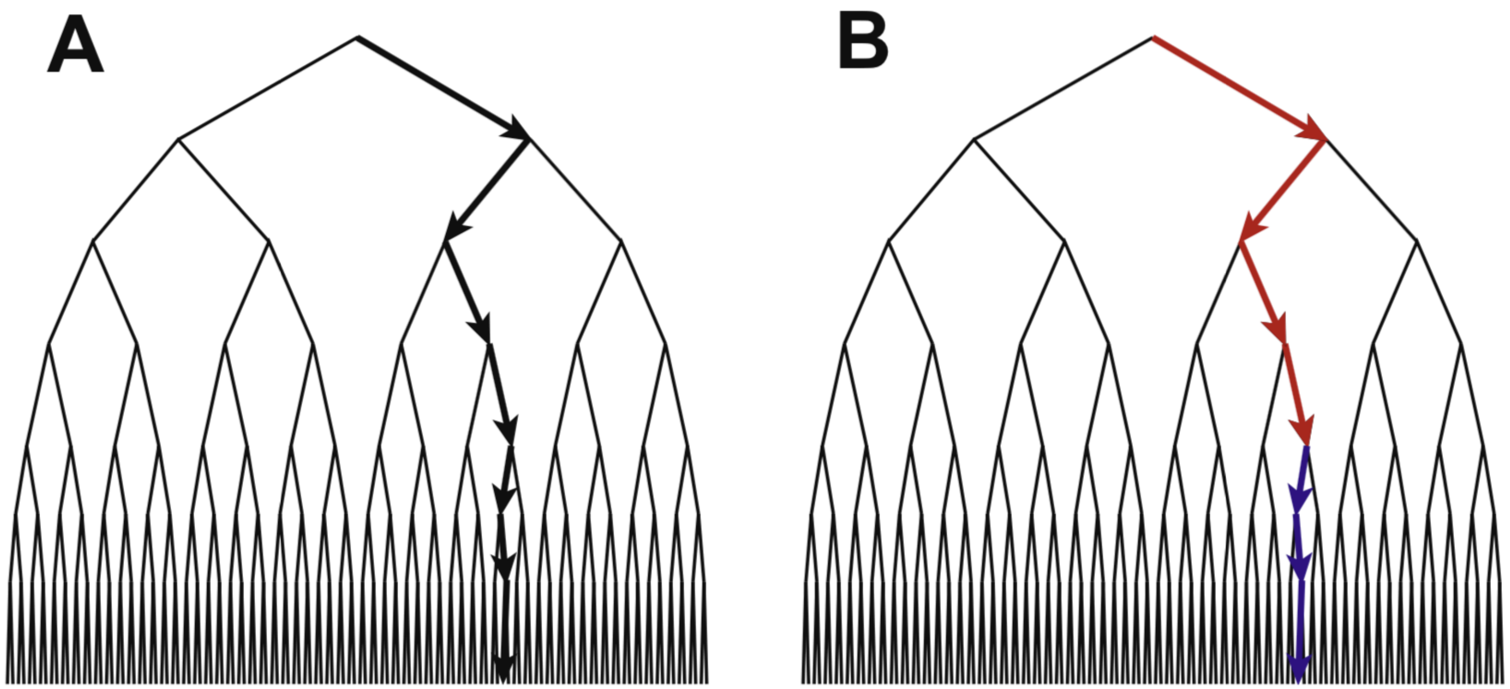
Q-Learning with experience replay memory

Algorithm Q-Learning with Experience Replay

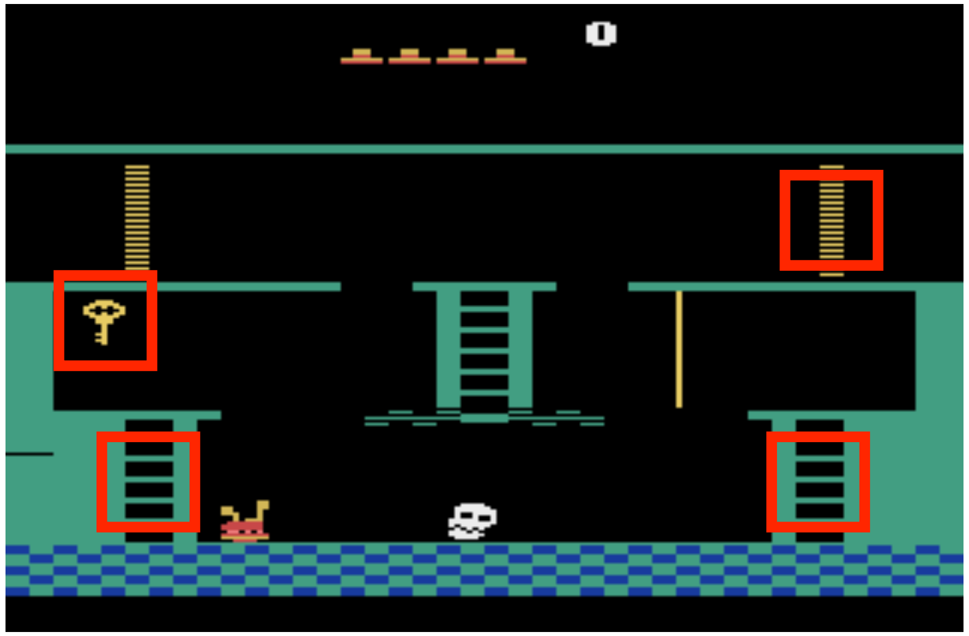
Initialize: replay memory $\mathcal{D}$
Initialize: weights of action-value function $q(s, a; w)$ arbitrarily

repeat (for each episode)
  initialize s
  repeat (for each step of episode $t = 1, \dots, T$)
    choose action a using policy derived by $q(s, a; w)$ (e.g. ϵ-greedy)
    take action a, observe reward r and next state s'
    store experience $e = (s, a, r, s')$ to experience memory $\mathcal{D}$
    sample random mini-batch from experience replay memory $\mathcal{D}$
    compute $\nabla_w L(w)$
    update weights (e.g. SGD step) $w \leftarrow w - \alpha \nabla_w L(w)$
    $s \leftarrow s'$, $a \leftarrow a'$
  until (s is terminal)
until (convergence or reaching to max number of episodes)

Failure: Sparse feedback

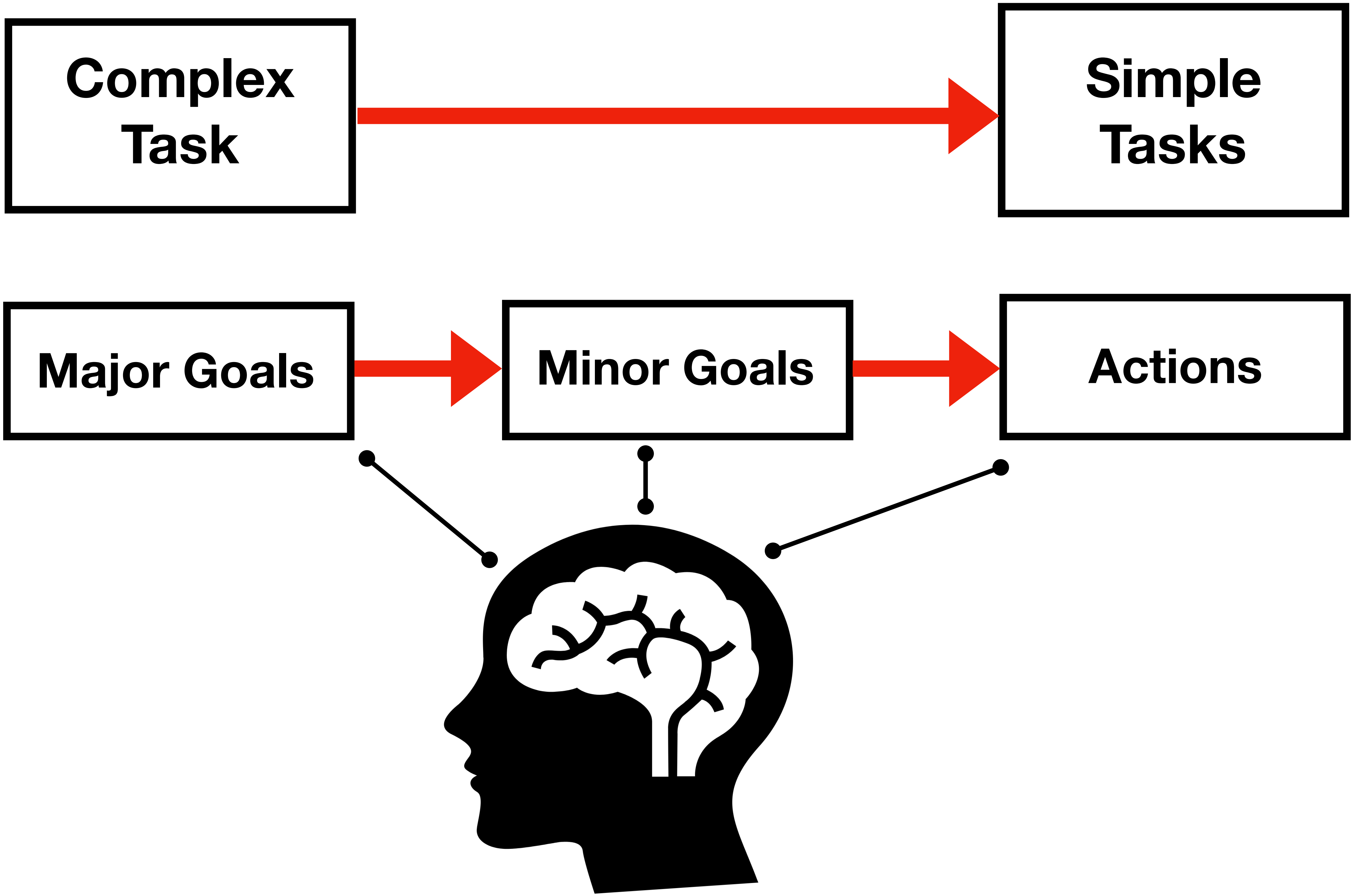


(Botvinick et al., 2009)



Subgoals

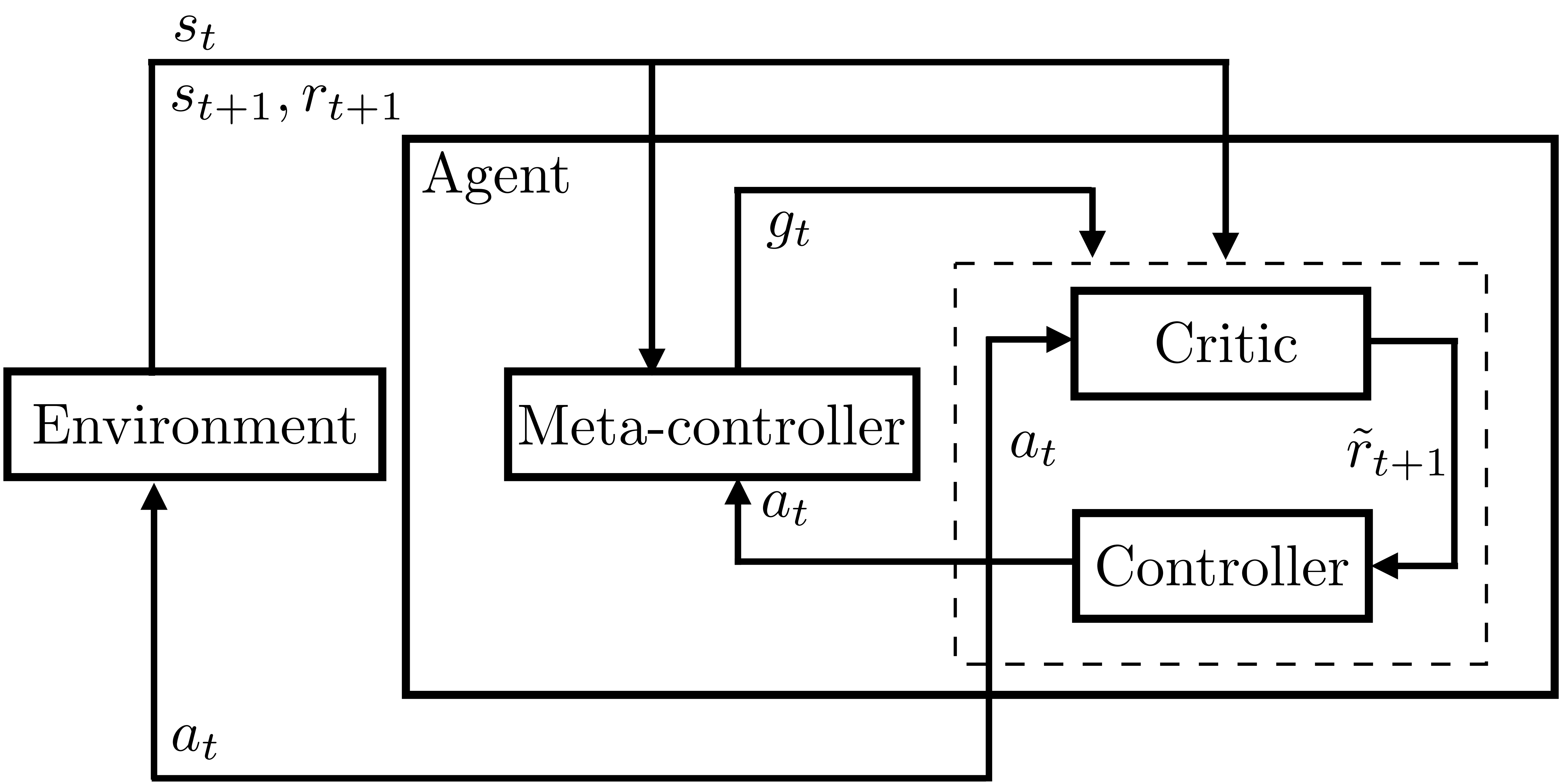
Hierarchy in Human Behavior & Brain Structure



Hierarchical Reinforcement Learning Subproblems

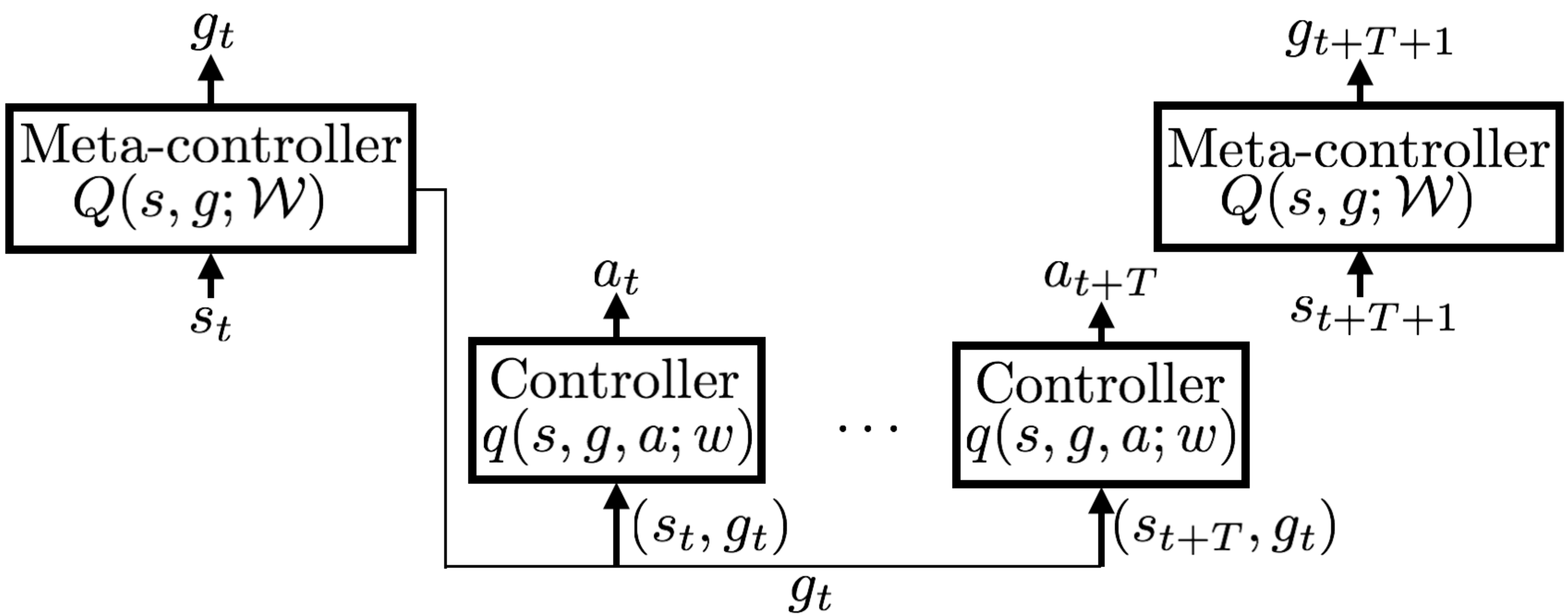
- **Subproblem 1:** Learning a meta-policy to choose a subgoal
- **Subproblem 2:** Developing skills through intrinsic motivation
- **Subproblem 3:** Subgoal discovery

Meta-controller/Controller Framework

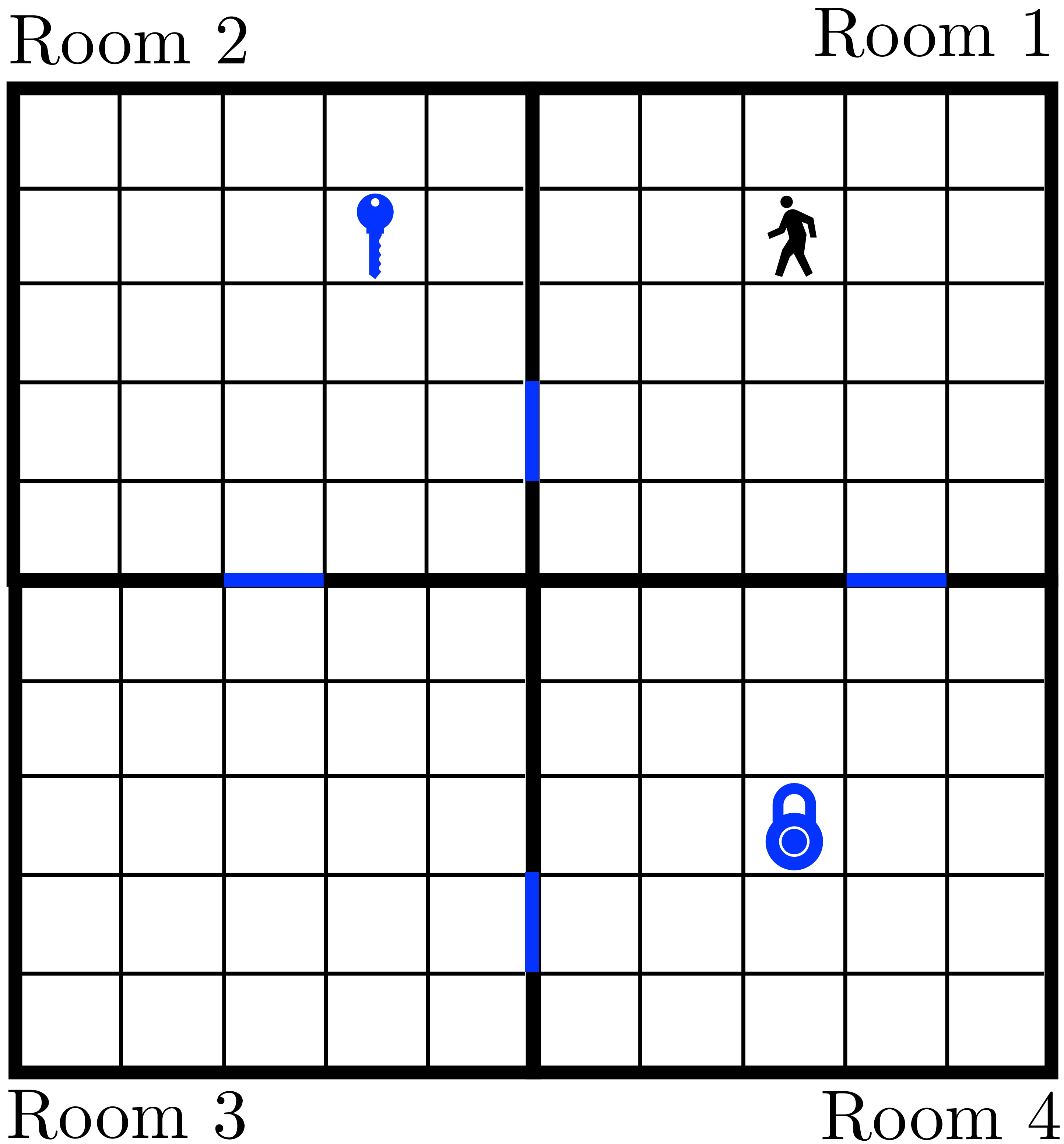


Kulkarni et. al. 2016

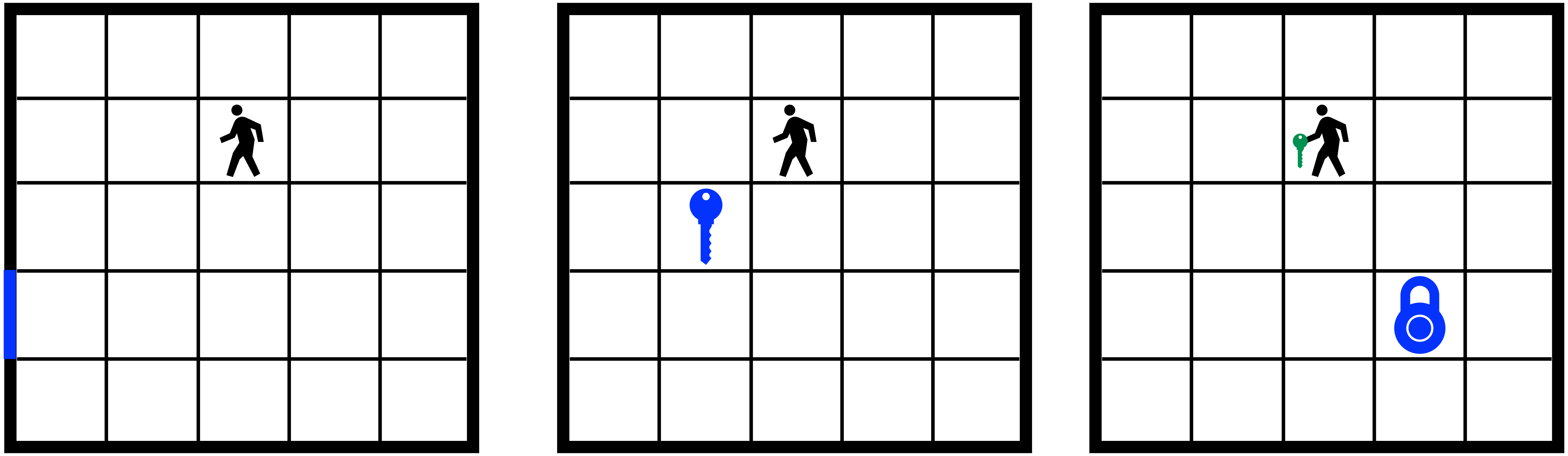
Subproblem 1: Temporal Abstraction



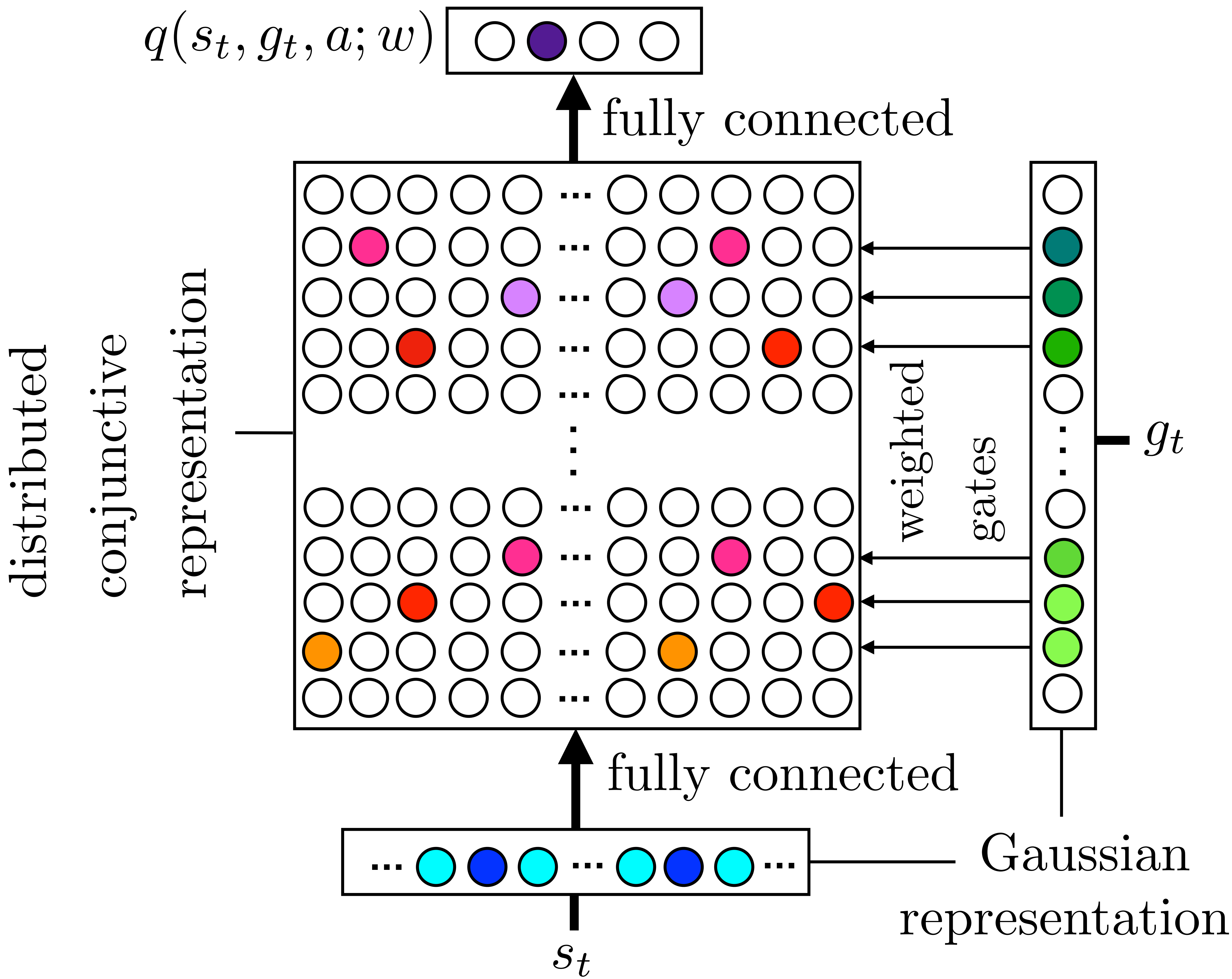
Rooms Task



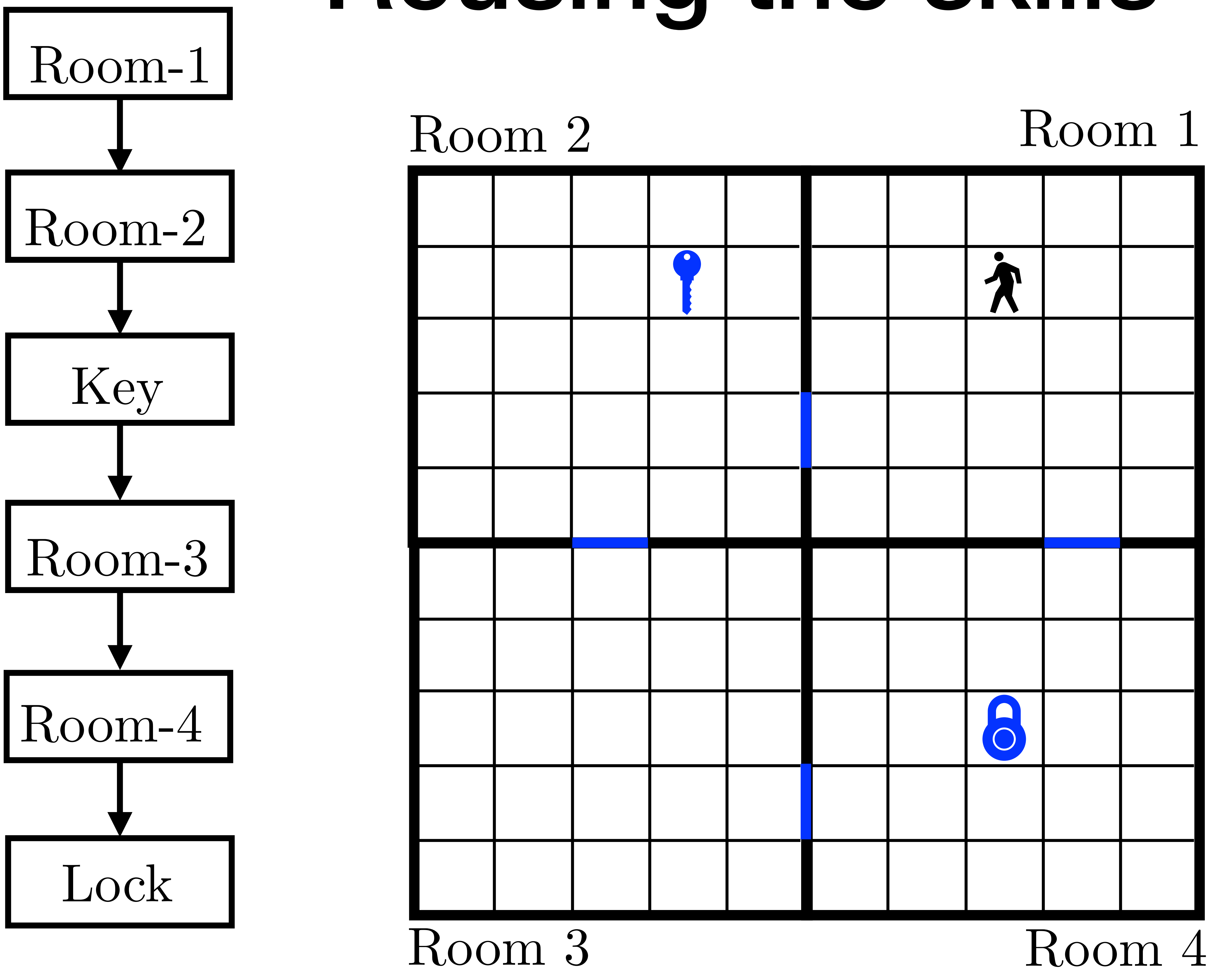
Subproblem 2.
Developing skills through Intrinsic Motivation



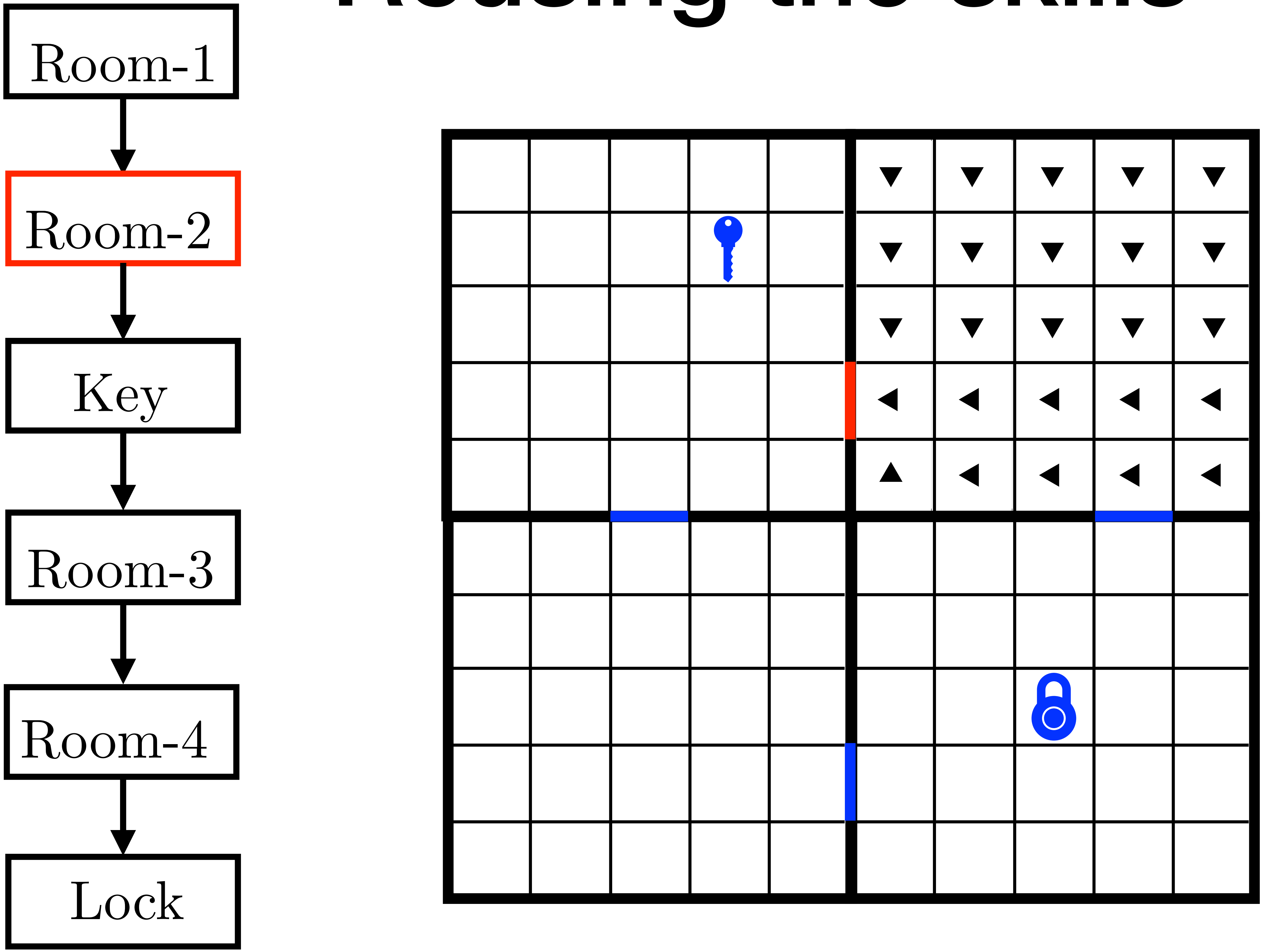
State-Goal Q Function



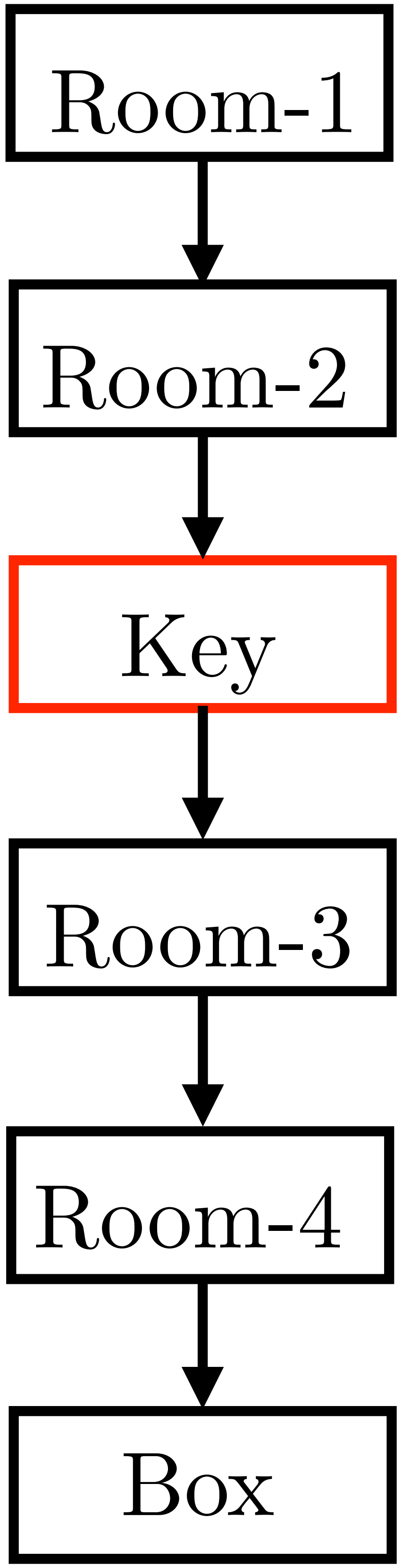
Reusing the skills

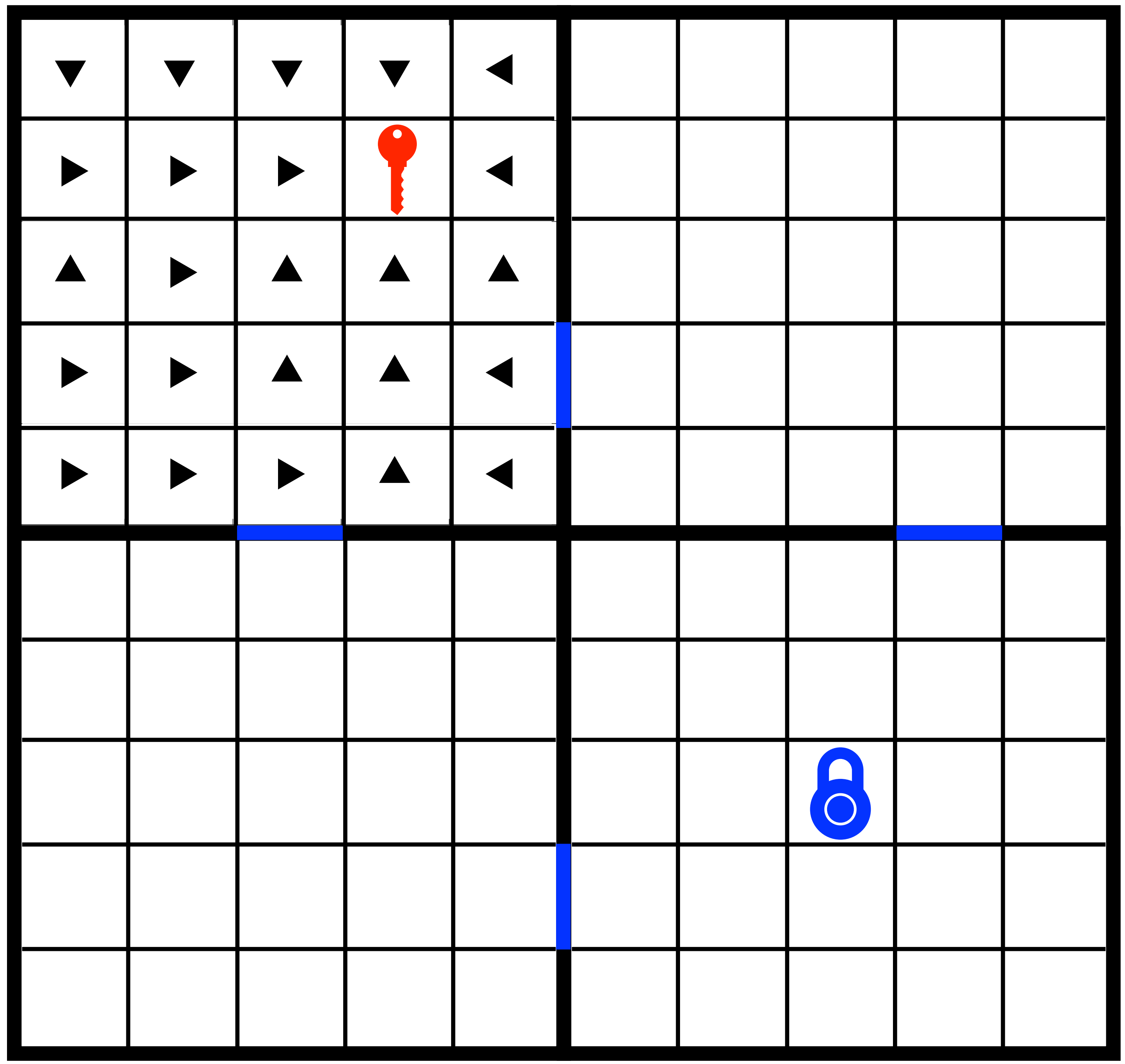


Reusing the skills

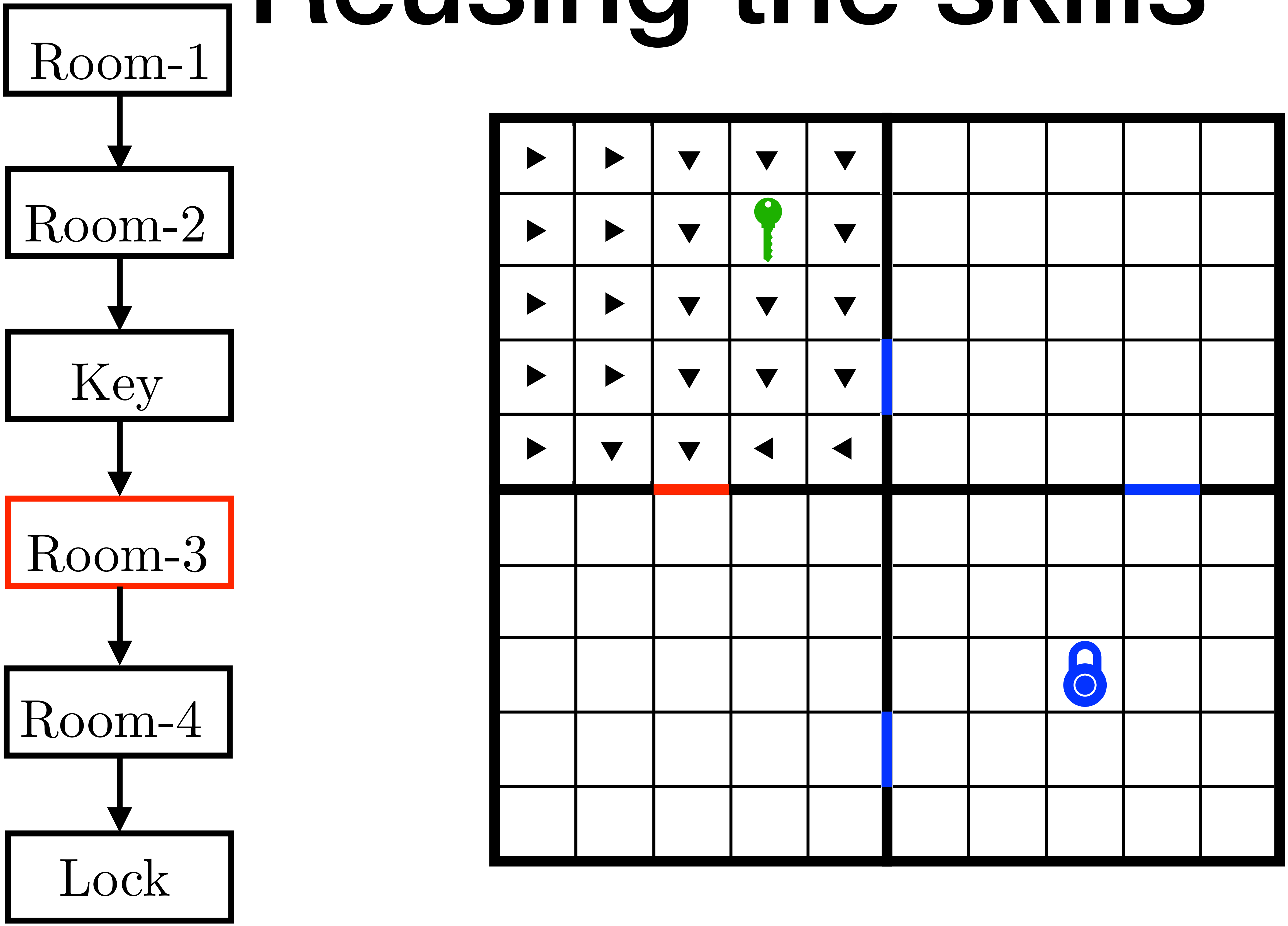


Reusing the skills



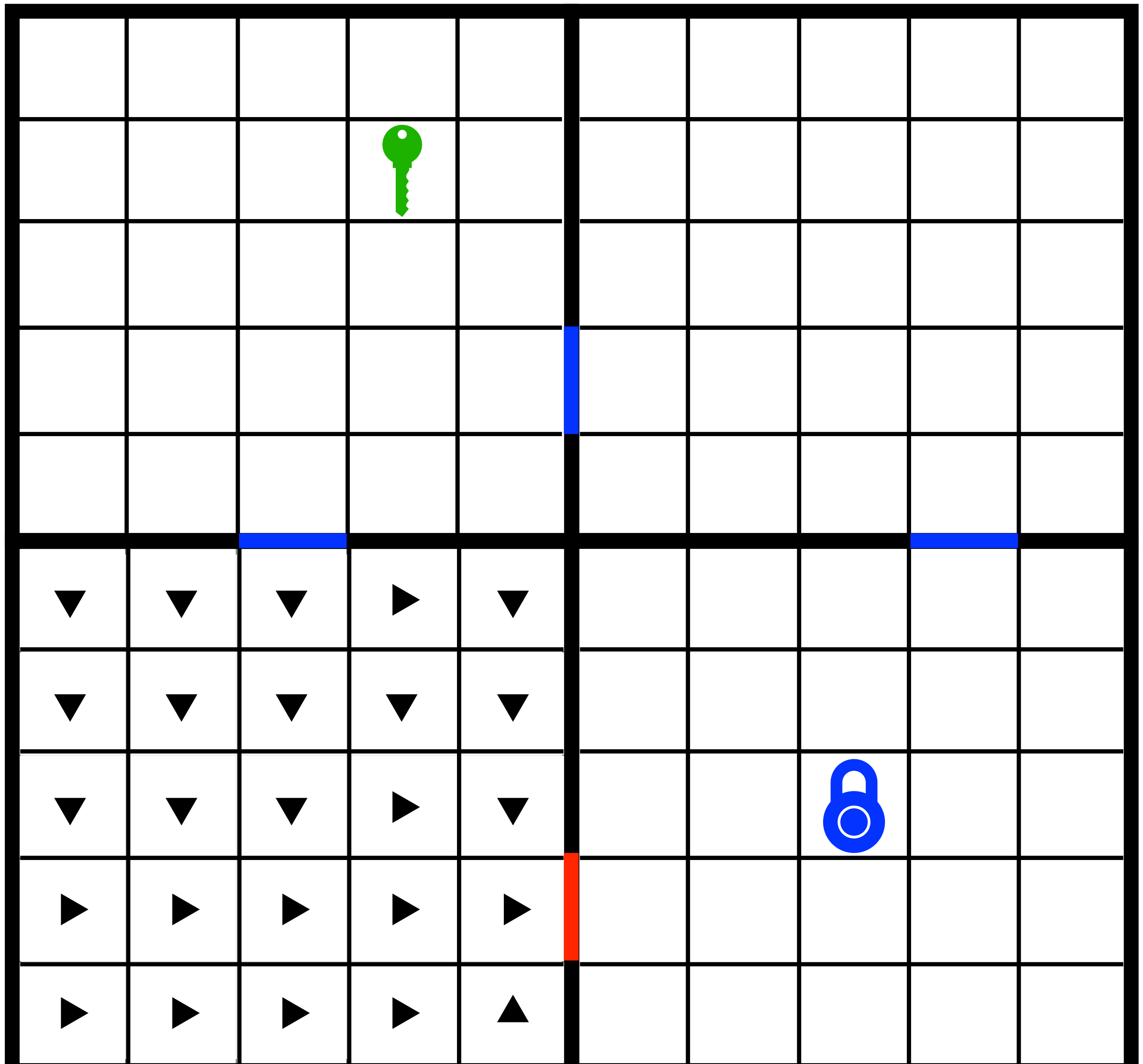


Reusing the skills

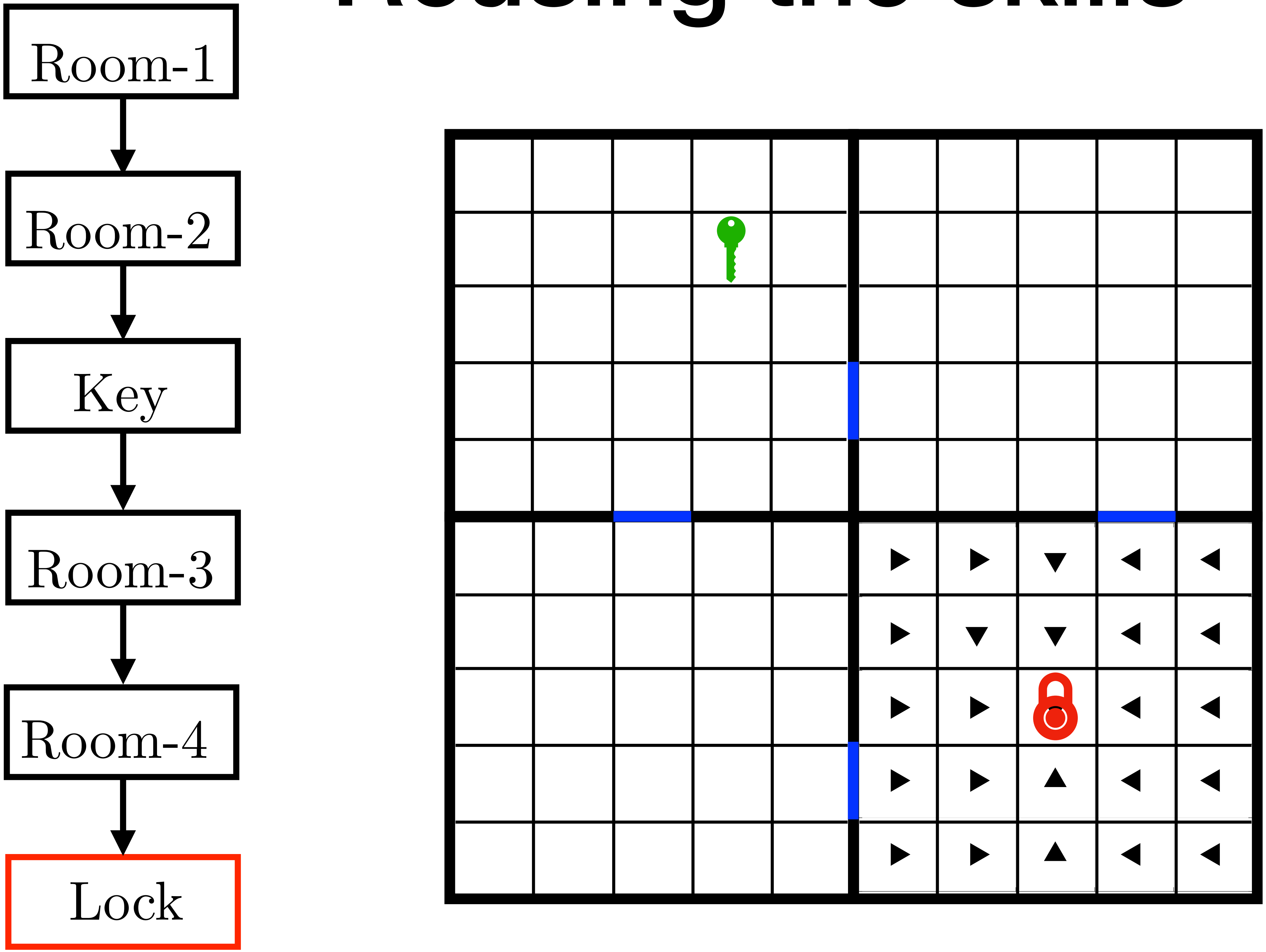


Reusing the skills

Room-1 → Room-2 → Key → Room-3 → Room-4 → Lock



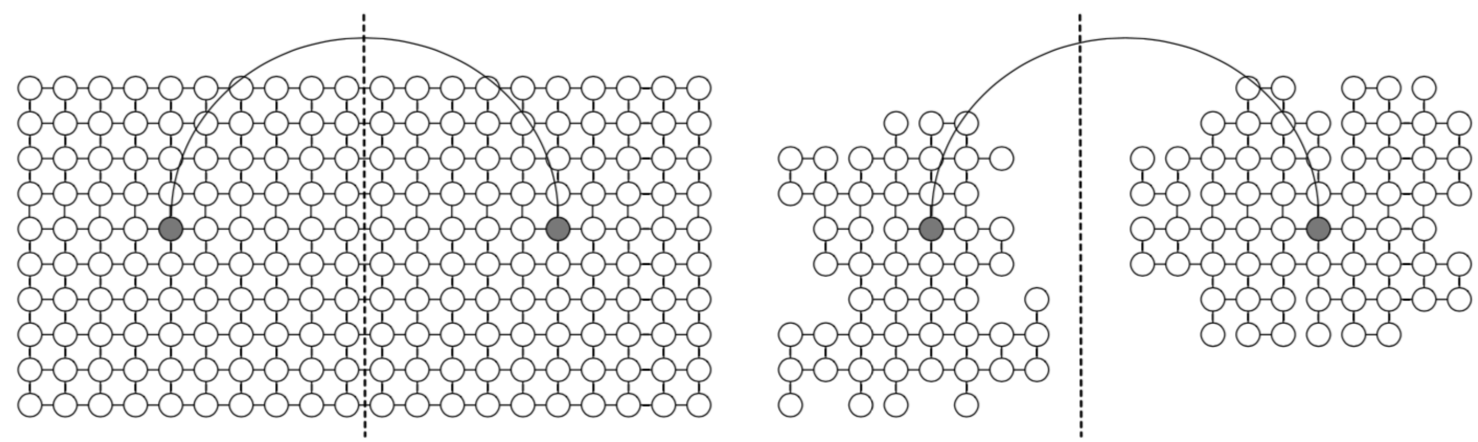
Reusing the skills



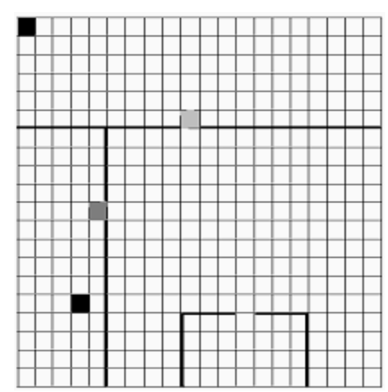
Subproblem 3.
Subgoal Discovery

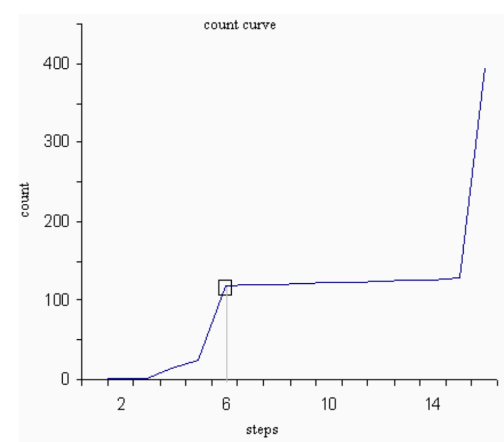
finding proper $\mathcal{G}$

(Sismek et al., 2005)

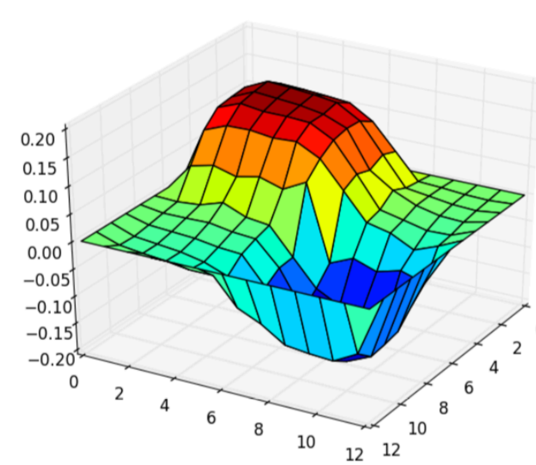


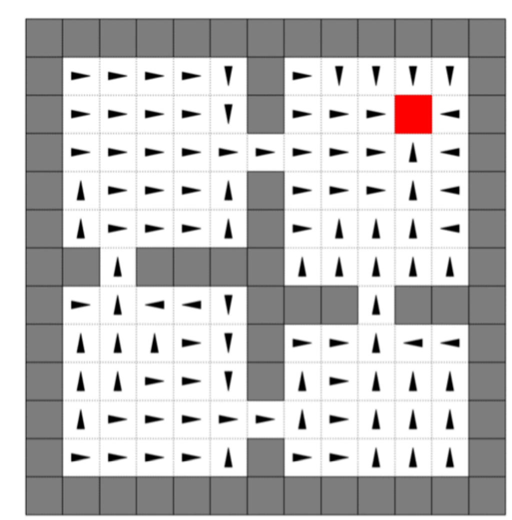
(Goel and Huber, 2003)





(Machado, et. al. 2017)





Subproblem 3.
Subgoal Discovery

- Purpose: Discovering promising states to pursue, i.e. finding $\mathcal{G}$
- Implementing subgoal discovery algorithm for large-scale model free reinforcement learning problem
- No access to MDP models (state-transition probabilities, environment reward function, State space)

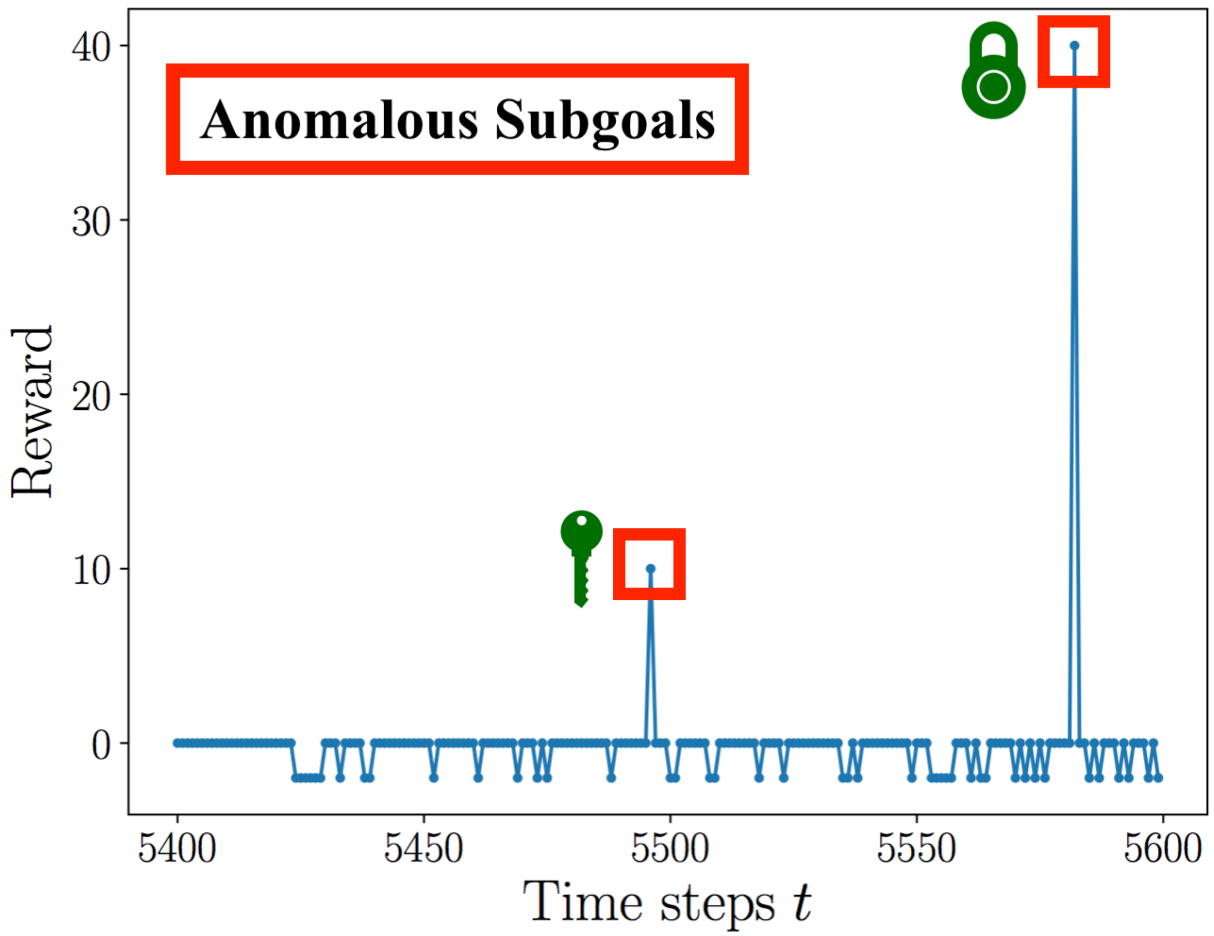
Subproblem 3. Candidate Subgoals

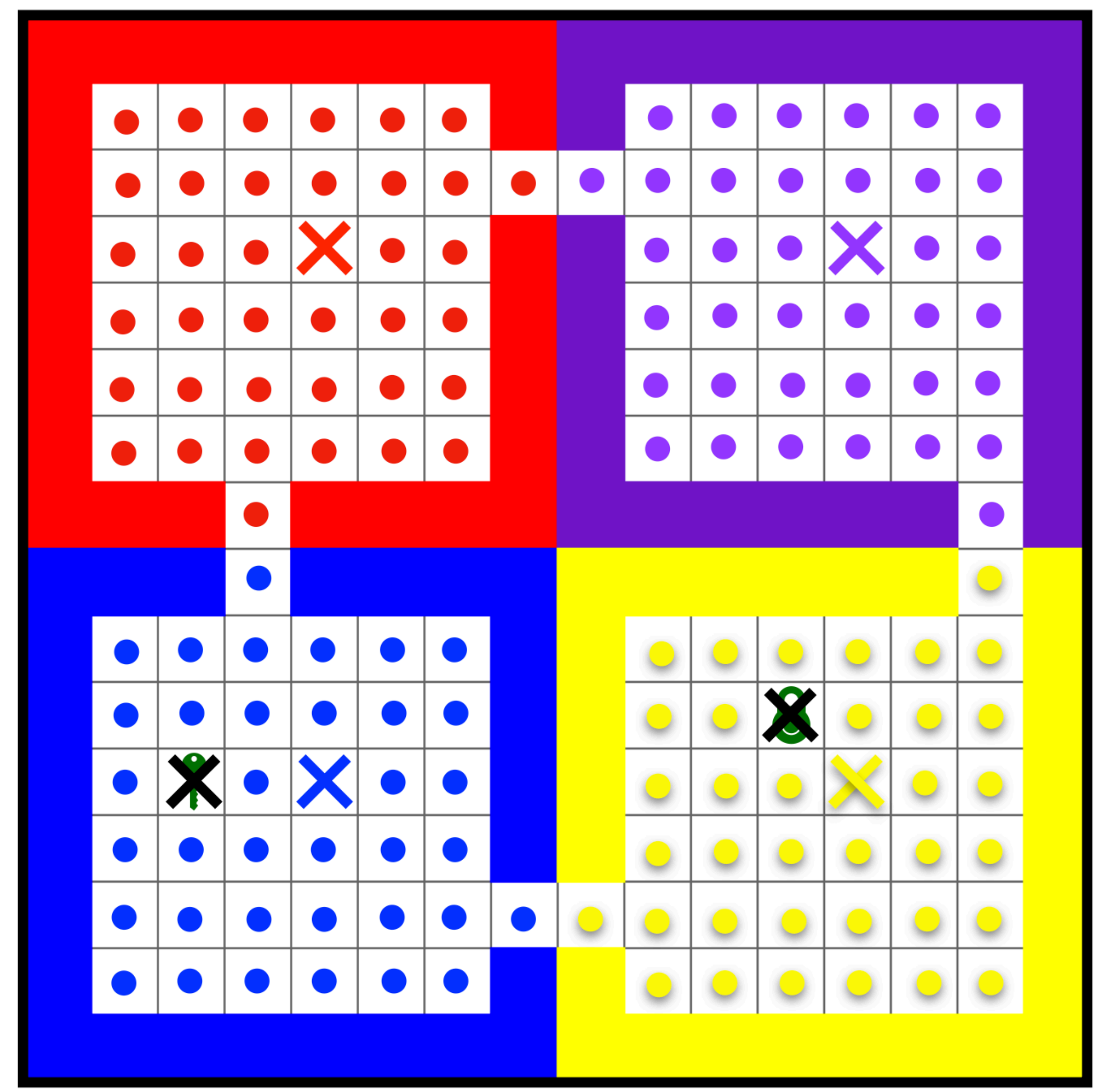
- It is close (in terms of actions) to a rewarding state.
- It represents a set of states, at least some of which tend to be along a state transition path to a rewarding state.

Subproblem 3.
Subgoal Discovery

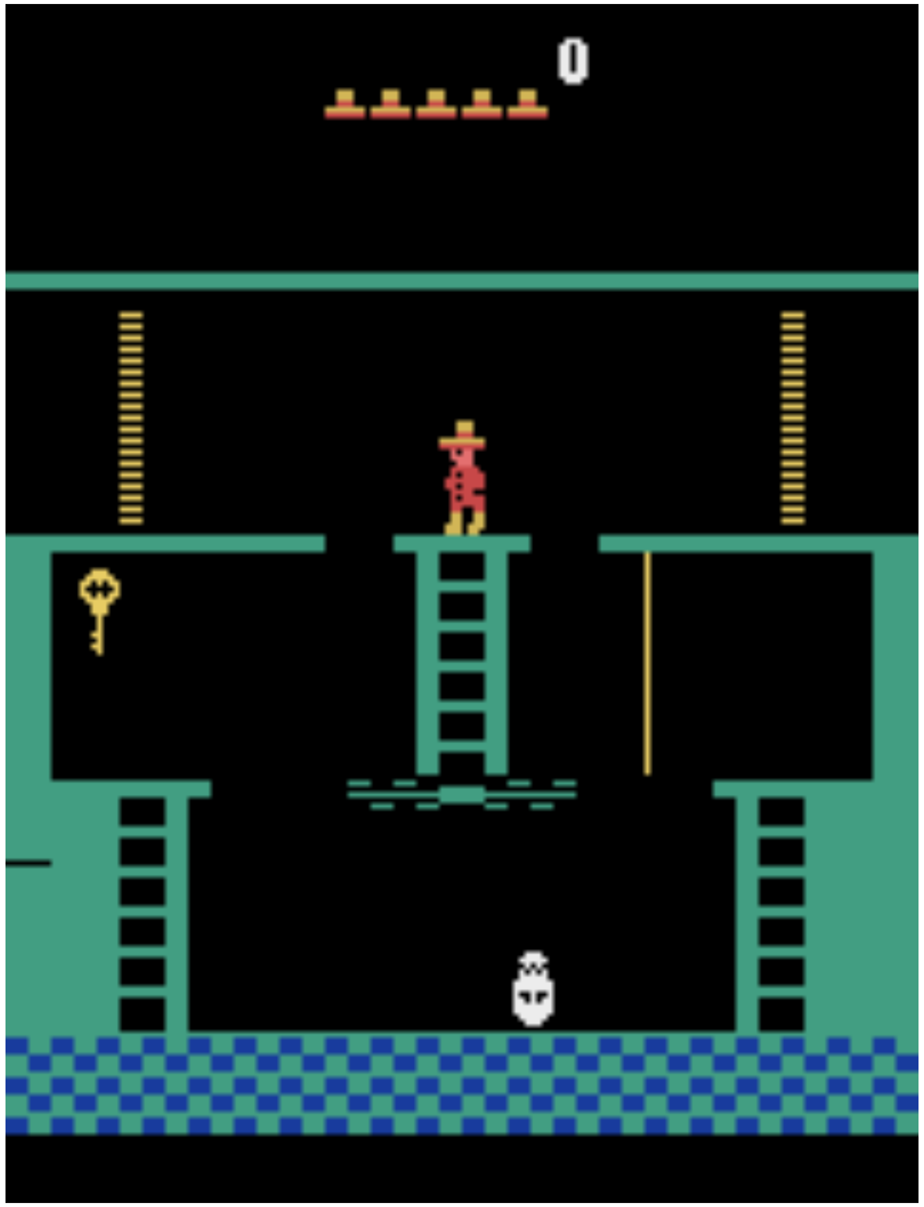
- Unsupervised learning (clustering) on the limited past experience memory collected during intrinsic motivation
- Centroids of clusters are useful subgoals (e.g. rooms)
- Detecting outliers as potential subgoals (e.g. key, box)
- Boundary of two clusters can lead to subgoals (e.g. doorway between rooms)

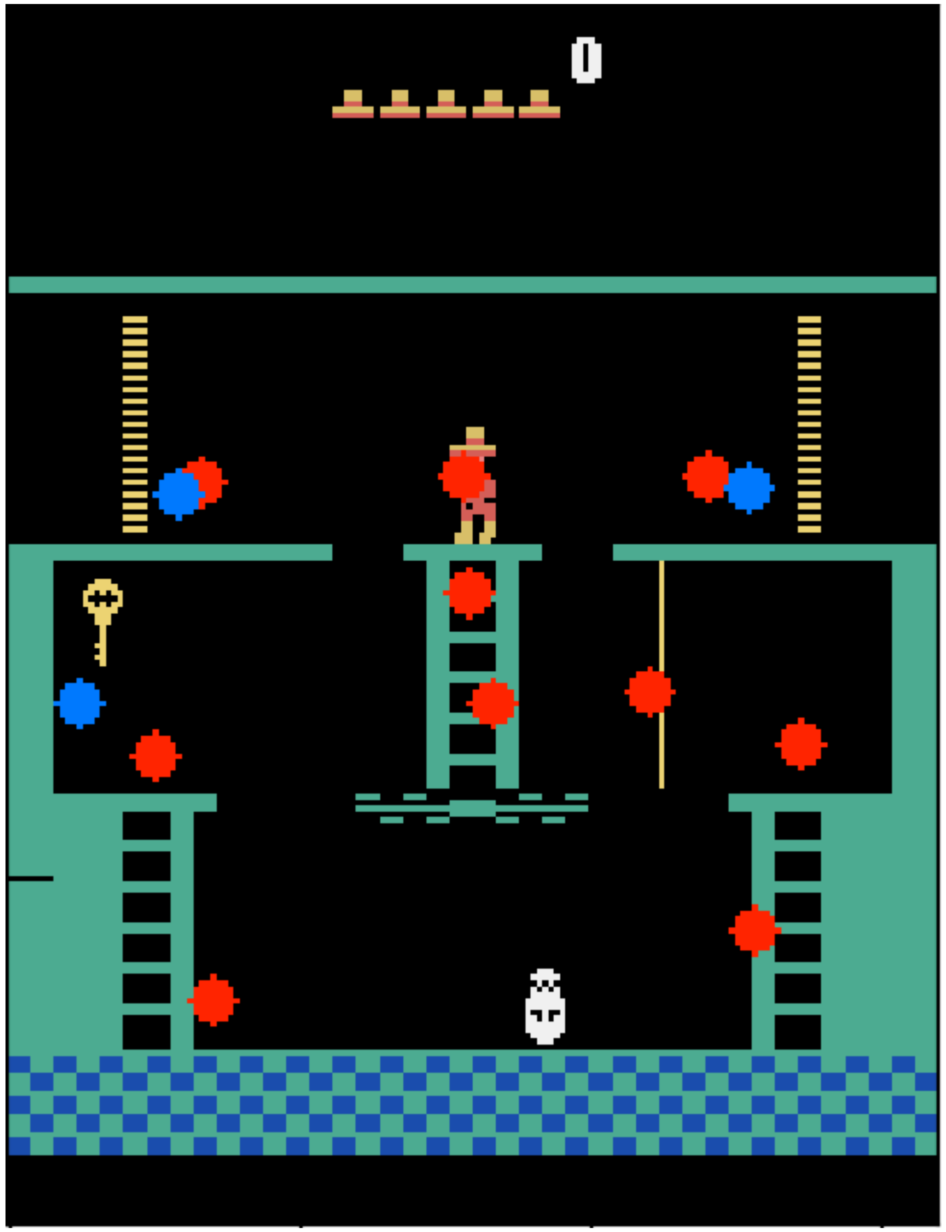
Unsupervised Subgoal Discovery





Unsupervised Subgoal Discovery

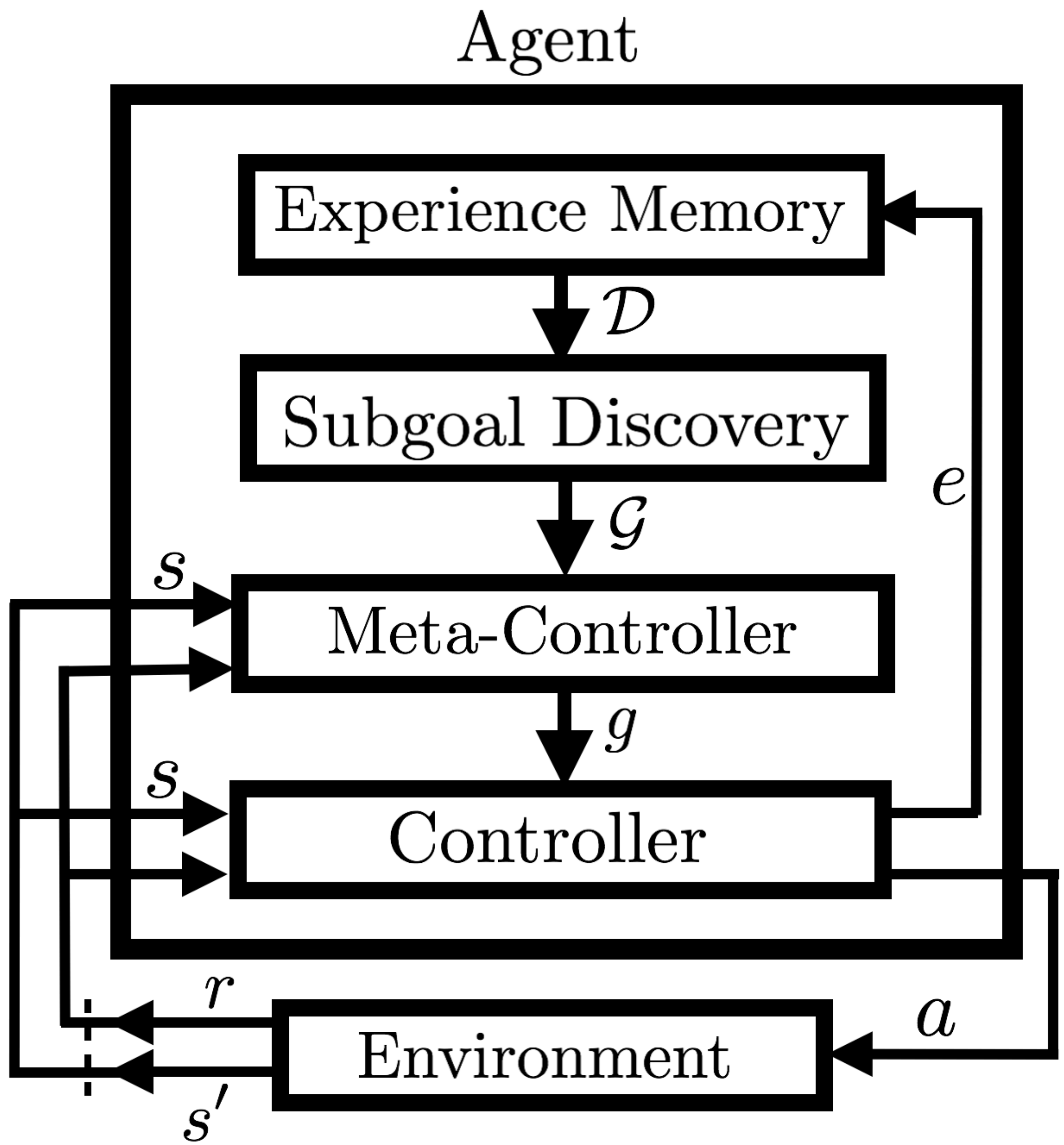




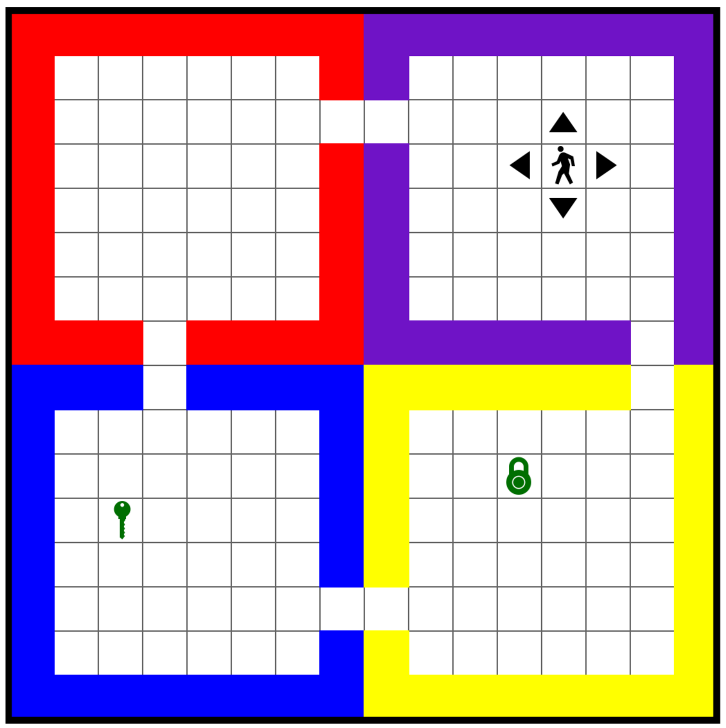
Unification of Hierarchical Reinforcement Learning Subproblems

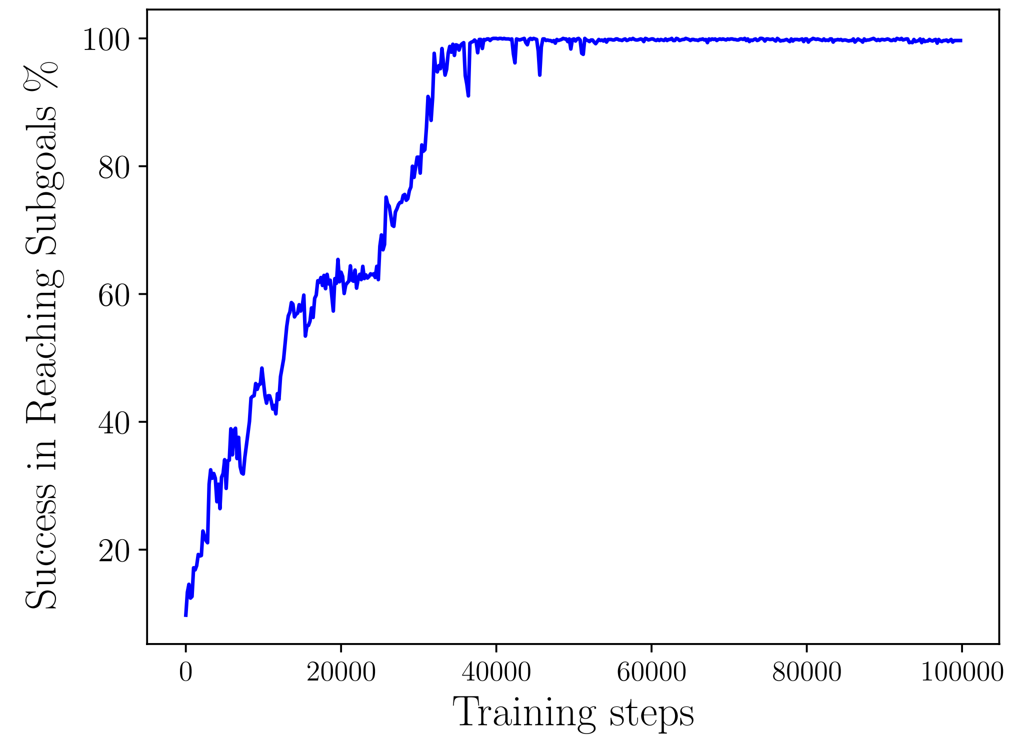
- Implementing a hierarchical reinforcement learning framework that makes it possible to simultaneously perform subgoal discovery, learn appropriate intrinsic motivation, and succeed at meta-policy learning
- The unification element is using experience replay memory $\mathcal{D}$

Model-Free HRL

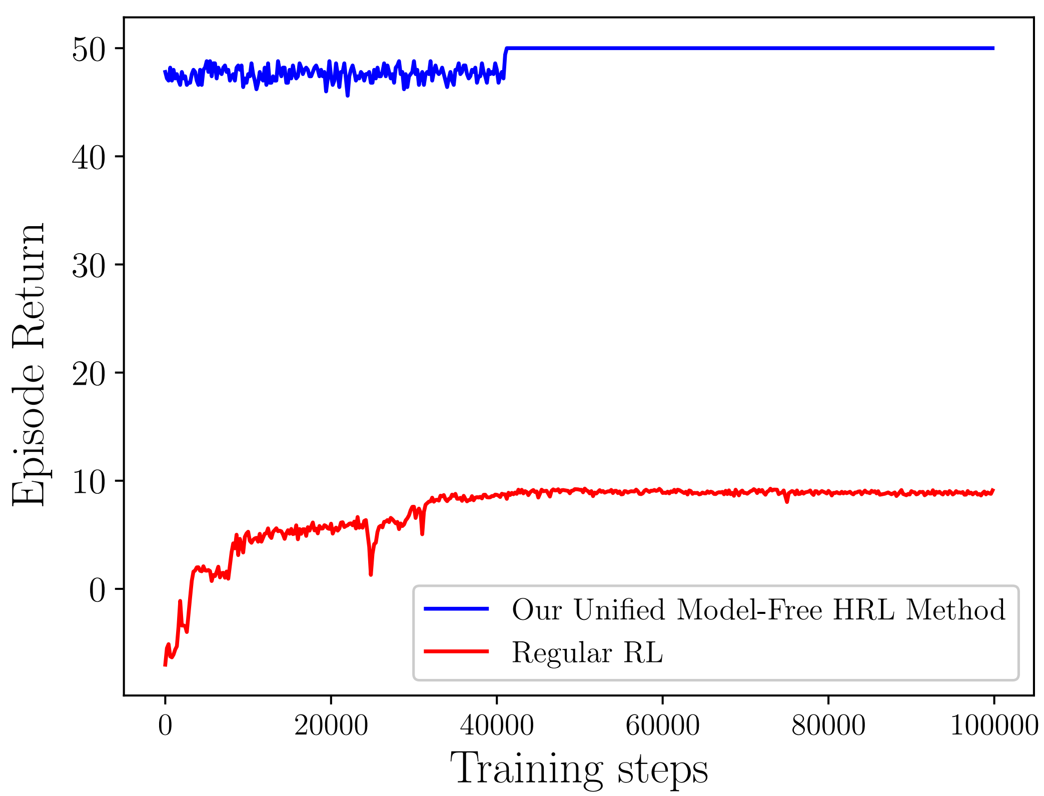


Rooms

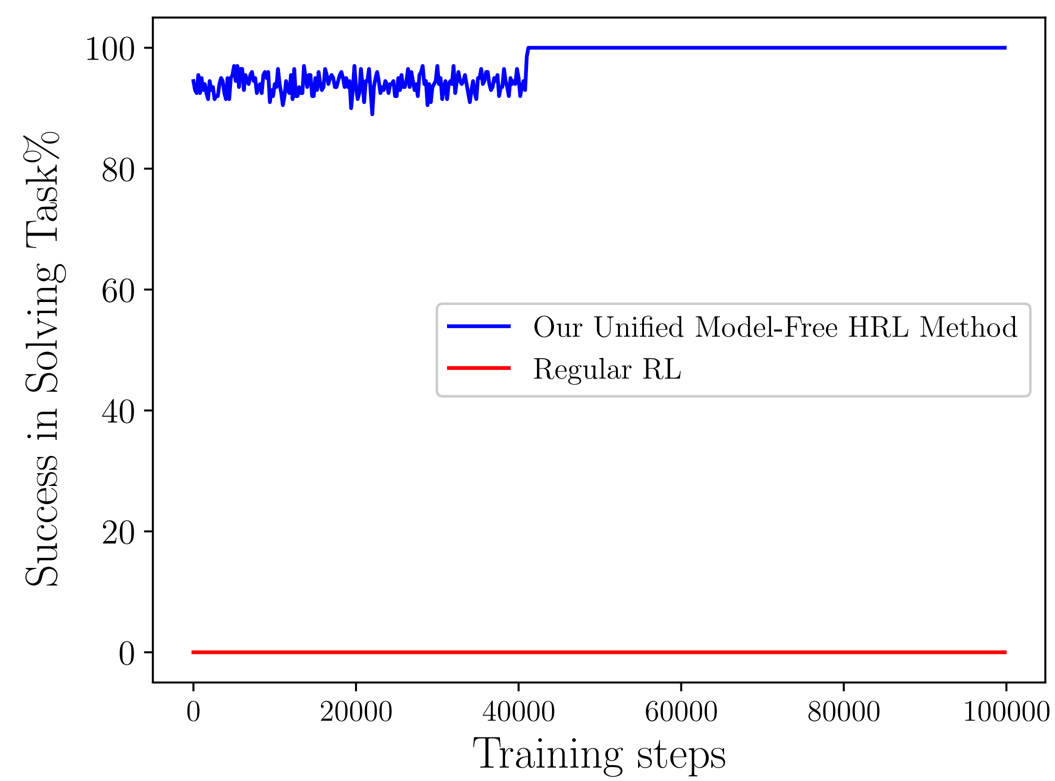




Success in Reaching Subgoals %
100
80
60
40
20
0
20000
40000
60000
80000
100000
Training steps

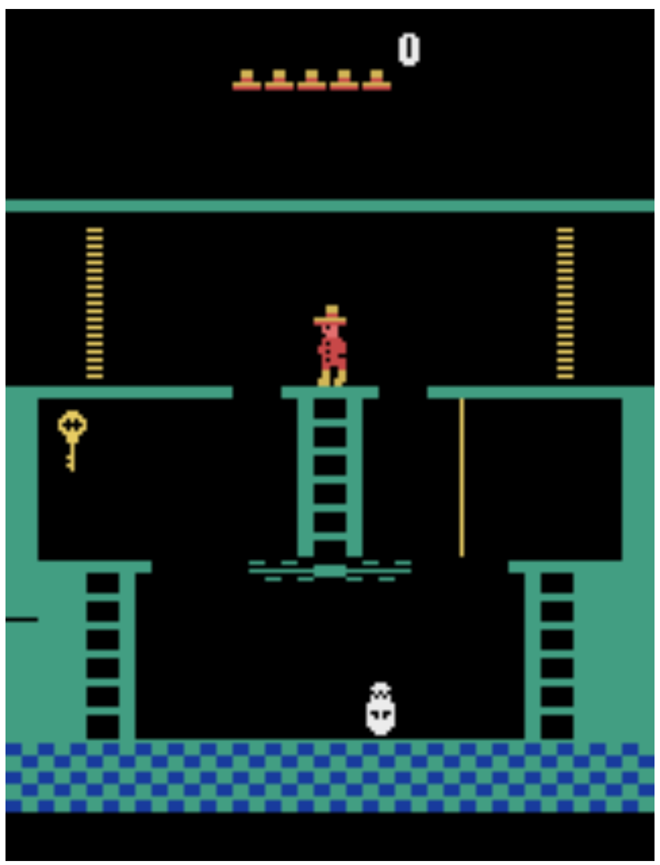


Episode Return
50
40
30
20
10
0
0
20000
40000
60000
80000
100000
Training steps
Our Unified Model-Free HRL Method
Regular RL

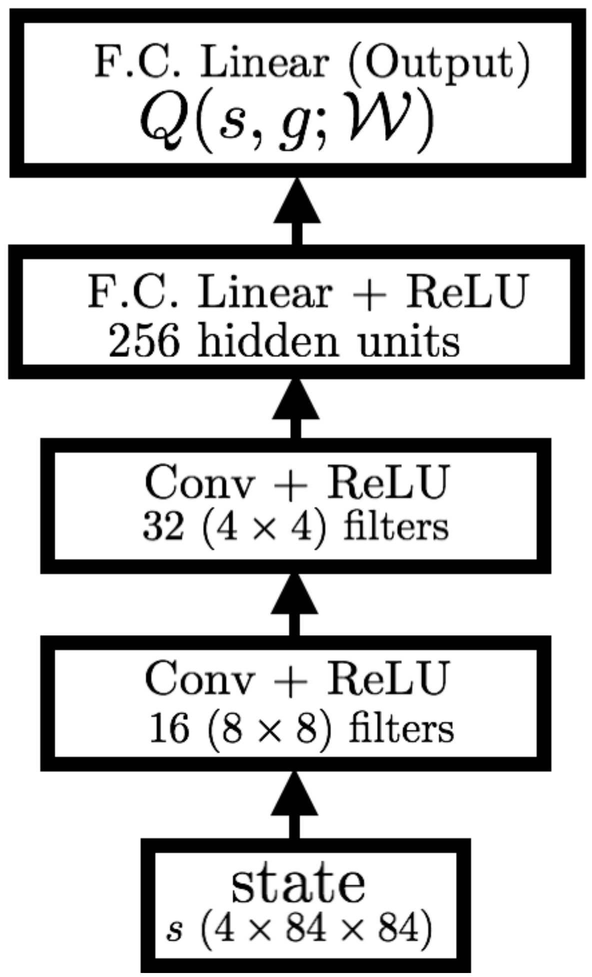


Success in Solving Task%
100
80
60
40
20
0
0
20000
40000
60000
80000
100000
Training steps
Our Unified Model-Free HRL Method
Regular RL

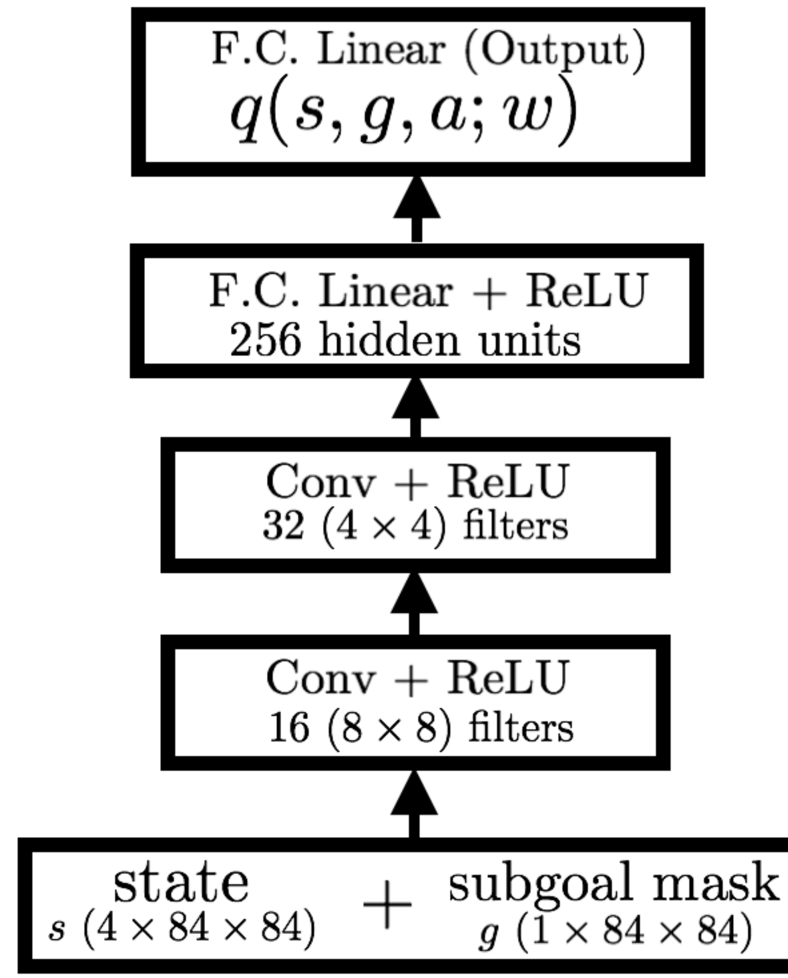
Montezuma's Revenge



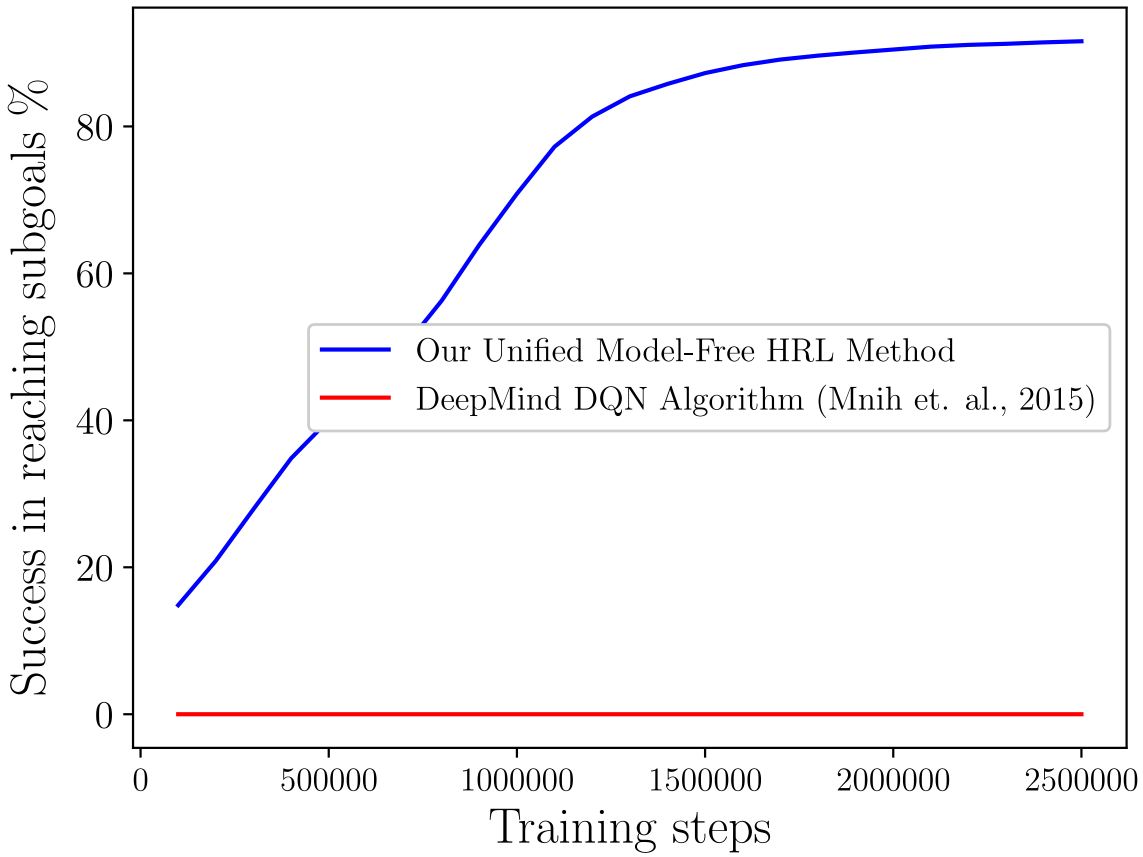
Meta-Controller

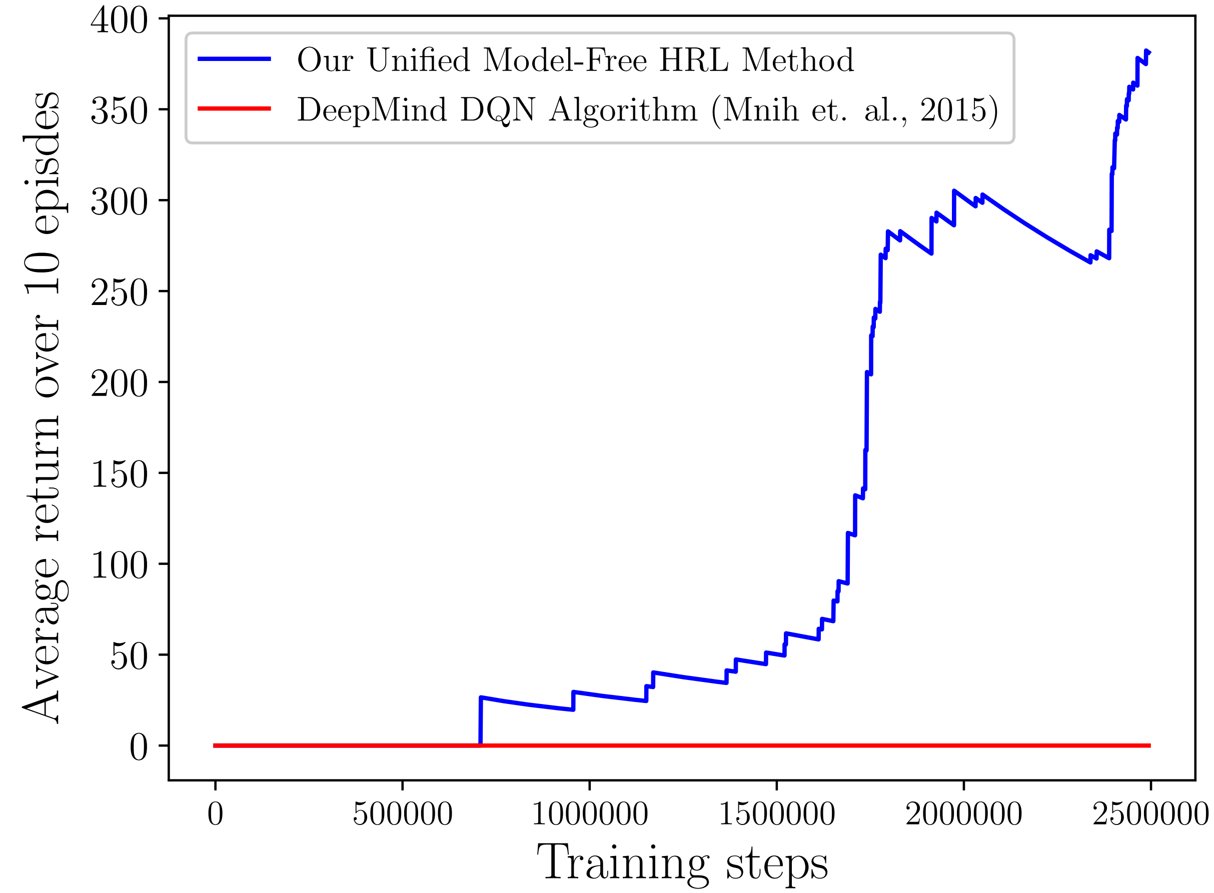


Controller



Montezuma's Revenge





Conclusions

- Unsupervised Learning can be used to discover useful subgoals in games.
- Subgoals can be discovered using model-free methods.
- Learning in multiple levels of temporal abstraction is the key to solve games with sparse delayed feedback.
- Intrinsic motivation learning and subgoal discovery can be unified in model-free HRL framework.

References

- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540):529–533.
- Sutton, R. S., and Barto, A. G. (2017). Reinforcement Learning: An Introduction. MIT Press. 2nd edition.
- Botvinick, M. M., Niv, Y., and Barto, A. C. (2009). Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. Cognition, 113(3):262 – 280.
- Goel, S. and Huber, M. (2003). Subgoal discovery for hierarchical reinforcement learning using learned policies. In Russell, I. and Haller, S. M., editors, FLAIRS Conference, pages 346–350. AAAI Press.
- Kulkarni, T. D., Narasimhan, K., Saeedi, A., and Tenenbaum, J. B. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. NeurIPS 2016.
- Machado, M. C., Bellemare, M. G., and Bowling, M. H. (2017). A laplacian framework for option discovery in reinforcement learning. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, pages 2295–2304.
- Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 112(1): 181 – 211.

Slides, Paper, and Code:

http://rafati.net

Poster Session on Wednesday.