南京航空航天大學

Nanjing University of Aeronautics and Astronautics

Learning Representations in Model-Free Hierarchical Reinforcement Learning

Jacob Rafati David C. Noelle

AAAI 2018

How to Learn with sprase delayed reward ?

(1)

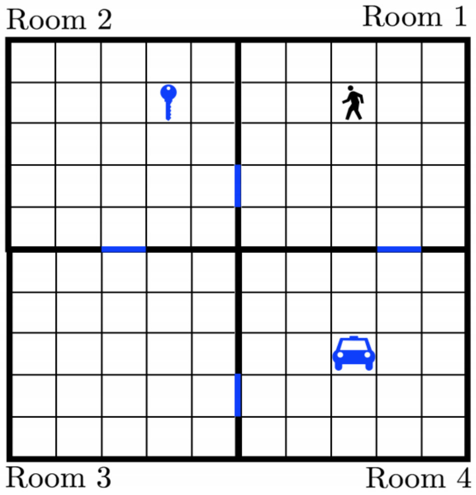
take “play” :

$r_{t+1}=1$, $r_{t+2}=...=r_{t+99}=0$, $r_{t+100}=-100$;

take “learn” :

$r_{t+1}=-1$, $r_{t+2}=...=r_{t+99}=0$, $r_{t+100}=100$;

(2)



action = {North,South, East, West}

r = +10 reaching the key

r = +100 moves to the car while carrying the key

r = −2 bumping to the wall

no reward or punishment for exploring the space

Hierarchical Reinforcement Learning

Main Idea :

Rely on an upper-level policy to decompose the entire task, and then use the lower-level policy to gradually execute。

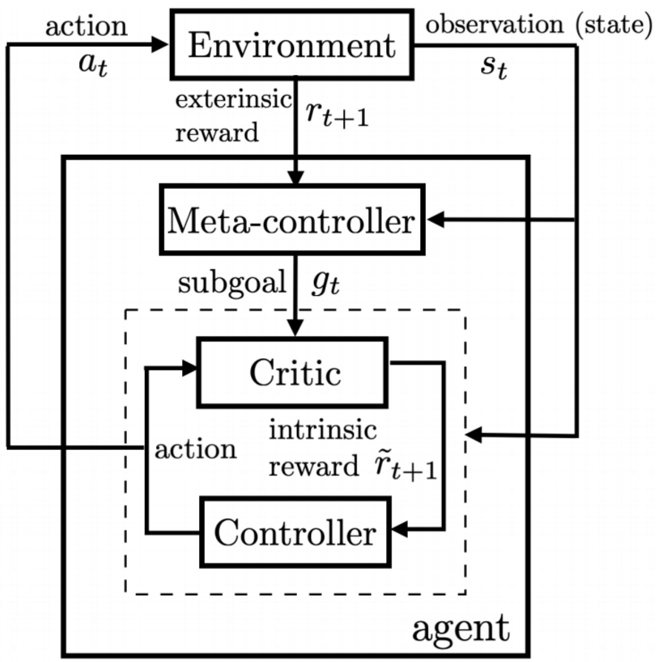
Problems :

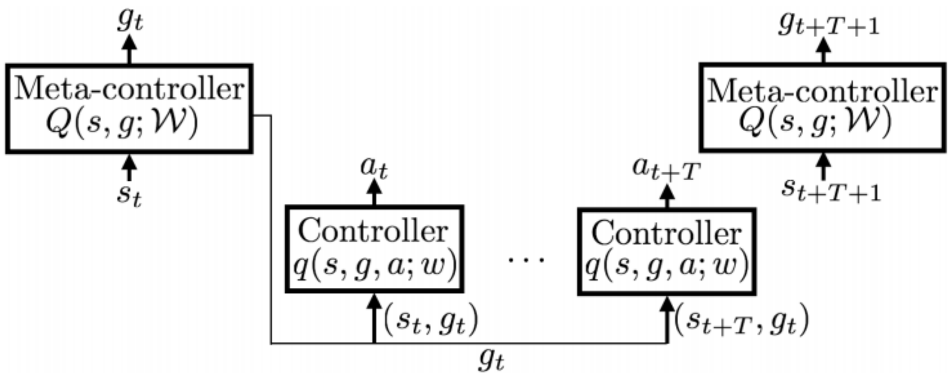
Subprolem1 : Learning a meta-policy to choose a subgoal

Subprolem2 : Exploring the state space while learning subtask through intrinsic motivation

Subprolem3 : Subgoal discovery

Meta-controller/Controller Framework





meta-controller :

$$G_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_t$$

$$Q(s,g) = \mathbf{E}_{\pi_g}\left[G_t | s_t = s, g_t = g\right]$$

$$\mathcal{L}(\mathcal{W}) \triangleq \mathbb{E}_{(s,g,G,s_{t'})\sim\mathcal{D}_2}\left[\left(G + \gamma \max_{g'} Q(s_{t'}, g'; \mathcal{W}) - Q(s, g; \mathcal{W})\right)^2\right]$$

controller :

$$\tilde{G}_t = \sum_{t'=t}^{t+T} \gamma^{t'-t} \tilde{r}_t(g)$$

$$q(s,g,a) = \mathbf{E}_{\pi_{ag}}\left[\tilde{G}_t | s_t = s, g_t = g, a_t = a\right]$$

$$L(w) \triangleq \mathbb{E}_{(s,g,a,\tilde{r},s')\sim\mathcal{D}_1}\left[\left(\tilde{r} + \gamma \max_{a'} q(s', g, a'; w) - q(s, g, a; w)\right)^2\right]$$

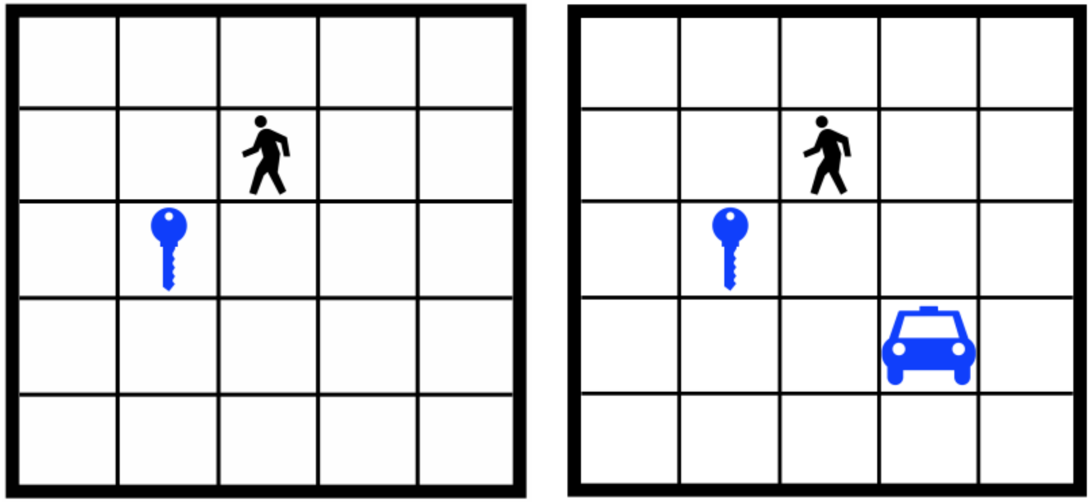
intrinsic motivation learning

Assuming that there is an oracle to give almost good subgoals , at least two benifits can get:

(1) exploration of large scale state spaces

(2) enabling the reuse of skills in varied environments

test

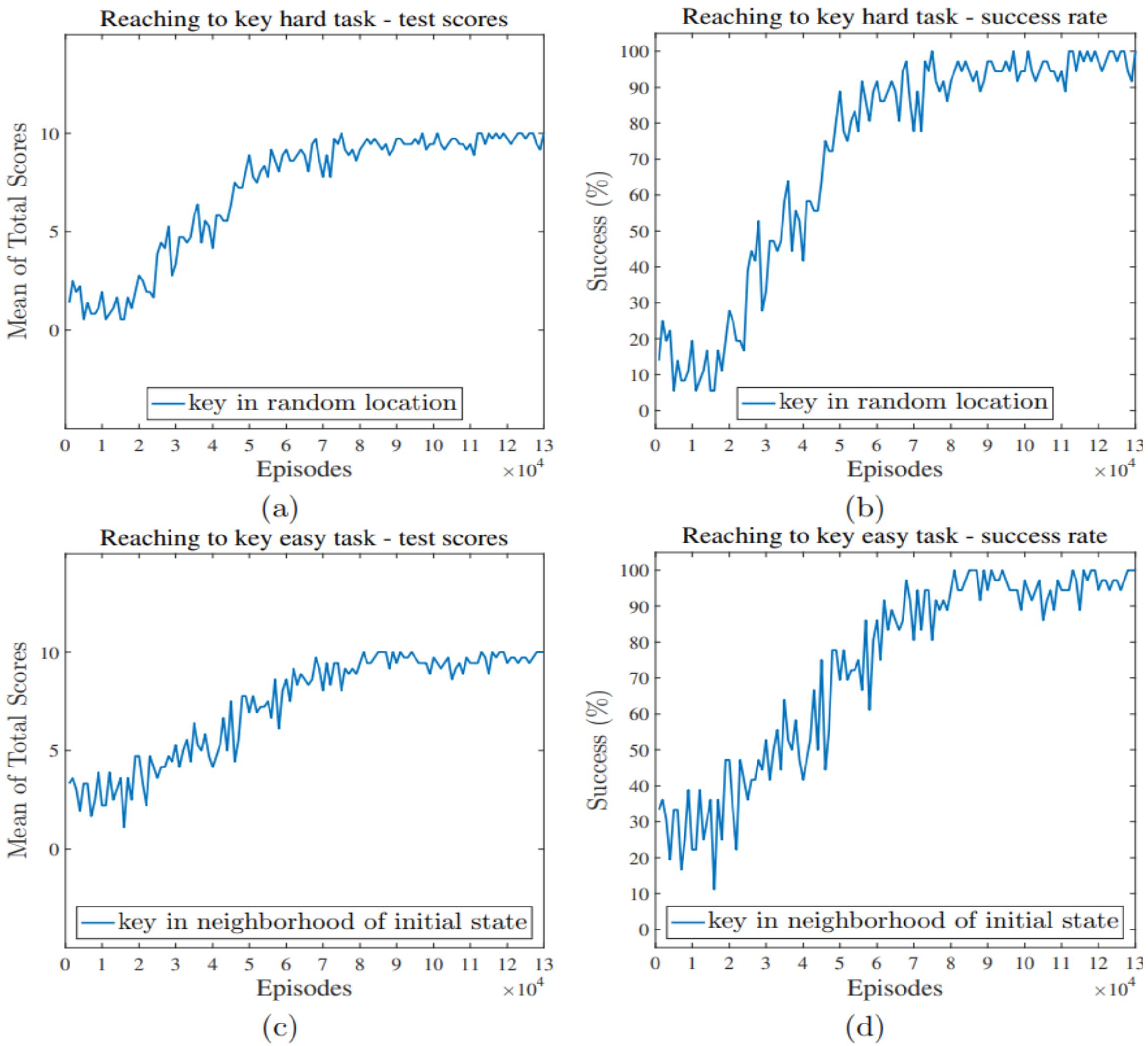


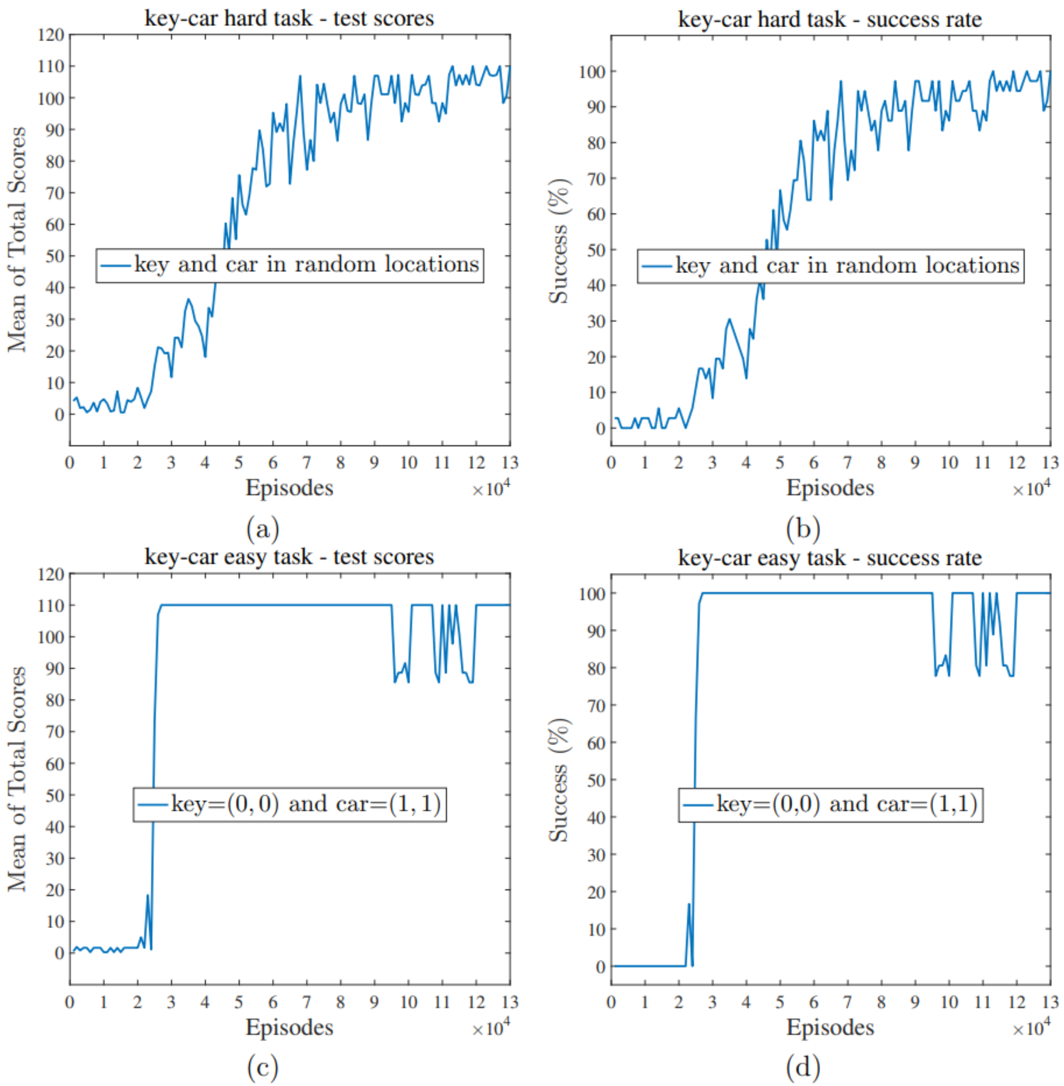
(a) (b)

$$\tilde{r}_{t+1} = \begin{cases} \min(r_{t+1}, -1) & \text{if } s_{t+1} \text{ is not terminal} \\ +1 & \text{if } s_{t+1} \text{ achieves the goal, } g_t \end{cases}$$

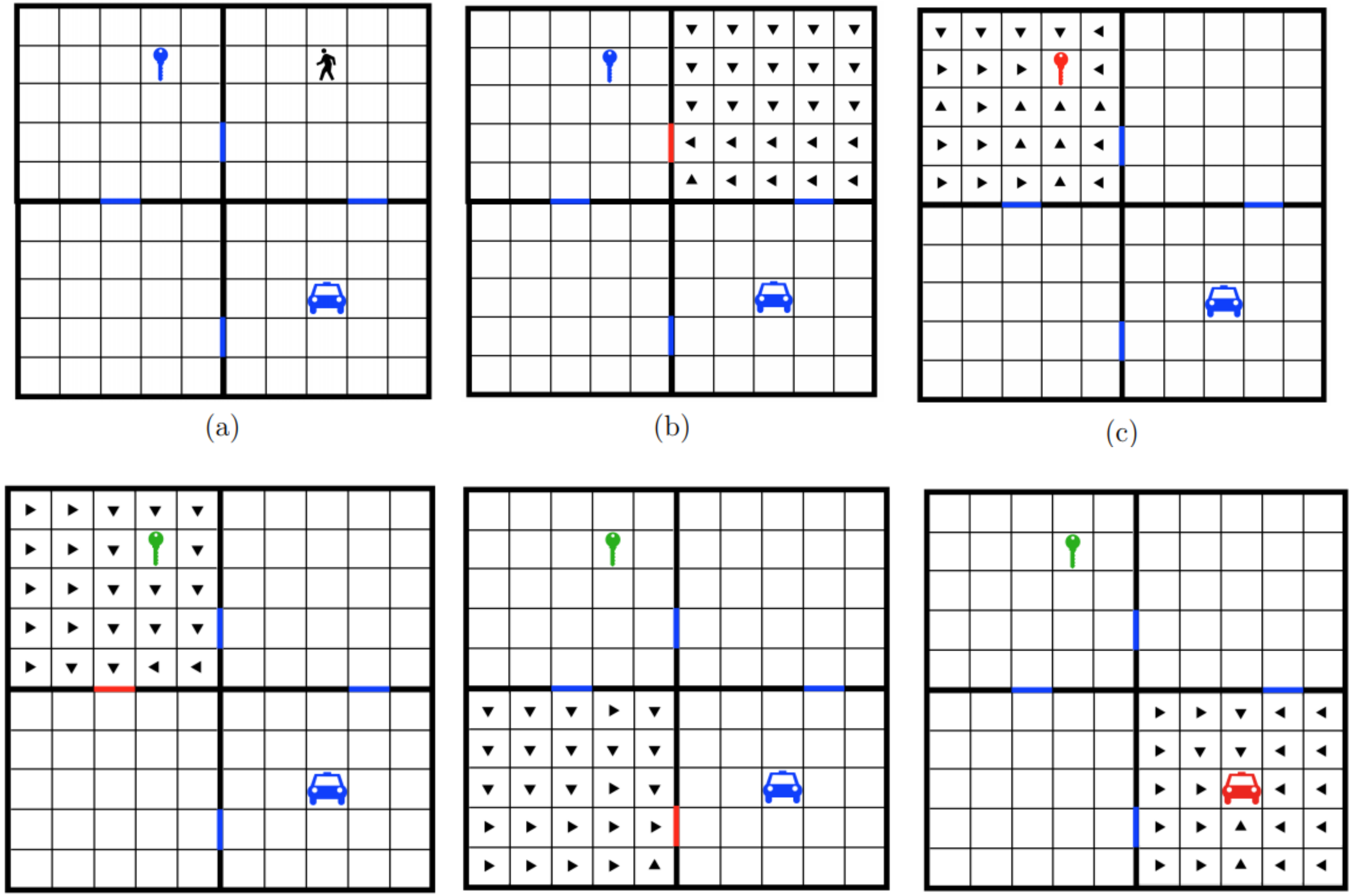
- **Key Task, Hard Placement.** In this simplified version of the task, the agent was trained to move to the key, producing a policy, π_{ag}, for reaching a randomly located goal g (key). This is illustrated in Figure 7(a). For each starting $s \in S$, a random goal, g, was assigned and the cumulative reward was obtained. We report the average reward scores and the average success percentage in Figure 8 (a) and (b), respectively.
- **Key Task, Easy Placement.** This version of the task is the same as the last, except that the goal, g, was always randomly placed in a location adjacent to the starting state, s. (See Figure 7 (a).) We report the average reward scores and the average success percentage in Figure 8 (c) and (d), respectively.
- **Key-Car Task, Hard Placement.** In this version of the task, both the key, g_{key}, and the car, g_{car}, were randomly placed. The agent received positive reward when the agent moved to the key (+10) and subsequently moved to the car (+100). (See Figure 7 (b).) We report the average scores and the average success percentage in Figure 9 (a) and (b), respectively.
- **Key-Car Task, Easy Placement.** This version of the task is the same as the last, except that the key was always located at $(0, 0)$, and the car was always located at $(1, 1)$. We report the average reward scores and the average success percentage in Figure 9 (c) and (d), respectively.

result


Reaching to key hard task - test scores
Mean of Total Scores
Episodes
×10^4
key in random location
(a)
Reaching to key hard task - success rate
Success (%)
Episodes
×10^4
key in random location
(b)
Reaching to key easy task - test scores
Mean of Total Scores
Episodes
×10^4
key in neighborhood of initial state
(c)
Reaching to key easy task - success rate
Success (%)
Episodes
×10^4
key in neighborhood of initial state
(d)


key-car hard task - test scores
Mean of Total Scores
Episodes
×10^4
key and car in random locations
(a)
key-car hard task - success rate
Success (%)
Episodes
×10^4
key and car in random locations
(b)
key-car easy task - test scores
Mean of Total Scores
Episodes
×10^4
key=(0, 0) and car=(1, 1)
(c)
key-car easy task - success rate
Success (%)
Episodes
×10^4
key=(0,0) and car=(1,1)
(d)

Reusing Learned Skills



(a) (b) (c) (d) (e) (f)

Good Subgoal Assumtions

(1) attending to the states associated with anomalous transition experiences.
(large rewards、large changes in state features)

(2) clustering experiences based on a similarity measure and collecting the set of associated states into a potential subgoal.

Methods:

merges anomaly (outlier) detection with the K-means clustering of experiences.

Subgoal discovery Algorithm

Algorithm 4 Unsupervised Subgoal Discovery Algorithm

for each $e = (s, a, r, s')$ stored in $\mathcal{D}$ **do**
  if experience e is an outlier (anomaly) **then**
    Store s' to the subgoals set $\mathcal{G}$
    Remove e from $\mathcal{D}$
  end if
end for
Fit a K-means Clustering Algorithm on $\mathcal{D}$ using previous centroids as initial points
Store the updated centroids to the subgoals set $\mathcal{G}$

Algorithm

Algorithm 5 Unified Model-Free HRL Algorithm

```
Pretrain controller using Algorithm 2 on a set of random subgoals G'
Initialize experience memories D, D_1 and D_2
Walk controller for M' episodes on random subgoals G', and store (s, a, s', r) to D
Run Unsupervised Subgoal Discovery on D to initialize G
for episode = 1, ..., M do
    Initialize state s_0 ∈ S, s ← s_0
    G ← 0
    g ← EPSILON-GREEDY(Q(s, G; W), ϵ_2)
    repeat for each step t = 1, ..., T
        compute q(s, g, a; w)
        a ← EPSILON-GREEDY(q(s, g, A; w), ϵ_1)
        Take action a, observe s' and external reward r
        Compute intrinsic reward r̃ from internal critic
        Store controller's intrinsic experience, (s, g, a, r̃, s') to D_1
        Store agent's transition experience, (s, a, r, s') to D
        Sample J_1 ⊂ D_1 and compute ∇L
        Update controller's parameters,   w ← w − α_1∇L
        Sample J_2 ⊂ D_2 and compute ∇ℒ
        Update meta-controller's parameters,   W ← W − α_2∇ℒ
        s ← s',   G ← G + r
        Decay exploration rate of controller ϵ_1
        if experience e is an outlier (anomaly) then
            Store s' to the subgoals set G
            Remove e from D
        end if
    until s is terminal or subgoal g is attained
    Decay exploration rate of meta-controller ϵ_2
    Store meta-controller's experience, (s_0, g, G, s') to D_2
    Fit a K-means clustering on D every N step to update centroids of G
end for
```

Algorithm 2 Intrinsic Motivation Learning

```
Specify Subgoals space G
Initialize w in q(s, g, a; w)
Initialize controller's experience memory, D_1
Initialize agent's experience memory, D
for episode = 1, ..., M do
    Initialize state s_0 ∈ S, s ← s_0
    Select a random subgoal g from G
    repeat for each step t = 1, ..., T
        compute q(s, g, a; w)
        a ← EPSILON-GREEDY(q(s, g, A; w), ϵ_1)
        Take action a, observe s' and external reward r
        Compute intrinsic reward r̃ from internal critic
        Store controller's intrinsic experience, (s, g, a, r̃, s') to D_1
        Store agent's experience, (s, a, s', r) to D
        Sample J_1 ⊂ D_1 and compute ∇L
        Update controller's parameters,   w ← w − α_1∇L
        s ← s'
        Decay exploration rate of controller ϵ_1
    until s is terminal or subgoal g is attained
end for
```

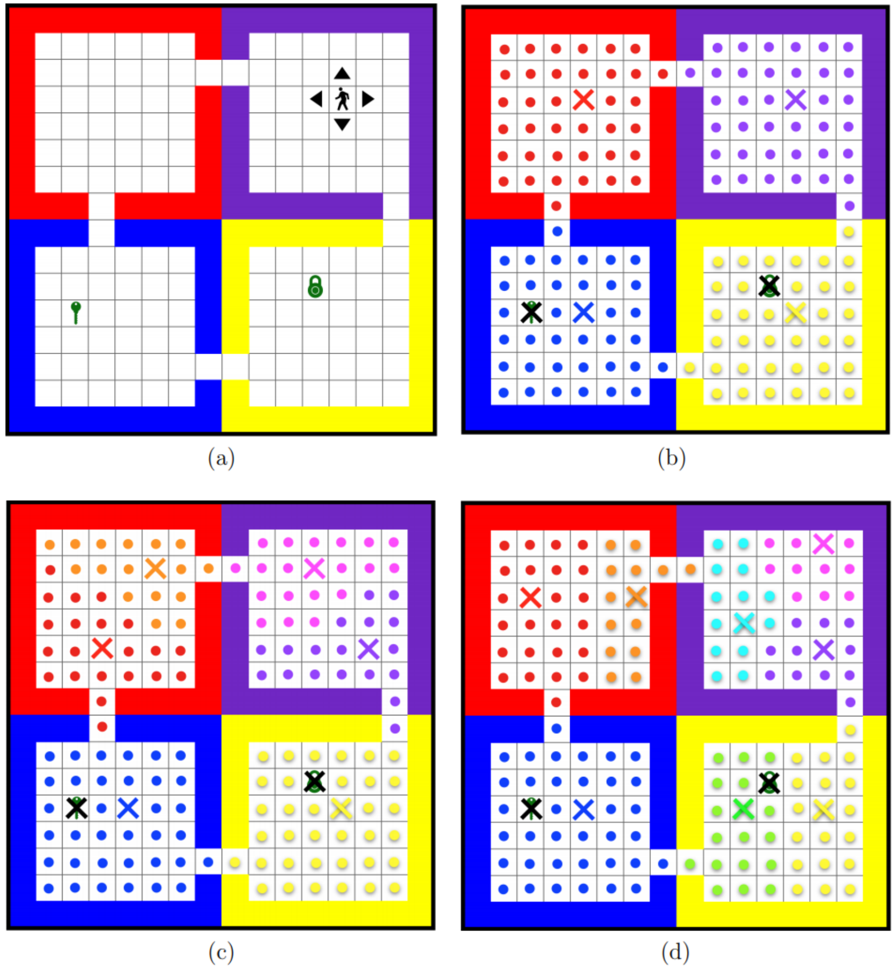
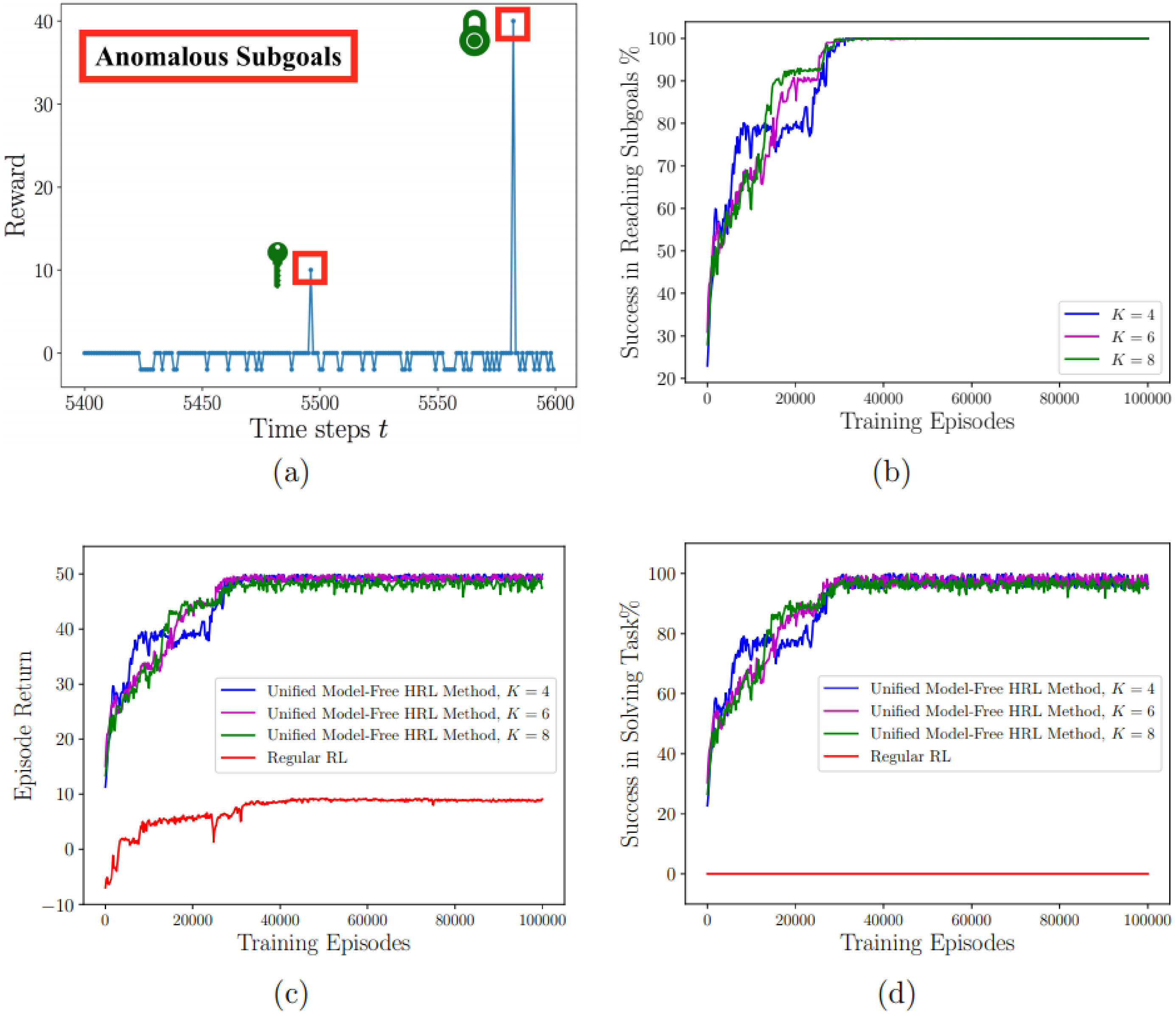


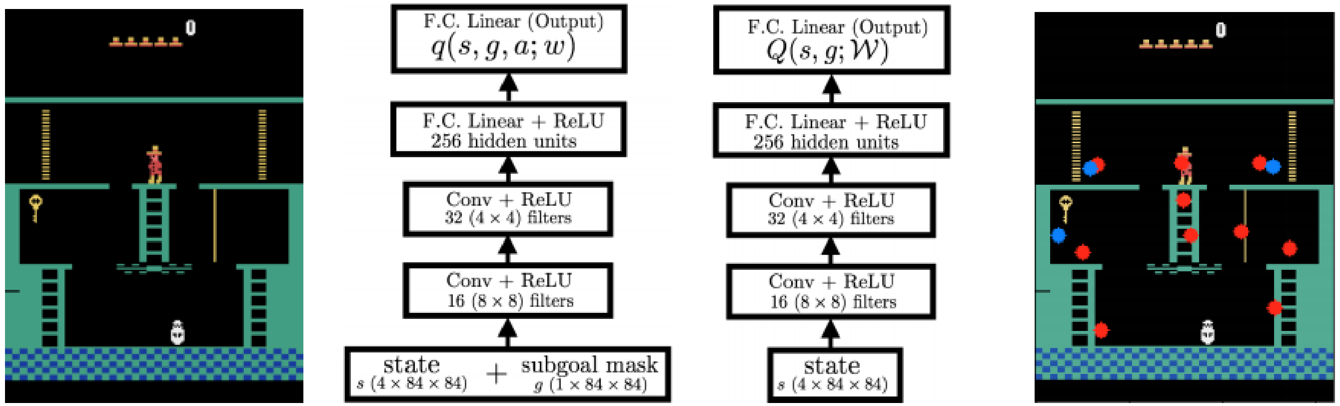
Figure 12: (a) The *4-room* task with a key and a lock. (b) The results of the unsupervised subgoal discovery algorithm with *anomalies* marked with black Xs and *centroids* with colored ones. The number of clusters in K-means algorithm was set to $K = 4$. (c) The result of the unsupervised subgoal discovery for $K = 6$. (d) The results of the unsupervised subgoal discovery for $K = 8$.

experiment

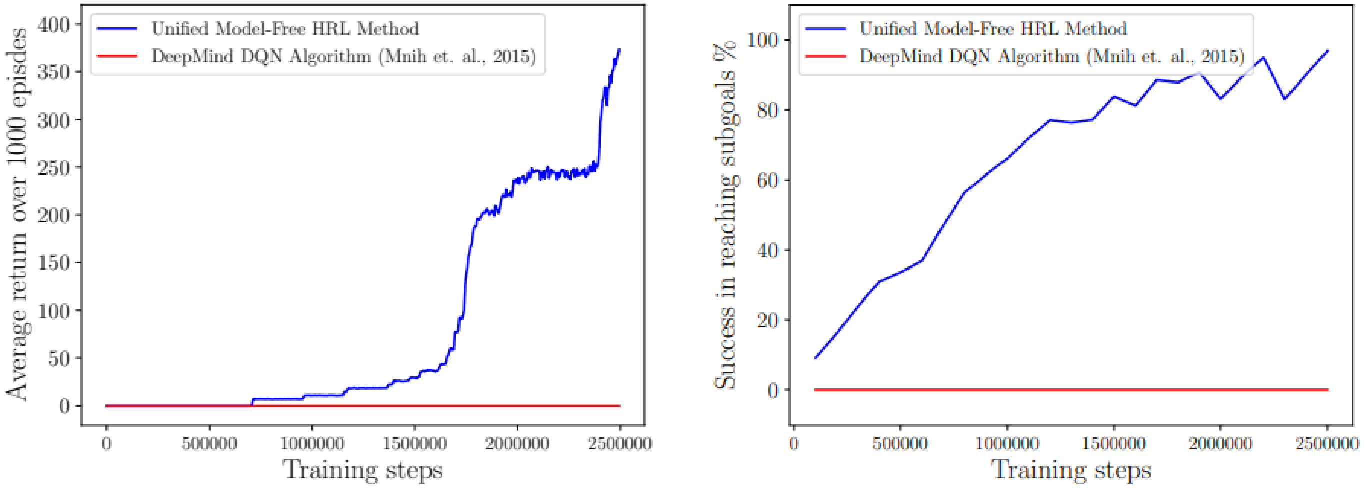


Anomalous Subgoals
Reward
Time steps t
(a)
Success in Reaching Subgoals %
Training Episodes
$K = 4$
$K = 6$
$K = 8$
(b)
Episode Return
Training Episodes
Unified Model-Free HRL Method, $K = 4$
Unified Model-Free HRL Method, $K = 6$
Unified Model-Free HRL Method, $K = 8$
Regular RL
(c)
Success in Solving Task%
Training Episodes
Unified Model-Free HRL Method, $K = 4$
Unified Model-Free HRL Method, $K = 6$
Unified Model-Free HRL Method, $K = 8$
Regular RL
(d)

Montezuma's Revenge



(a) (b) (c) (d)



(e) (f)