Q1)

$A_1 = 1$ $\quad R_1 = -1$

$A_2 = 2$ $\quad R_2 = 1$

$A_3 = 2$ $\quad R_3 = -2$

$A_4 = 2$ $\quad R_4 = 2$

$A_5 = 3$ $\quad R_5 = 0$

Q_0 =	0	0	0	0	0
Q_1 =	-1	0	0	0	0
Q_2 =	-1	1	0	0	0
Q_3 =	-1	-1	0	0	0
Q_4 =	-1	1	0	0	0
Q_5 =	-1	1	0	0	0

→ There is a possibility that action A_3 could have been a random selection

→ We can positively state that actions A_4 & A_5 are definitely ε-cases where selection was done ~~so~~ randomly to explore.

Q2)

$$Q_{n+1} = Q_n + \alpha_n [R_n - Q_n]$$

$$Q_{n+1} = \alpha R_n + (1-\alpha_n) Q_n$$

$$\cancel{Q_{n+1} = \alpha R_n + (1-\alpha)[Q_{n-1} + \alpha}$$

$$Q_{n+1} = \alpha_n R_n + (1-\alpha_n)[\alpha_{n-1} R_{n-1} + (1-\alpha_{n-1}) Q_{n-1}]$$

$$Q_{n+1} = \alpha_n R_n + \alpha_{n-1}(1-\alpha_n) R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})[\alpha_{n-2} R_{n-2} + (1-\alpha_{n-2}) Q_{n-2}]$$

$$Q_{n+1} = \alpha_n R_n + (1-\alpha_n)\alpha_{n-1} R_{n-1} + \dots + (1-\alpha_n)(1-\alpha_{n-1}) \dots (1-\alpha_2)\alpha_1 R_1 + (1-\alpha_n)(1-\alpha_{n-1}) \dots (1-\alpha_1) Q_1$$

$$Q_{n+1} = \left[\prod_{i=1}^{n} (1-\alpha_i)\right] Q_1 + \sum_{i=1}^{n} \left[\alpha_i R_i \cdot \prod_{j=i+1}^{n} (1-\alpha_j)\right]$$

3.

(a)

Equation 2.1:

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_i = a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i = a}}$$

By ~~th~~ looking at the equation, we can say that all the rewards are equally weighted and therefore this equation gives the ~~e~~ expectation (E) of $Q_t(a)$.

By the law of large numbers, $Q_t(a)$ will eventually converge to $q_*^*(a)$ as the number of samples reach infinity. So we can say that the ~~e~~ estimate is ~~a~~ unbiased if a ~~large~~ sufficiently large number is chosen.

$$E(Q_n) = E\left[\frac{R_1 + R_2 + \dots R_{n-1}}{n-1}\right] = E[E(R_{n-1})] = E[q^*] = q^*$$

$$E(Q_n) = q^*$$

Q3)

(d)

$$Q_n = Q_{n-1} + \alpha\left[R_{n-1} - Q_{n-1}\right]$$

If $Q_1 = 0$ and is far from q^*, ~~in th~~ initially for $n > 1$,

Q_n will be ~~biased~~ biased

$$Q_2 = \cancel{Q_1}^{0} + \alpha\left[R_1 - \cancel{Q_1}^{0}\right] = \alpha R_1$$

(c)

For Q_n to be unbiased, Q_1 should be set to q^*

$$Q_1 = q^*$$

3.

(d)

$$Q_{n+1} = Q_n + \alpha [R_n - Q_n]$$

$$E(Q_{n+1}) = E(Q_n) + \alpha [E(R_n) - E(Q_n)]$$

w.k.t.,

at $n \to \infty$, $E(R_n) \to q^*$

$$E(Q_{n+1}) = E(Q_n) + \alpha [q^* - E(Q_n)]$$

$$E(Q_{n+1}) = (1-\alpha) E(Q_n) + \alpha q^*$$

$$= (1-\alpha)\{(1-\alpha) E(Q_{n-1}) + \alpha q^*\} + \alpha q^*$$

$$= (1-\alpha)^n E(Q_1) + \alpha q^* + \alpha(1-\alpha) q^* + \alpha (1-\alpha)^2 q^* + \dots$$

$$= (1-\alpha)^n E(Q_1) + \sum_{i=1}^{n-1} \alpha (1-\alpha)^{n-1} q^*$$

$$= (1-\alpha)^n E(Q_1) + \alpha q^* \sum_{i=0}^{n-1} (1-\alpha)^i$$

$$\cancel{\alpha} q^* \left[\frac{1-(1-\alpha)^{n-1}}{\cancel{1}-(\cancel{1}-\cancel{\alpha})} \right]$$

$$= q^* [1-(1-\alpha)^{n-1}]$$

$$E(Q_{n+1}) = (1-\alpha)^n E(Q_1) + q^* \{1 - (1-\alpha)^{n-1}\}$$

As $n \to \infty$:

$$E(Q_{n+1}) = \cancelto{0}{(1-\alpha)^n} Q_1 + q^* [1 - \cancelto{0}{(1-\alpha)^{n-1}}]$$

$$\boxed{E(Q_{n+1}) = q^*}$$

3)

Exponential recency - weighted average is ~~of the form~~ given by

$$Q_{n+1} = Q_n + \alpha \left[R_n - Q_n \right]$$

It is of the form,

$$\boxed{\text{New\_Estimate} = \text{Old\_Estimate} + \text{Step\_Size} \left[\text{Target} - \text{Old\_Estimate} \right]}$$

The nature of the equation is to decrease the difference between the Reward and the estimate

4.

$$\pi_t(a) = \frac{e^{H_t(a)}}{e^{H_t(a)} + e^{H_t(b)}}$$

$$\boxed{\pi_t(a) = \frac{1}{1 + e^{H_t(b) - H_t(a)}}}$$

Sigmoid :

$$\boxed{\sigma(x) = \frac{1}{1 + e^{-x}}}$$

5) The algorithm ~~makes~~ ensures that the probability of the optimal action being selected is at least $1-\varepsilon$.

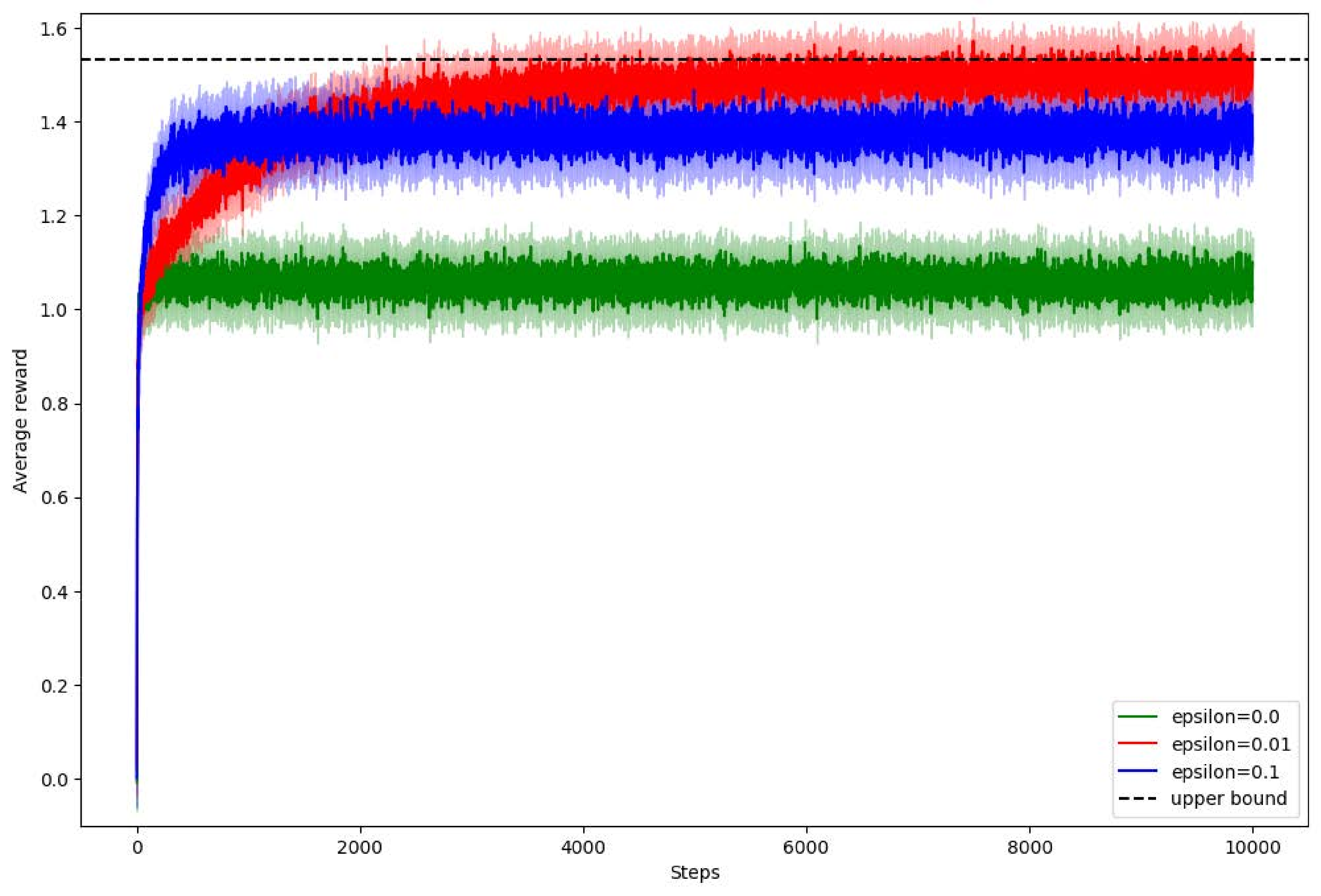
It is at least $1-\varepsilon$ because there is a chance that the optimal action will be chosen when the ~~alog~~ algorithm tries to explore ~~make~~ by making random selection.

6)

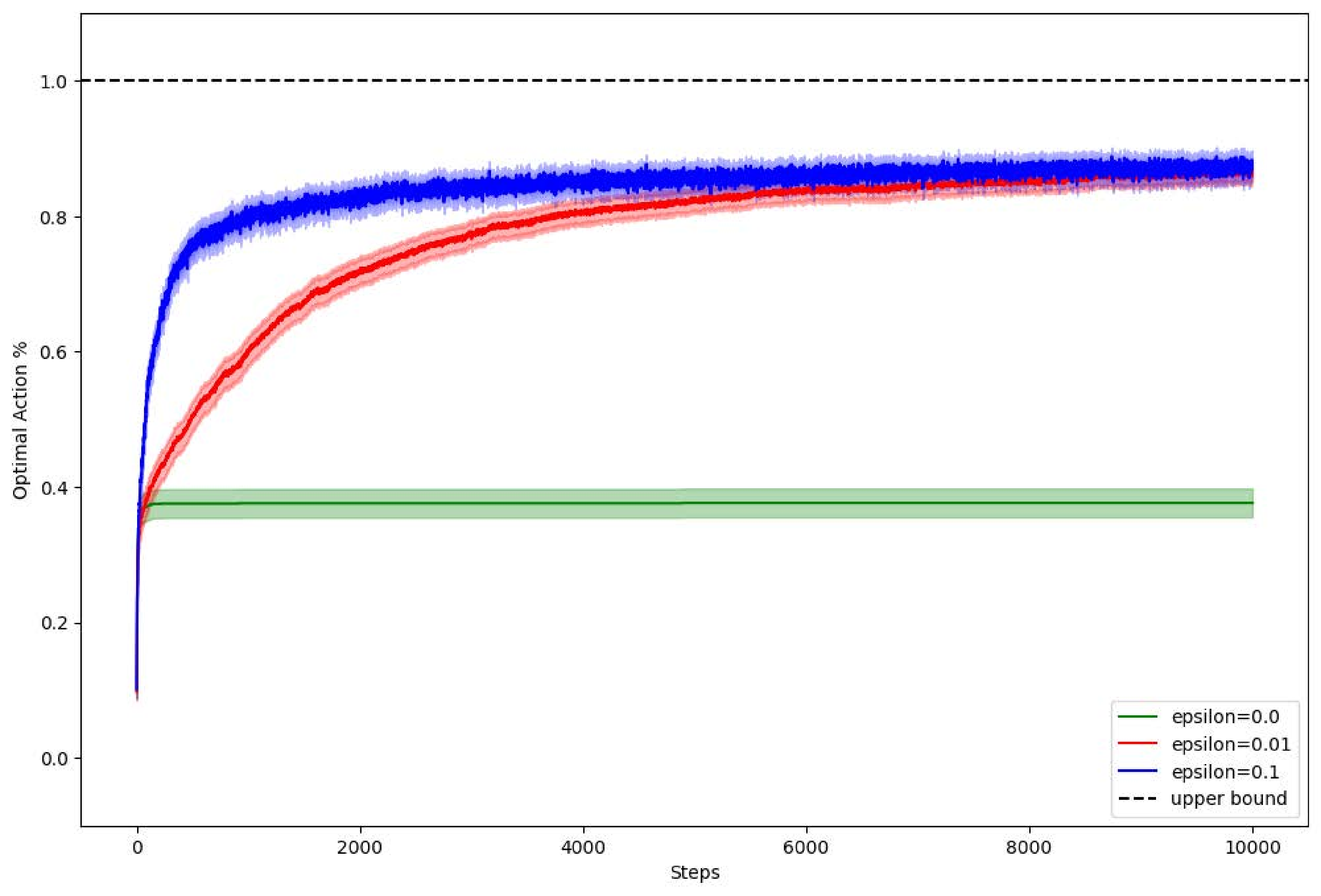
UCB starts slow since it starts off exploring all the actions. The sharp increase comes from the algorithm repeatedly choosing the action which ~~it~~ has the highest Q-value ensuring that it takes the most optimal action.

$$A_t = \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\ln(t)}{N_t(a)}} \right]$$

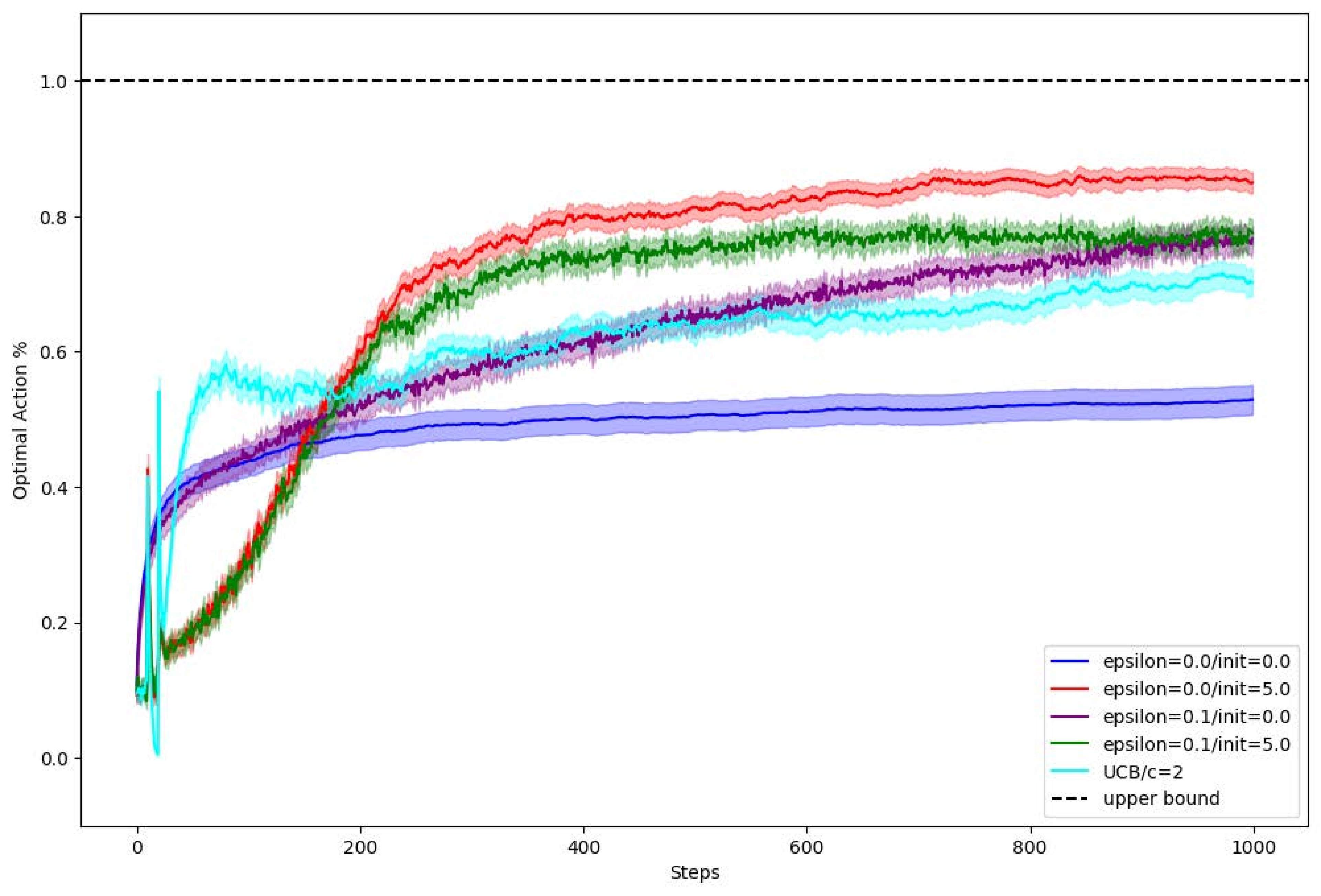
As the UCB keeps choosing the optimal actions, $\sqrt{\frac{\ln(t)}{N_t(a)}}$ decreases as $N_t(a)$ ~~we~~ increases ~~[illegible]~~ faster that $\ln(t)$. This ~~[illegible]~~ forces UCB to choose other actions until the ~~o~~ performance of the optimal action ~~[illegible]~~ becomes the highest. This is the reason for the sudden decrease in the % of optimal action chosen.



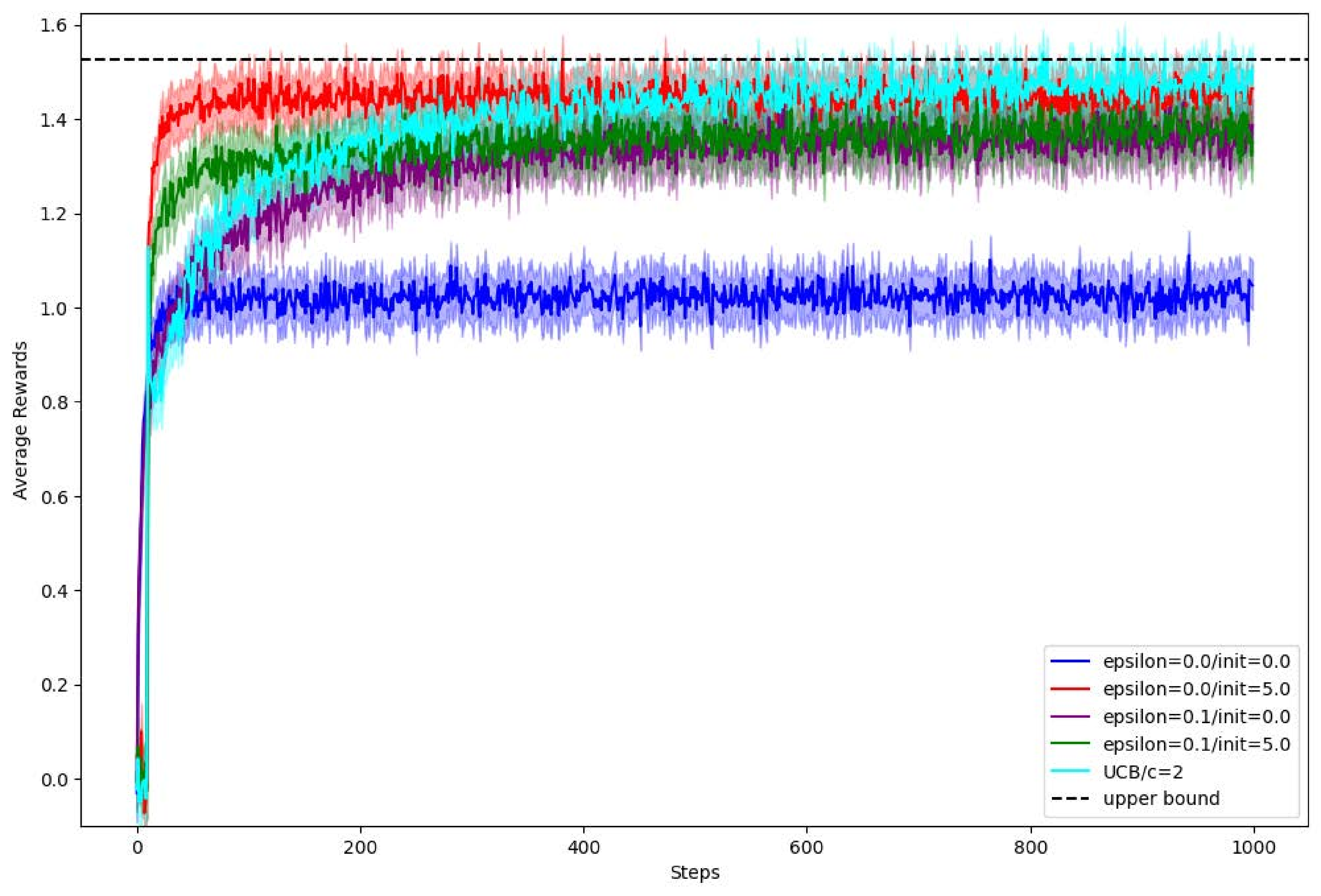
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Average reward
0
2000
4000
6000
8000
10000
Steps
epsilon=0.0
epsilon=0.01
epsilon=0.1
upper bound



1.0
0.8
0.6
0.4
0.2
0.0
Optimal Action %
0
2000
4000
6000
8000
10000
Steps
epsilon=0.0
epsilon=0.01
epsilon=0.1
upper bound



1.0
0.8
0.6
0.4
0.2
0.0
Optimal Action %
0
200
400
600
800
1000
Steps
epsilon=0.0/init=0.0
epsilon=0.0/init=5.0
epsilon=0.1/init=0.0
epsilon=0.1/init=5.0
UCB/c=2
upper bound



1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Average Rewards
0
200
400
600
800
1000
Steps
epsilon=0.0/init=0.0
epsilon=0.0/init=5.0
epsilon=0.1/init=0.0
epsilon=0.1/init=5.0
UCB/c=2
upper bound