OPTION DISCOVERY USING DEEP SKILL CHAINING

Akhil Bagaria
Department of Computer Science
Brown University
Providence, RI, USA
akhil_bagaria@brown.edu

George Konidaris
Department of Computer Science
Brown University
Providence, RI, USA
gdk@brown.edu

Presenter： LinusWangg

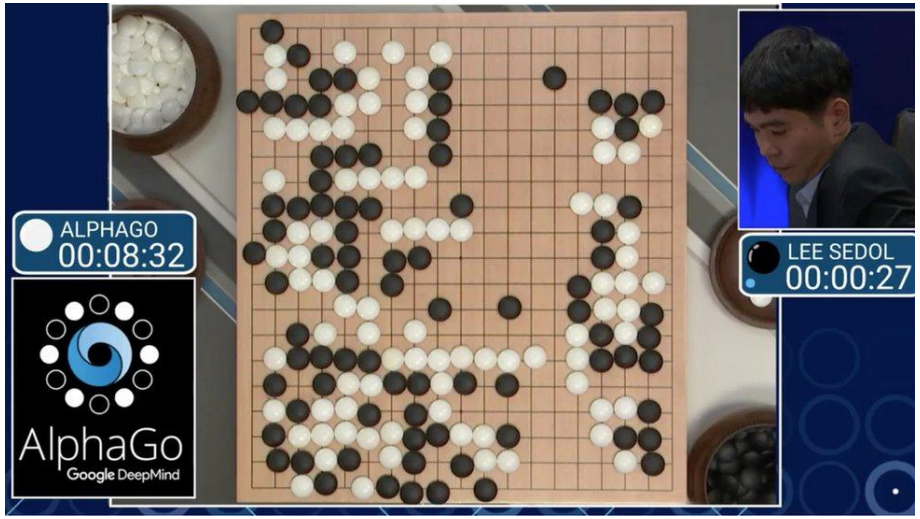
Background

Problems:

1. Simple algorithms suffer from the pressure of the feature representation.

2. Long-horizon tasks is hard to handle.

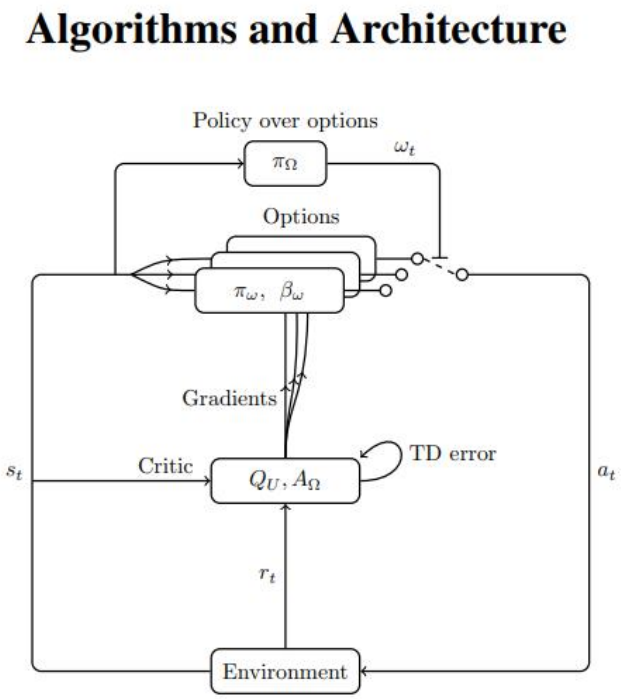
3. Sparse Reward.



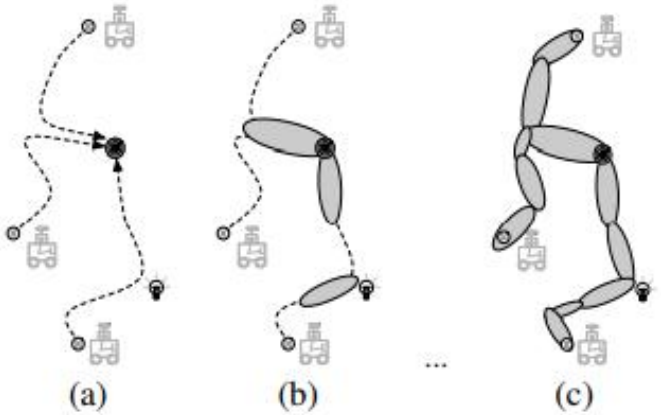
Alpha Go

FPS game

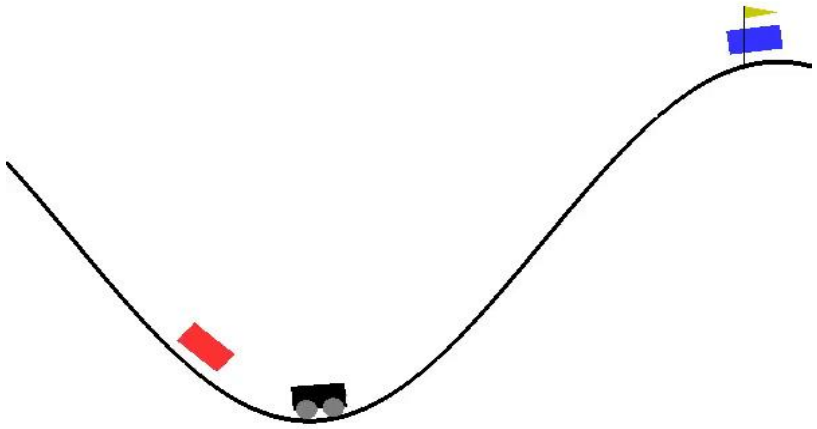
Literature reviews



Option-Critic Architecture



Skill Chaining



HAC

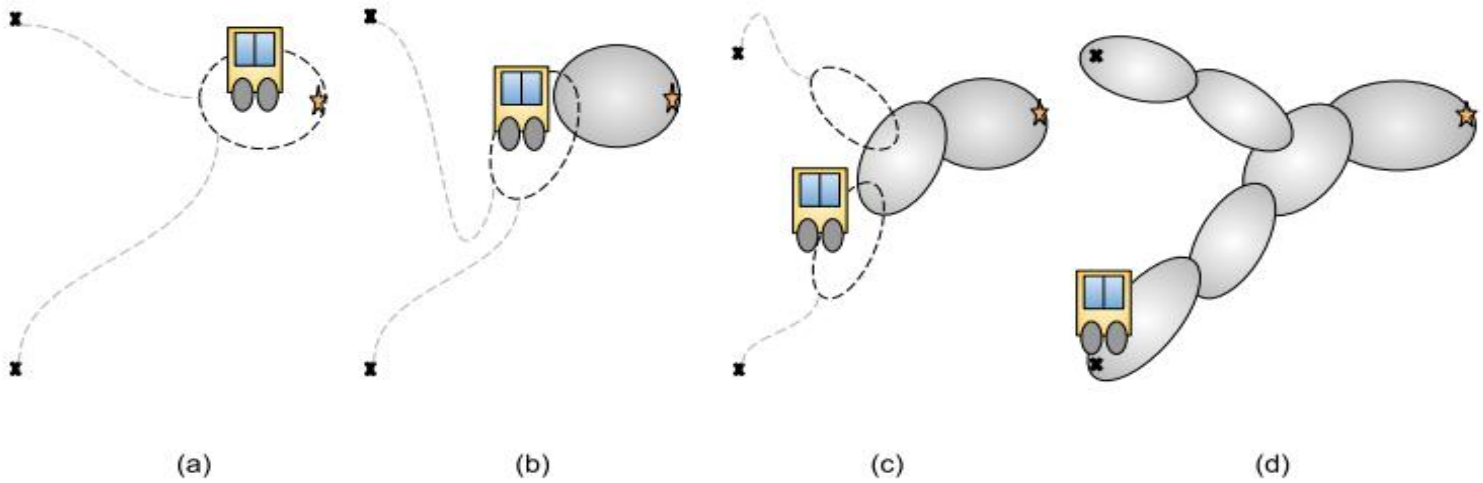
Method-Deep Skill Chaining

Implements:

1. Based on the Option-Critic Architecture.

2. Using the Neural Network.

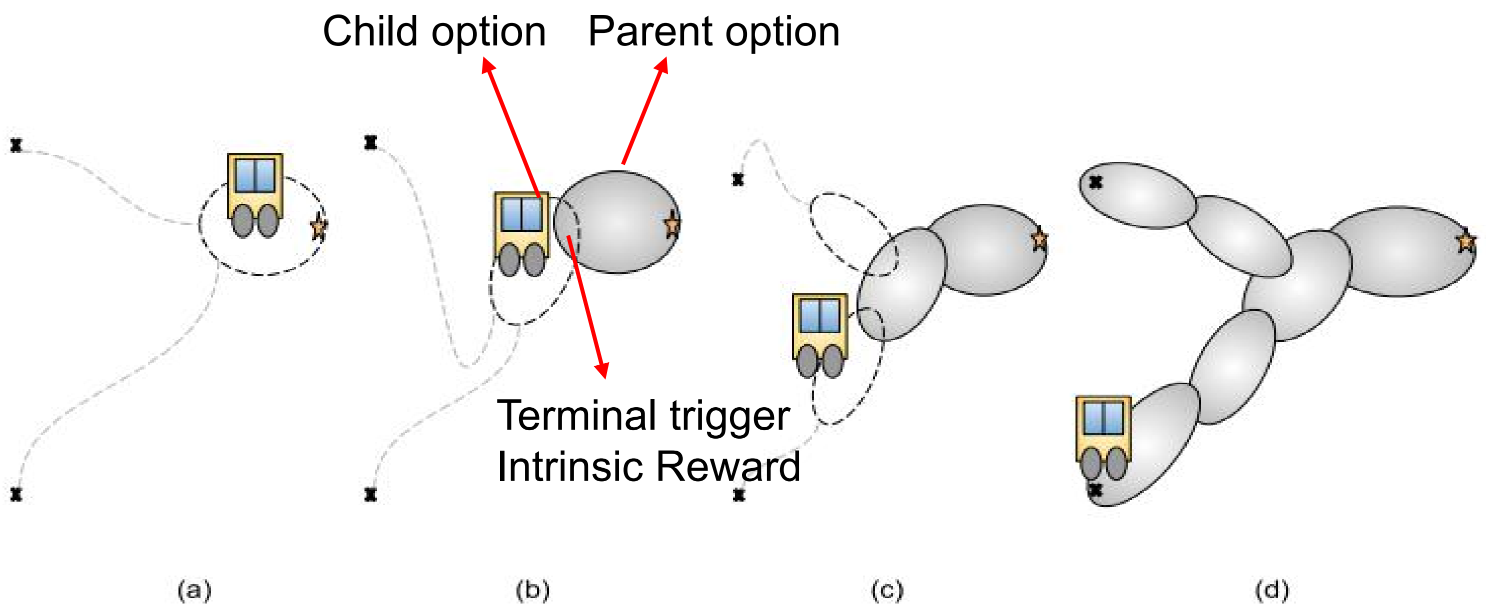
3. Extension of the original Skill Chaining.



Implement Graph

Method-Deep Skill Chaining

Graph illustration:



$$s_0, a_0, s_1, a_1, \ldots\ldots, s_t, a_t, \ldots\ldots, s_{t+H-1}, a_{t+H-1}, s_{t+H}, a_{t+H}, s_{t+H+1}, a_{t+H+1}, \ldots\ldots, s_T$$

$$s_0, a_0, s_1, a_1, \ldots\ldots, s_t, a_t, \ldots\ldots, s_{t+H-1}, a_{t+H-1}, s_{t+H}, a_{t+H}, s_{t+H+1}, a_{t+H+1}, \ldots\ldots, s_T$$

$$s_0, a_0, s_1, a_1, \ldots\ldots, s_t, a_t, \ldots\ldots, s_{t+H-1}, a_{t+H-1}, s_{t+H}, a_{t+H}, s_{t+H+1}, a_{t+H+1}, \ldots\ldots, s_T$$

Last K segments

Method-Deep Skill Chaining

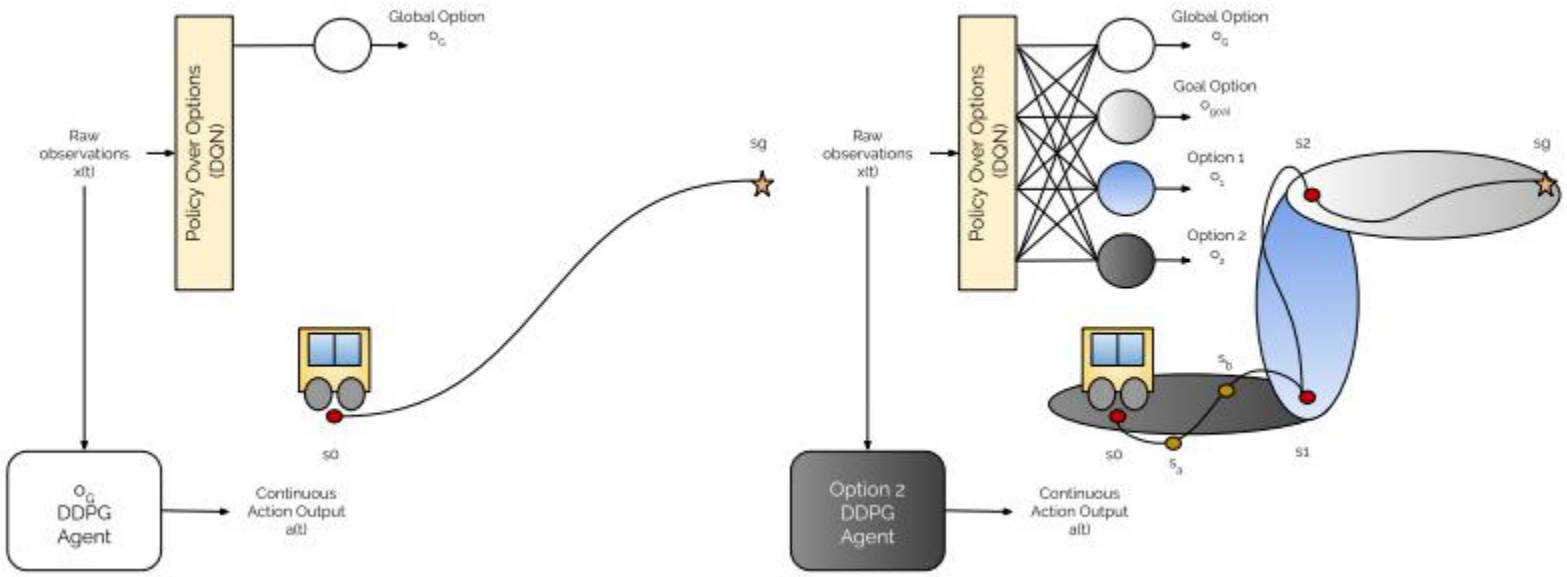
Option Select:

$$\mathcal{O}'(s_t) = \{o_i | \mathcal{I}_{o_i}(s_t) = 1 \cap \beta_{o_i}(s_t) = 0, \forall o_i \in \mathcal{O}\} \tag{2}$$

$$o_t = \underset{o_i \epsilon \mathcal{O}'(s_t)}{\arg\max} Q_\phi(s_t, o_i). \tag{3}$$

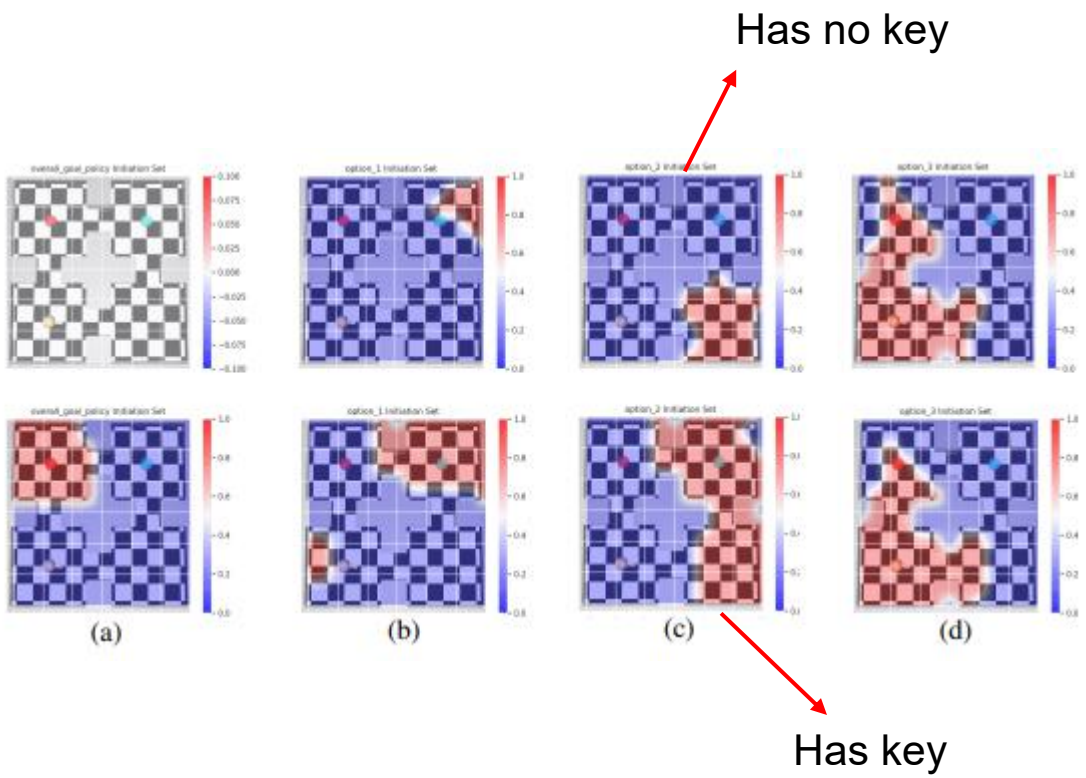
Option Value Update:

$$y_t = \sum_{t'=t}^{\tau} \gamma^{t'-t} r_{t'} + \gamma^{\tau-t} Q_{\phi'}(s_{t+\tau}, \underset{o' \epsilon \mathcal{O}'(s_{t+\tau})}{\arg\max} Q_\phi(s_{t+\tau}, o')). \tag{4}$$



Deep Q Learning

Experiment



Initialization Set Hot Map

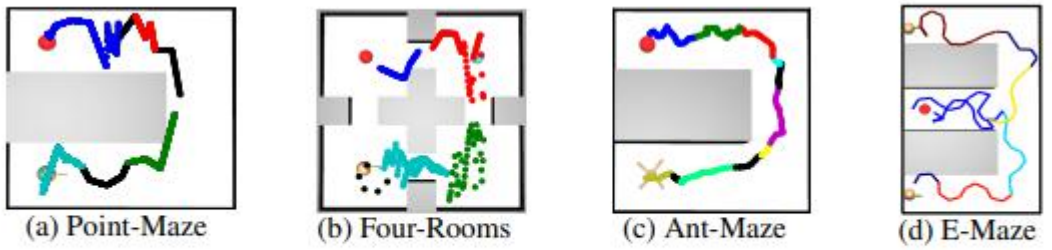
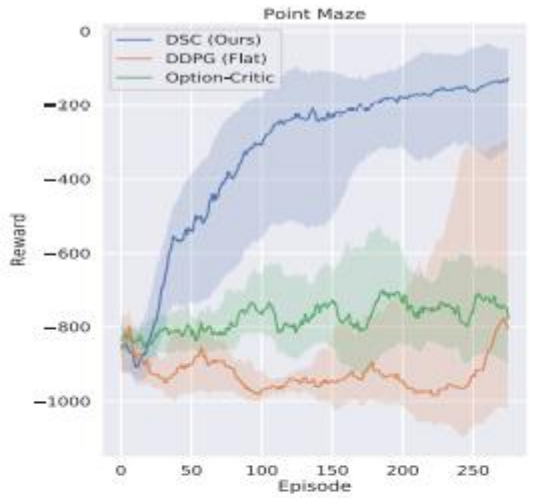
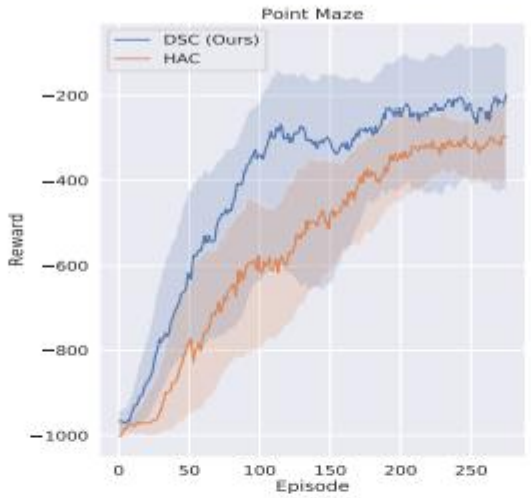


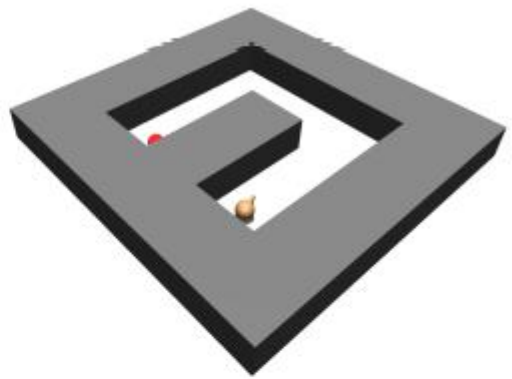
Figure 3: Solution trajectories found by deep skill chaining. Sub-figure (d) shows two trajectories corresponding to the two possible initial locations in this task. Black points denote states in which π_O chose primitive actions, other colors denote temporally extended option executions.

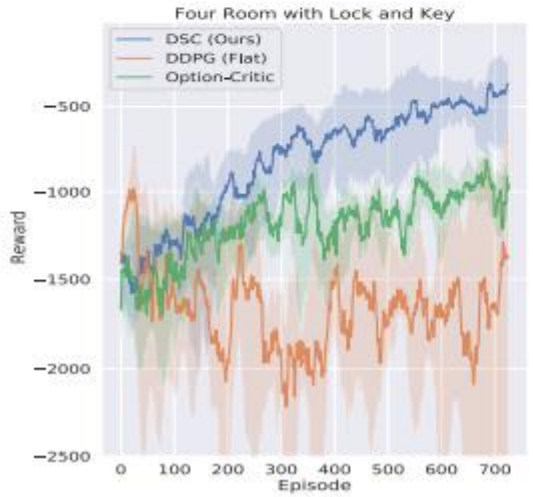
Individual Option Performance

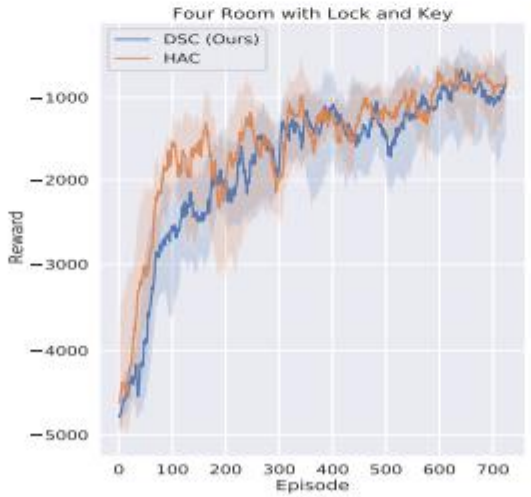
Experiment









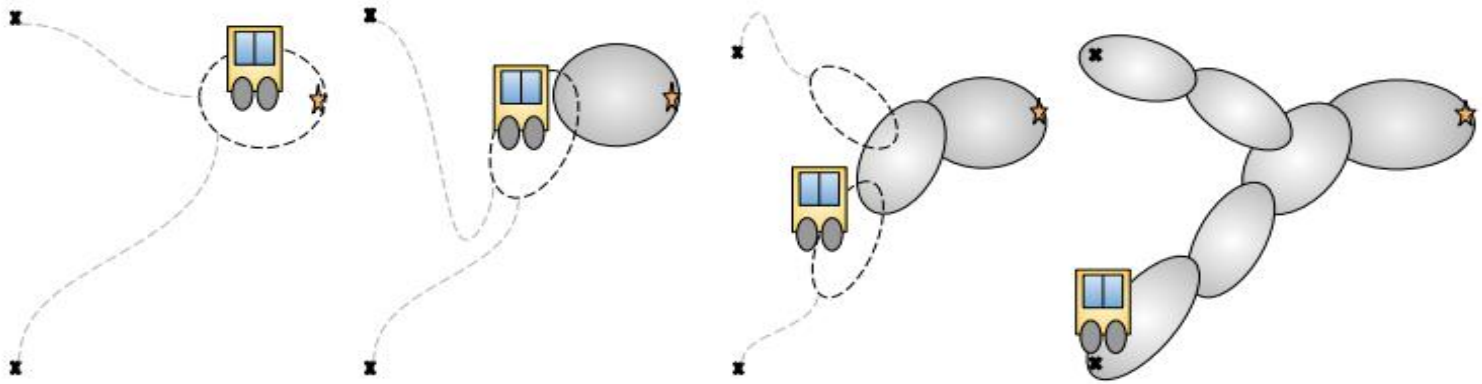


Reward Results

Discussions

Drawbacks:

1. This method require the agent to reach the goal state without any prior knowledge, it's too hard!
2. the gestation period of option is affected by the exploration.



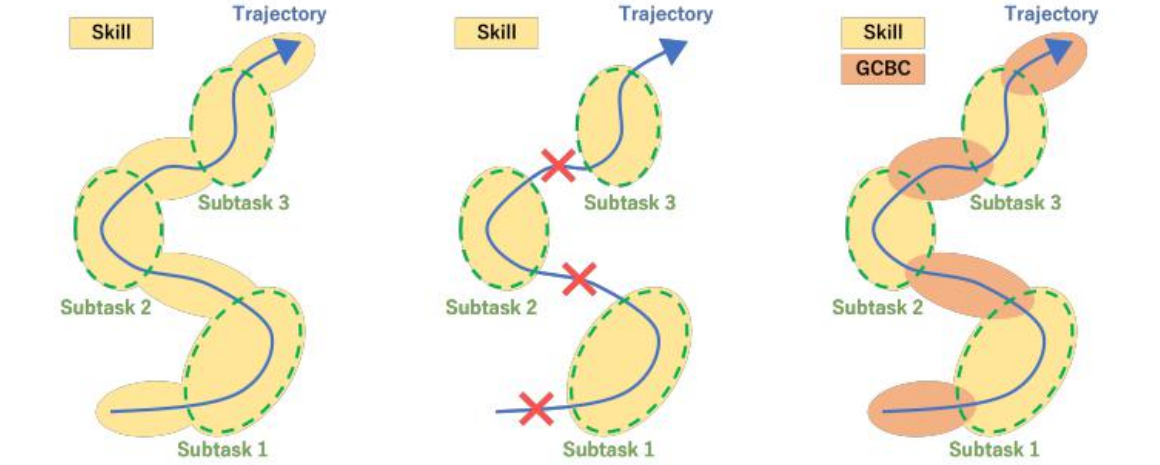
$$s_0, a_0, s_1, a_1, \ldots\ldots, s_t, a_t, \ldots\ldots, s_{t+H-1}, a_{t+H-1}, s_{t+H}, a_{t+H}, s_{t+H+1}, a_{t+H+1}, \ldots\ldots, s_T$$

Discussions

Solutions:

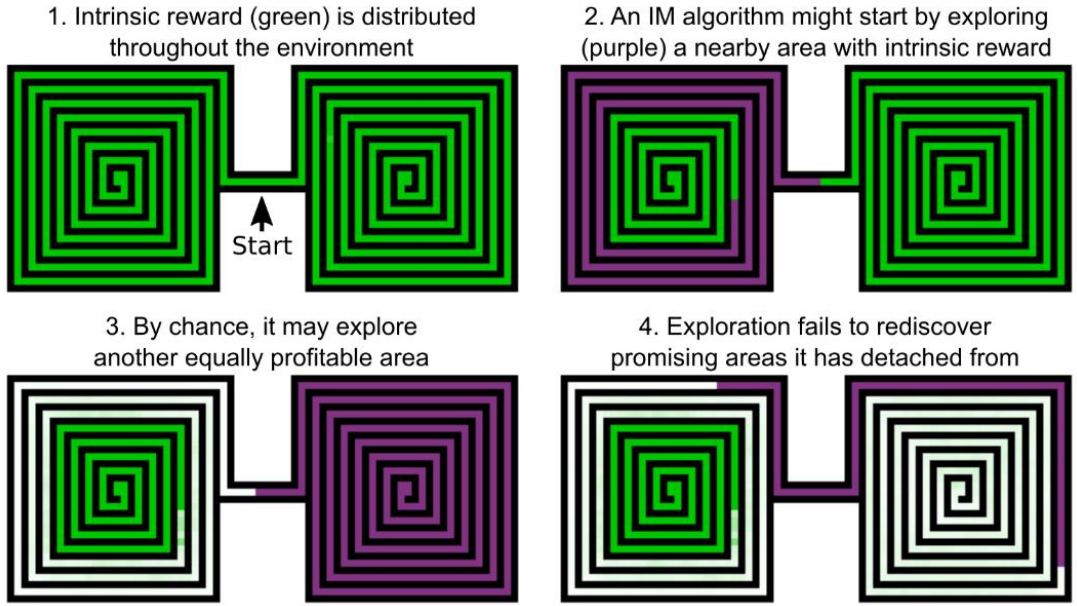
1. Skill extractions from the existing trajectories in offline RL or IL.

2. Use some more efficient explore policies.



(a) Previous skill-based imitation with demonstration-covering dataset (b) Previous skill-based imitation with subtask-local dataset (c) Our method with subtask-local dataset

Imitation Learning



Go explore

Conclusion

Architecture:

Option-Critic Architecture

Problems solved:

1. Automatically separate the long-horizon task into options.
2. Relieve the pressure of feature representation.
3. Realize the generalization of the Skill Chaining building.

Potential improvement:

1. Combining with some offline RL and IL algorithms.
2. Some good explore policies.

Thanks