Expectation

Input: Behaviour policy π, target policy π_i, discount factor gamma, initial state S_0.

$R \leftarrow []$, discounted rewards $\leftarrow 0$, ~~A←[]~~ $A \leftarrow []$, $Q \leftarrow [\,][\,]$

for $j = 0, 1, 2 .. t$ do: // t is terminal state

    Take an action a_j at s_j according to π, collect reward r_j and move on to s_{j+1}

    Store $R[j] \leftarrow r_j$

    Store $A[j] \leftarrow a$

    Store $Q[j][a] = \pi(a|s_j)$

$s = 0$

for r in ~~R do~~ reverse(R) do:

    if r is last element:

        $s = 0$

    else:

        $s = r + \gamma * s$

    ~~A~~ //Create a new list R1,

    R1.append(s) at 0^{th} position

~~for j in A do:~~

    ~~$L_\pi = \pi_j($~~

sum=0

for j in A do:

$$L_\pi = \frac{\pi_i(a|s_j)}{Q[j][A[j]]} \times R_1[j]$$

    sum += L_π

$$surr = sum - \frac{(4 \times max(R_1) \times gamma) \times d\_kl(\pi, \pi_i)}{(1-\gamma)^2}$$

return surr