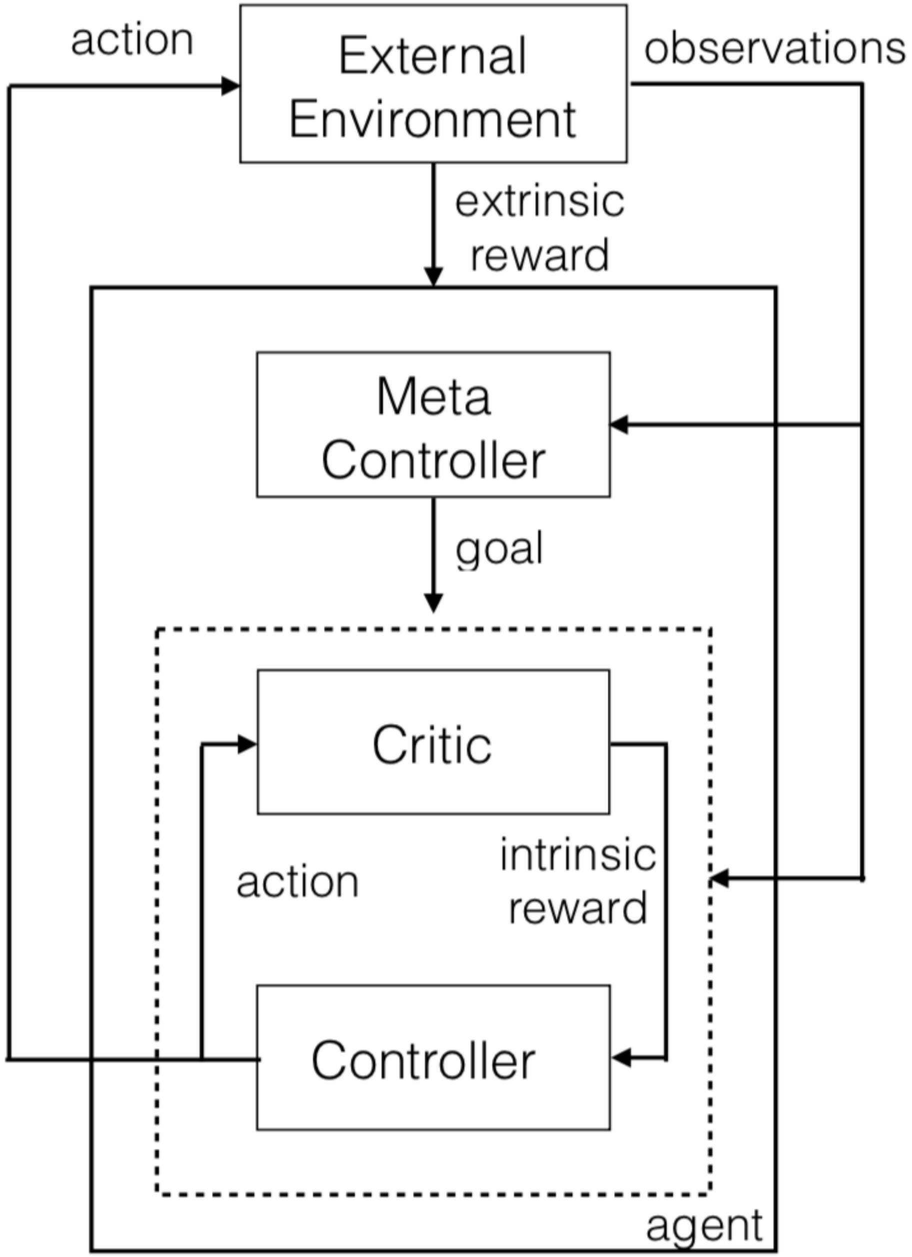
Artificial Intelligence Project: Hierarchy Deep Q-Learning

Yiheng Lin, Zhihao Jiang

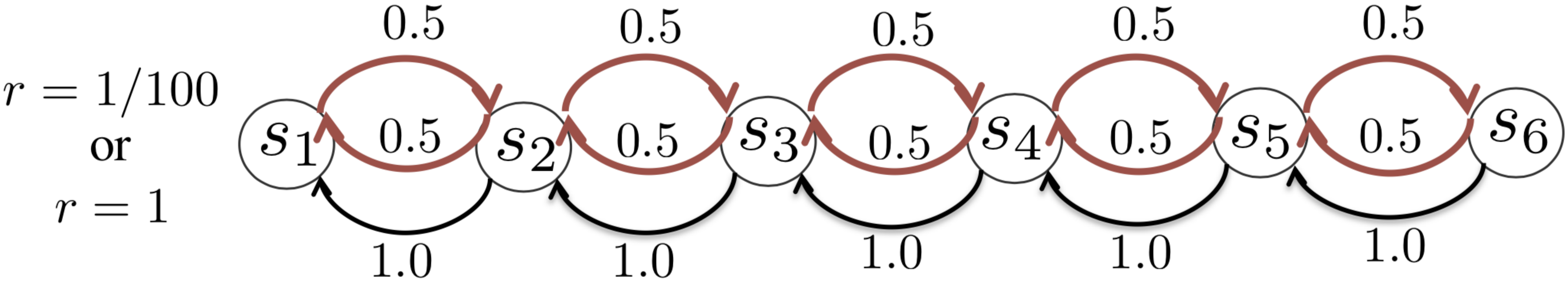
Introduction

Meta Controller:
(Higher Hierarchy)
Interact with External Environment (extrinsic reward), set goals for Controller;

Controller:
(Lower Hierarchy)
Try to achieve goals, receive intrinsic reward from Meta Controller.



Problem Setting

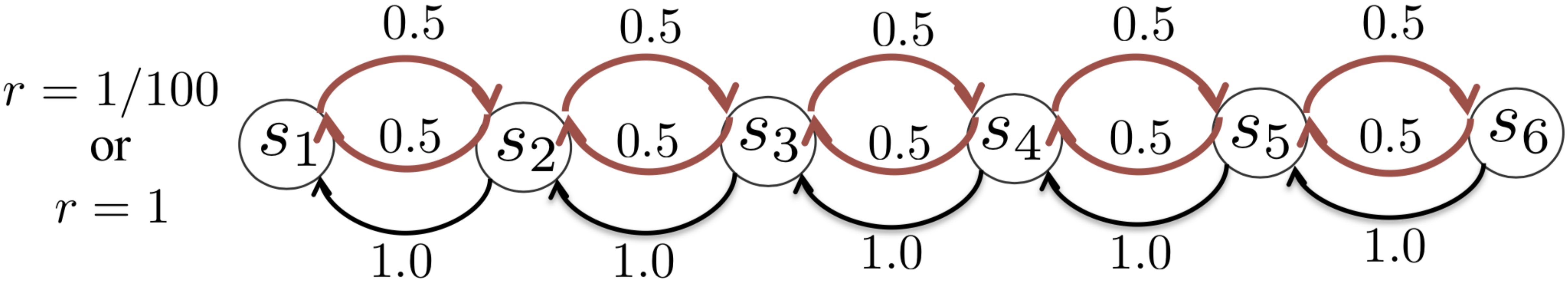


Reward: if s6 is visited, reward = 1; else, reward = 0.01

Actions: 1: move to left with probability 1;

2: move to right with probability 0.5; otherwise, move to left;

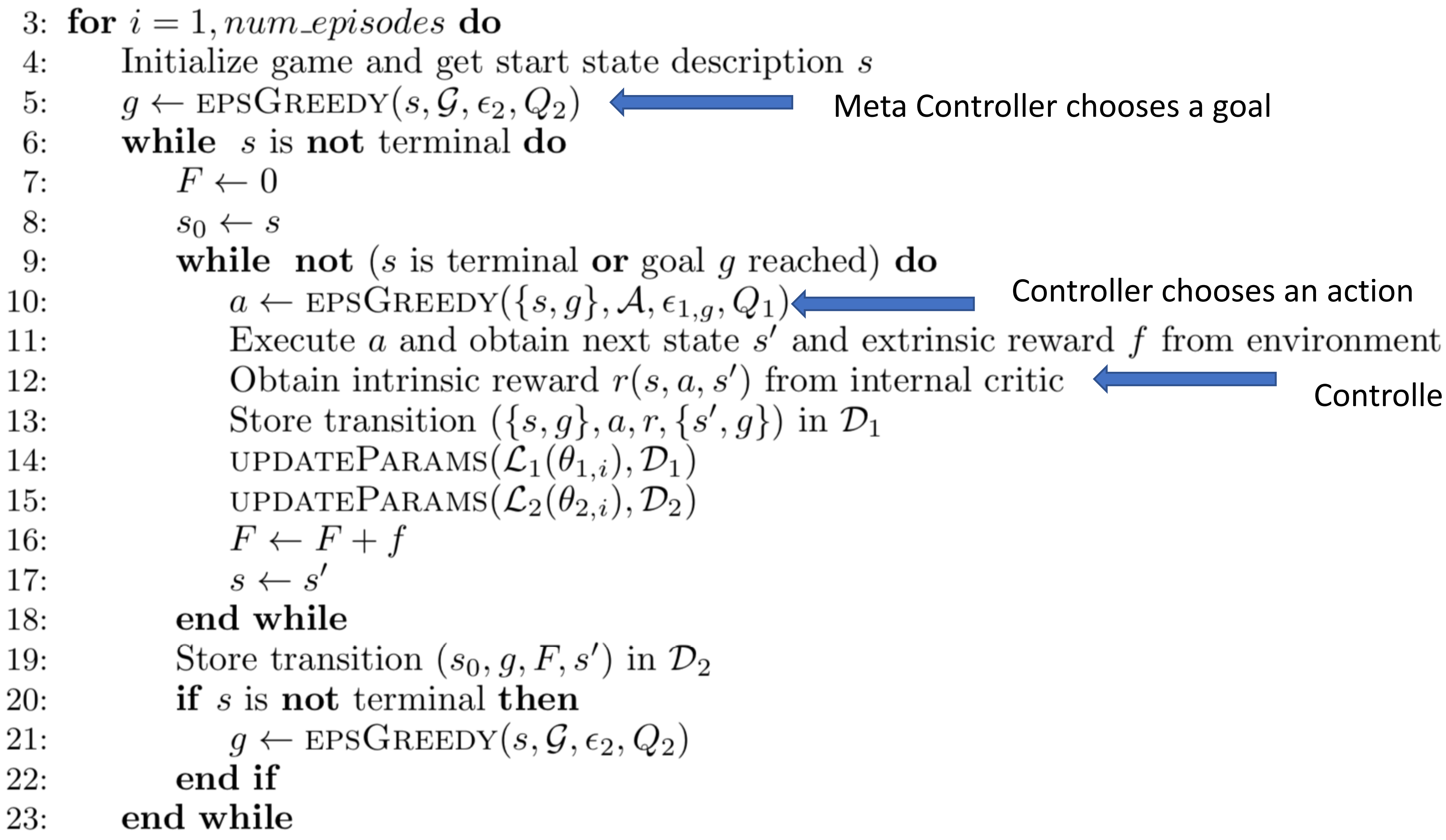
Difficulties



Hidden State: What if we have an additional state to indicate whether s6 is visited?

Search Efficiency: Can epsilon greedy agent keep taking action 2 for enough times?

Reimplementation



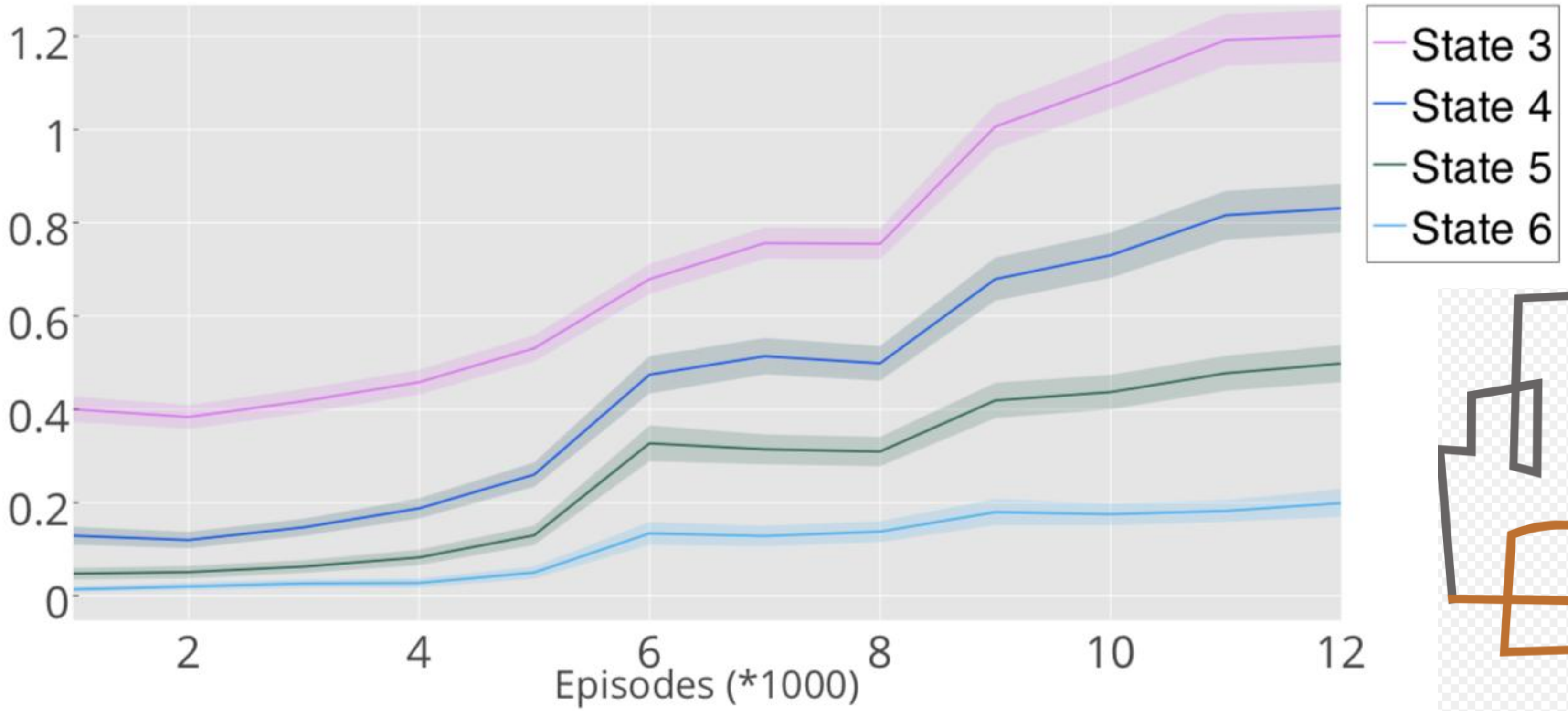
```
for $i = 1, num\_episodes$ do
    Initialize game and get start state description $s$
    $g \leftarrow \text{EPSGREEDY}(s, \mathcal{G}, \epsilon_2, Q_2)$
    while $s$ is not terminal do
        $F \leftarrow 0$
        $s_0 \leftarrow s$
        while not ($s$ is terminal or goal $g$ reached) do
            $a \leftarrow \text{EPSGREEDY}(\{s, g\}, \mathcal{A}, \epsilon_{1,g}, Q_1)$
            Execute $a$ and obtain next state $s'$ and extrinsic reward $f$ from environment
            Obtain intrinsic reward $r(s, a, s')$ from internal critic
            Store transition $(\{s, g\}, a, r, \{s', g\})$ in $\mathcal{D}_1$
            $\text{UPDATEPARAMS}(\mathcal{L}_1(\theta_{1,i}), \mathcal{D}_1)$
            $\text{UPDATEPARAMS}(\mathcal{L}_2(\theta_{2,i}), \mathcal{D}_2)$
            $F \leftarrow F + f$
            $s \leftarrow s'$
        end while
        Store transition $(s_0, g, F, s')$ in $\mathcal{D}_2$
        if $s$ is not terminal then
            $g \leftarrow \text{EPSGREEDY}(s, \mathcal{G}, \epsilon_2, Q_2)$
        end if
    end while
```

Meta Controller chooses a goal

Controller chooses an action

Controller’s reward

Reimplementation



Authors’ Result:

Ref: Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation

Reimplementation

Our result:

epoch	s1	s2	s3	s4	s5	s6
1	1	1.596	0.903	0.421	0.172	0.058
2	1	1.571	0.859	0.435	0.21	0.063
3	1	1.654	0.928	0.415	0.211	0.07
4	1	1.661	0.978	0.484	0.237	0.07
5	1	1.592	0.9	0.499	0.298	0.107
6	1	1.635	1.003	0.601	0.36	0.127
7	1	1.576	0.942	0.574	0.306	0.098

Ref: https://github.com/EthanMacdonald/h-DQN

Reimplementation

- Interesting Phenomena:

When goal is sited to s2, the controller often takes action 2…

Then it can visit s6…

Meta-Controller receives a big reward…

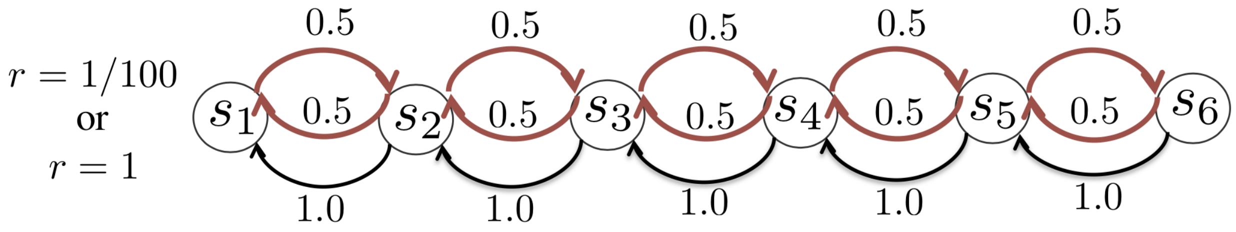
Meta-Controller tends to set s2 as the goal…

But, is this stable?

Maybe the experience of Controller is too volatile to train Meta Controller?

Quality input guarantees quality output…

Our Attempt



- Key idea: Explore efficiently

- The effect of subgoal in the previous example

- Other thinking

  Random subgoal

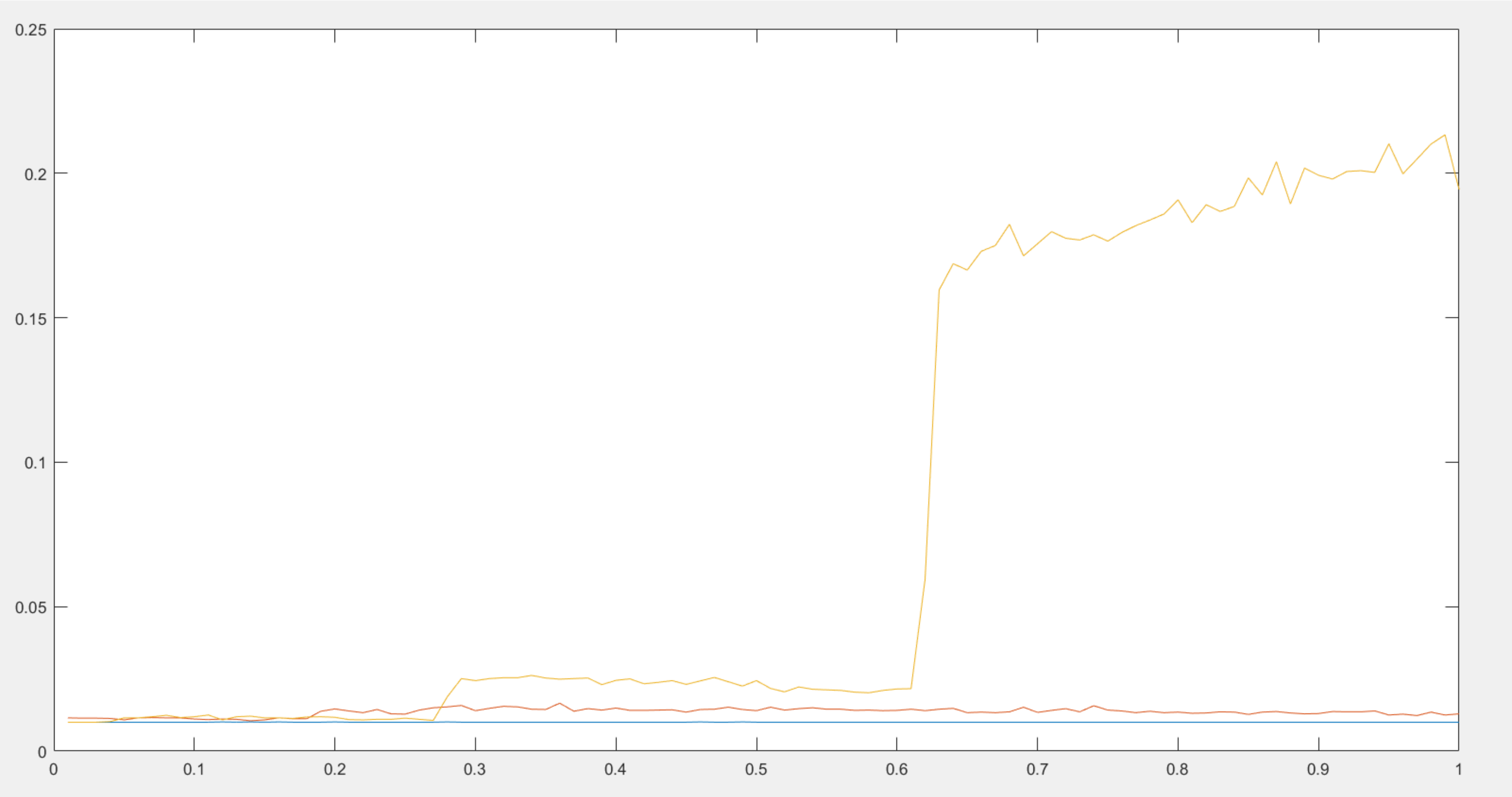
  “most unknown” subgoal

Implementation Details

- Set the subgoal

- Transmit the reward information

Performance



- Discovery: the line raises abruptly some time

Some Ideas

- Initial parameters in DQN using this method

- Combine this method with deep learning

Plan of Further Work

- Why the performance line raises abruptly
- How to implement our idea in more general model
- Is the ideas useful for DQN

Thanks