Indian Institute of Technology, Madras
CS6700: Reinforcement Learning
Reinforcement Learning Programming Assignment-III Report

Rajan Kumar Soni - CS18S038

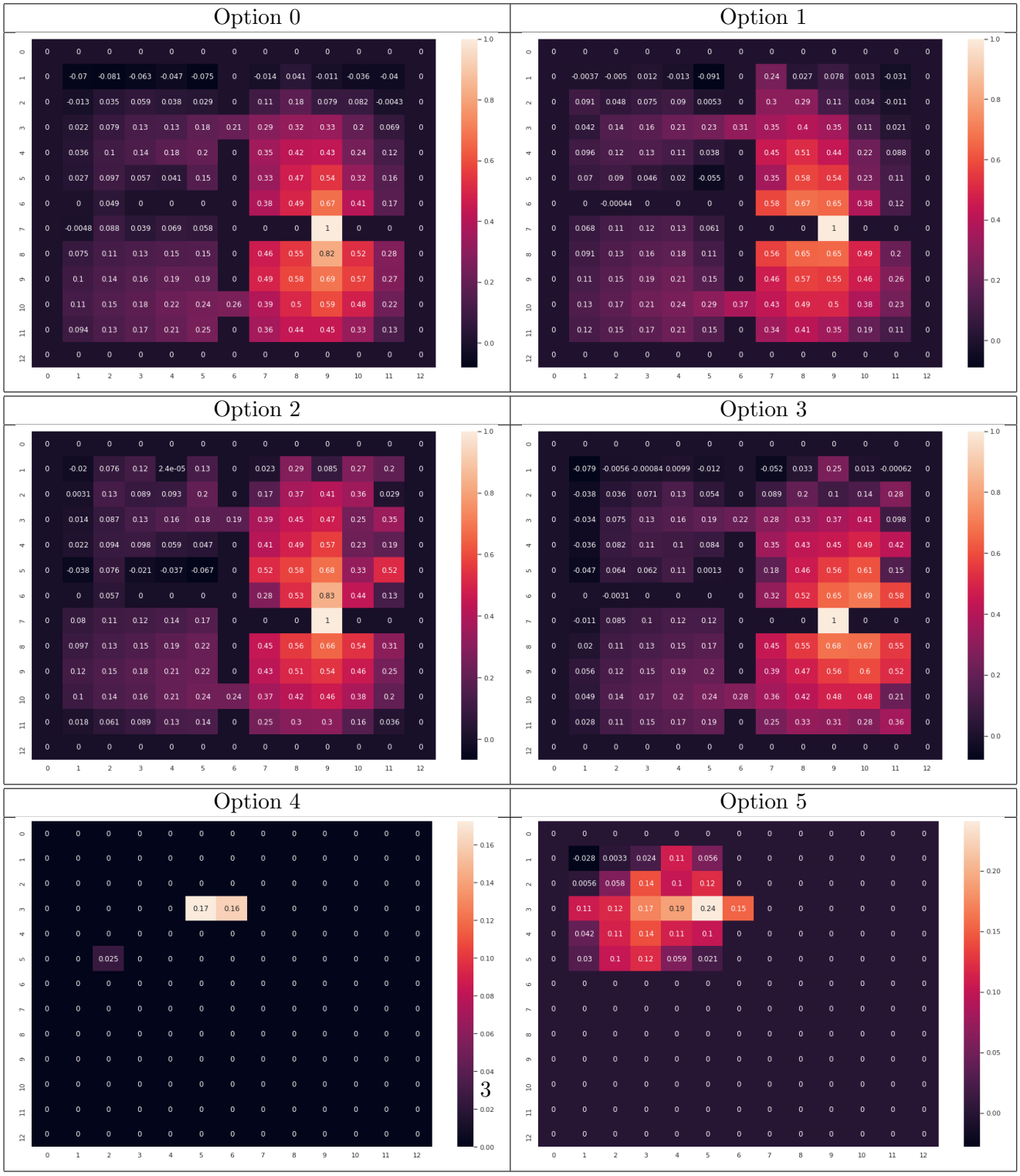
March 2020

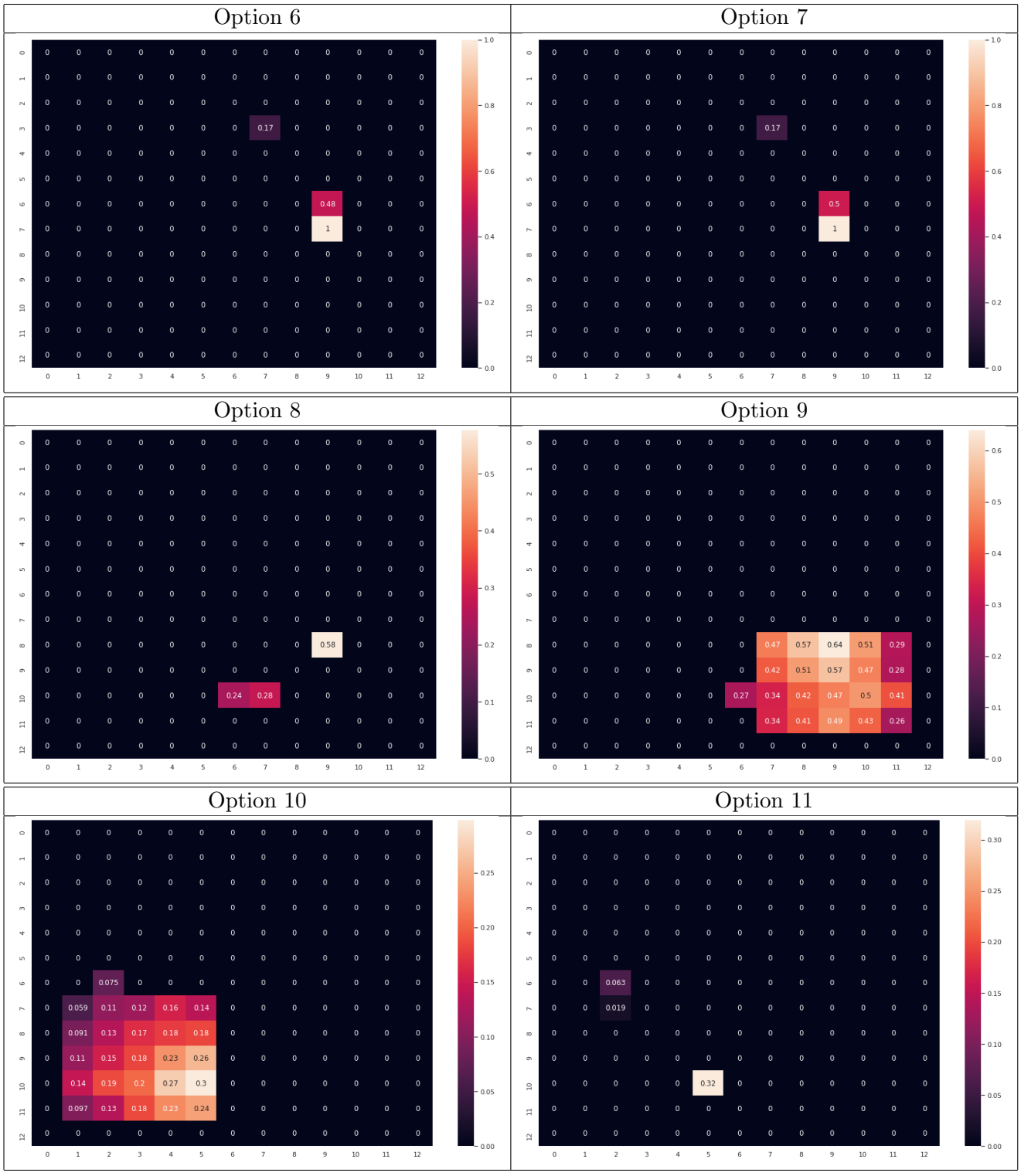
Contents

1 Q-Learning(Option 0,1,2,3 corresponds to UP, RIGHT, DOWN, LEFT) and other option named as clockwise according to room

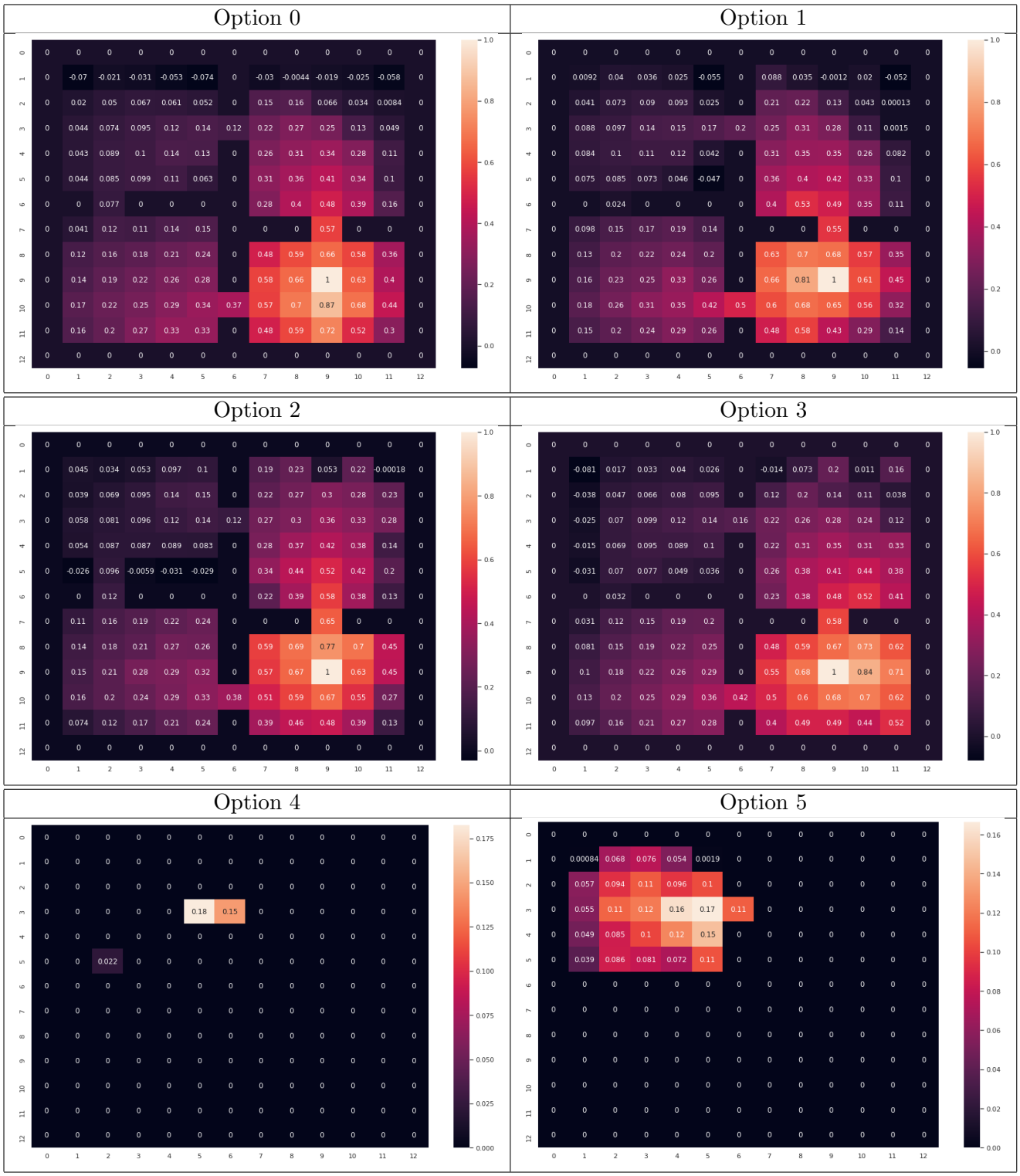
1.1 SMDP-Q Learning

1.1.1 Related Q-value plots with random start point with goal 1

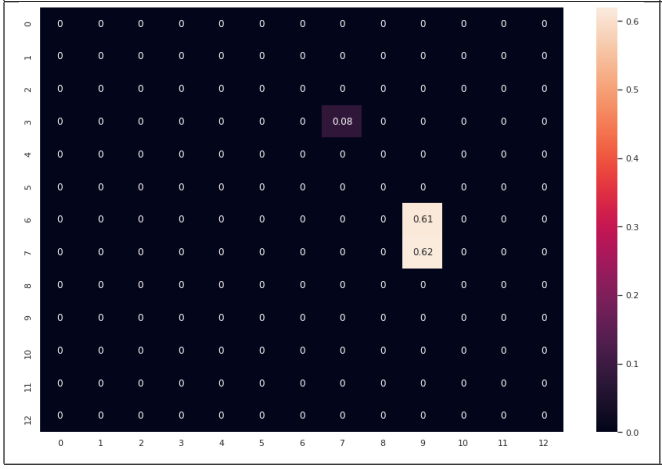



Option 6
Option 7
Option 8
Option 9
Option 10
Option 11

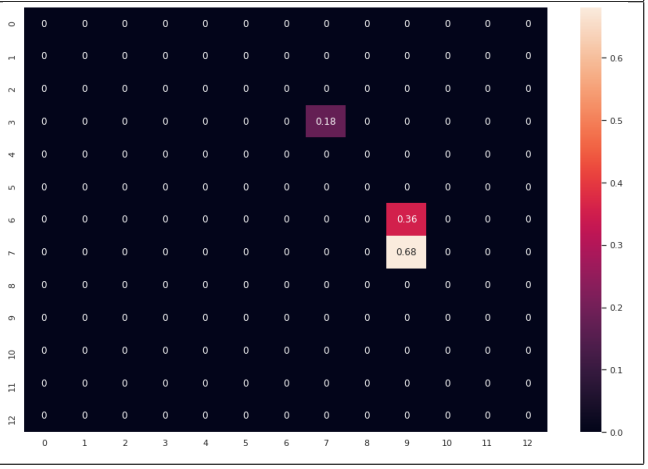
1.1.2 Related Q-value plots with random start point with goal 2



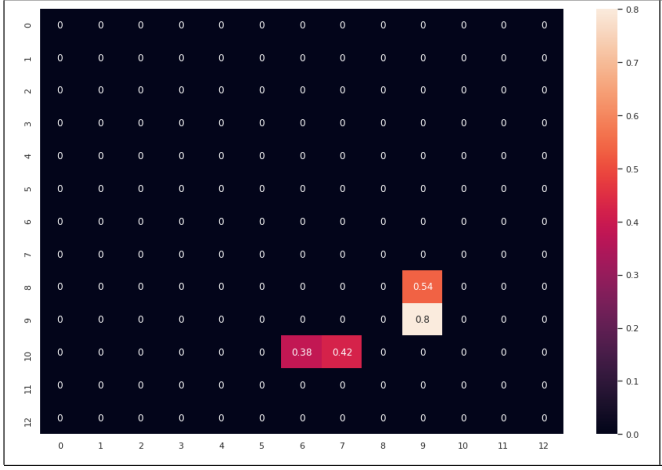
Option 6



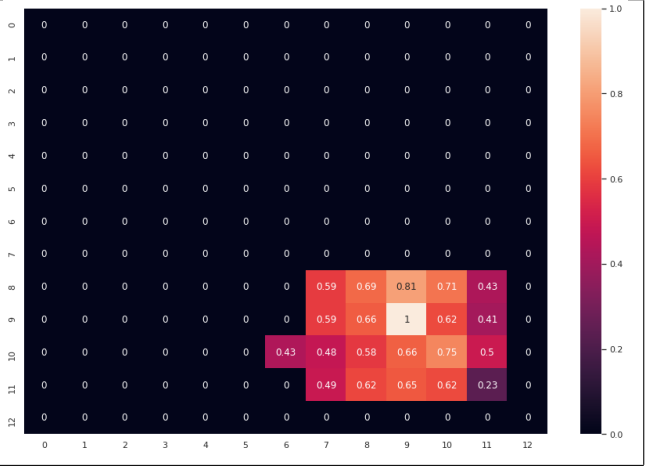
Option 7



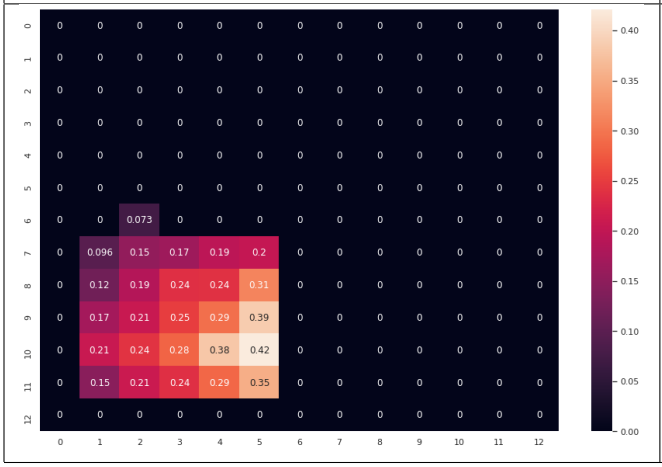
Option 8



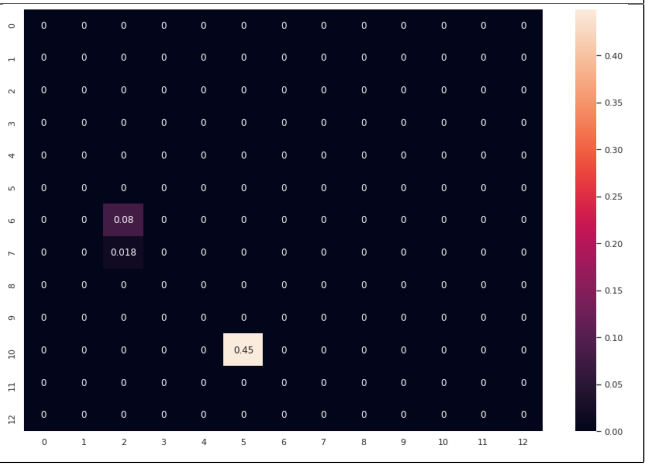
Option 9



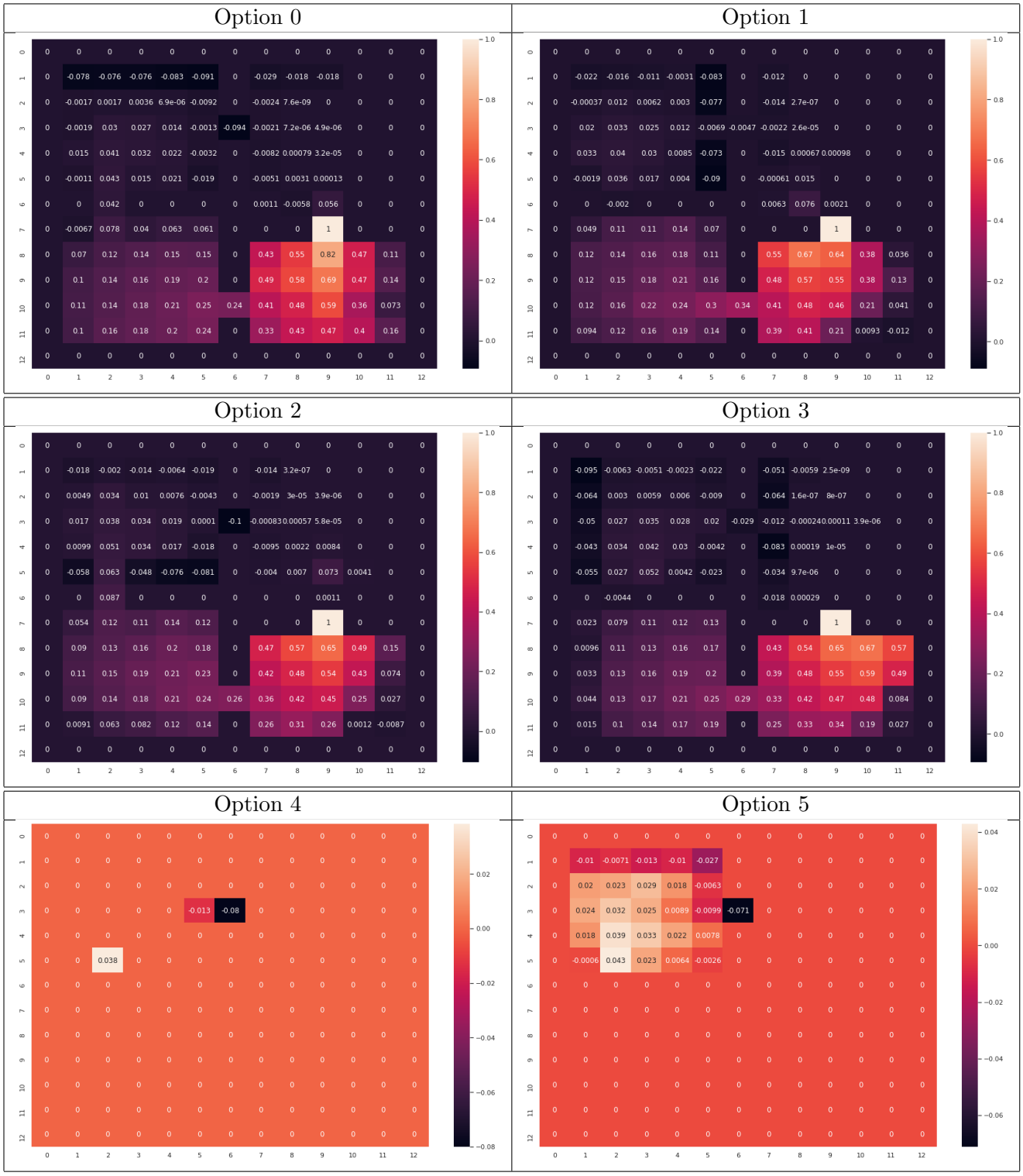
Option 10



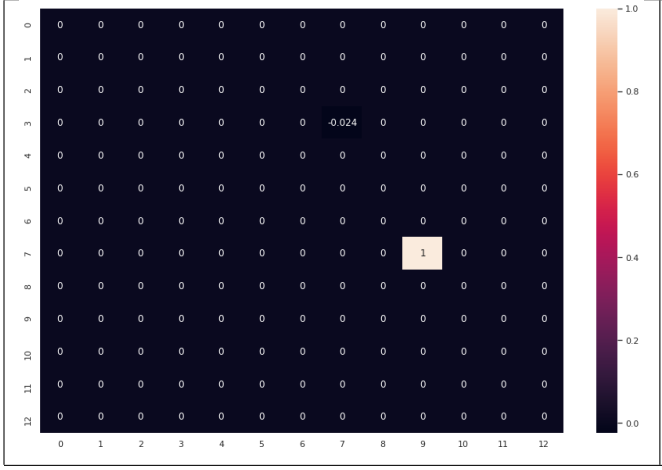
Option 11



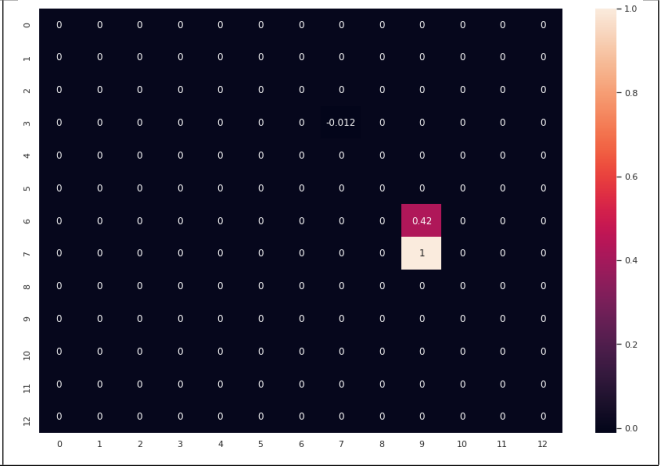
1.1.3 Related Q-value plots with fixed start point with goal 1



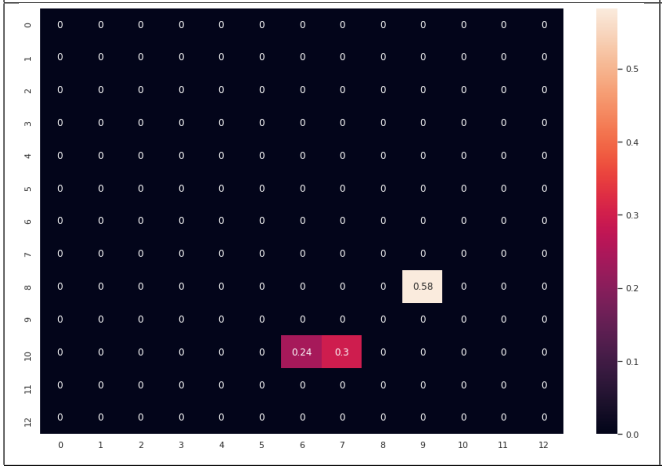
Option 6



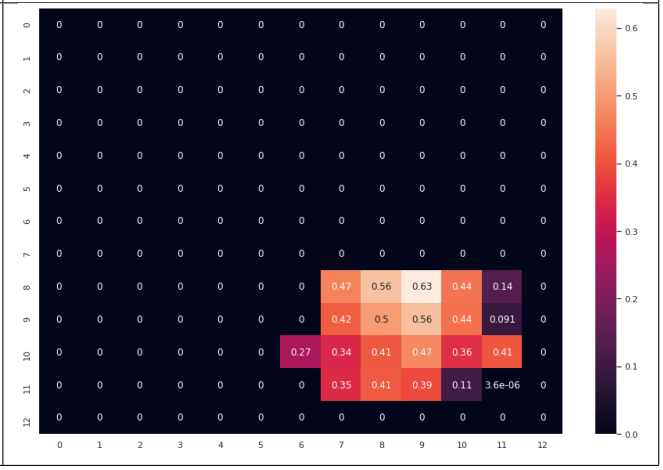
Option 7



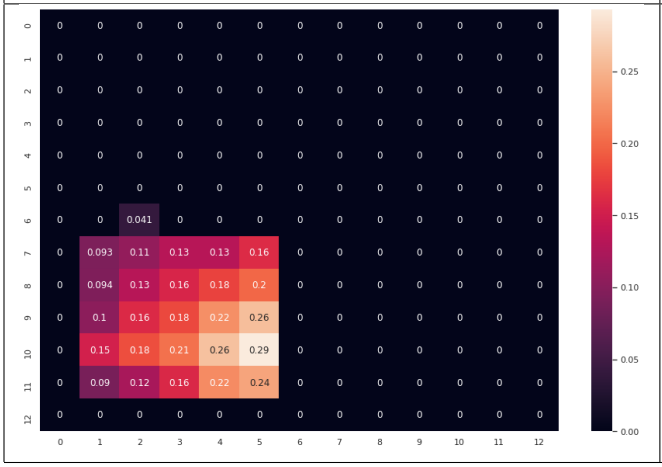
Option 8



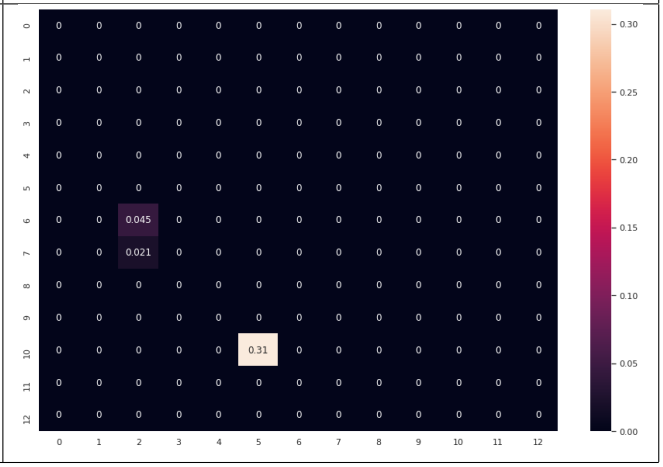
Option 9



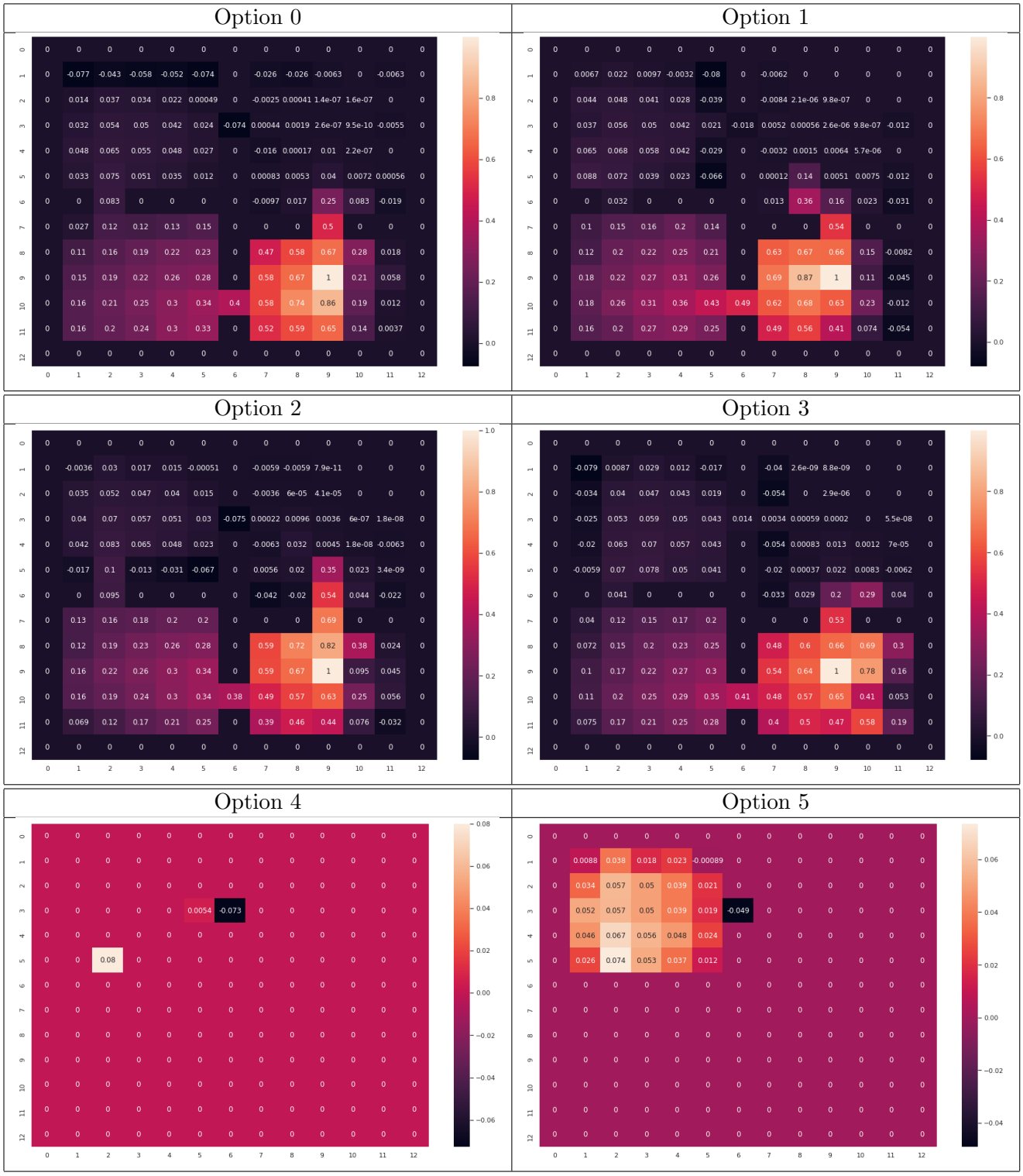
Option 10

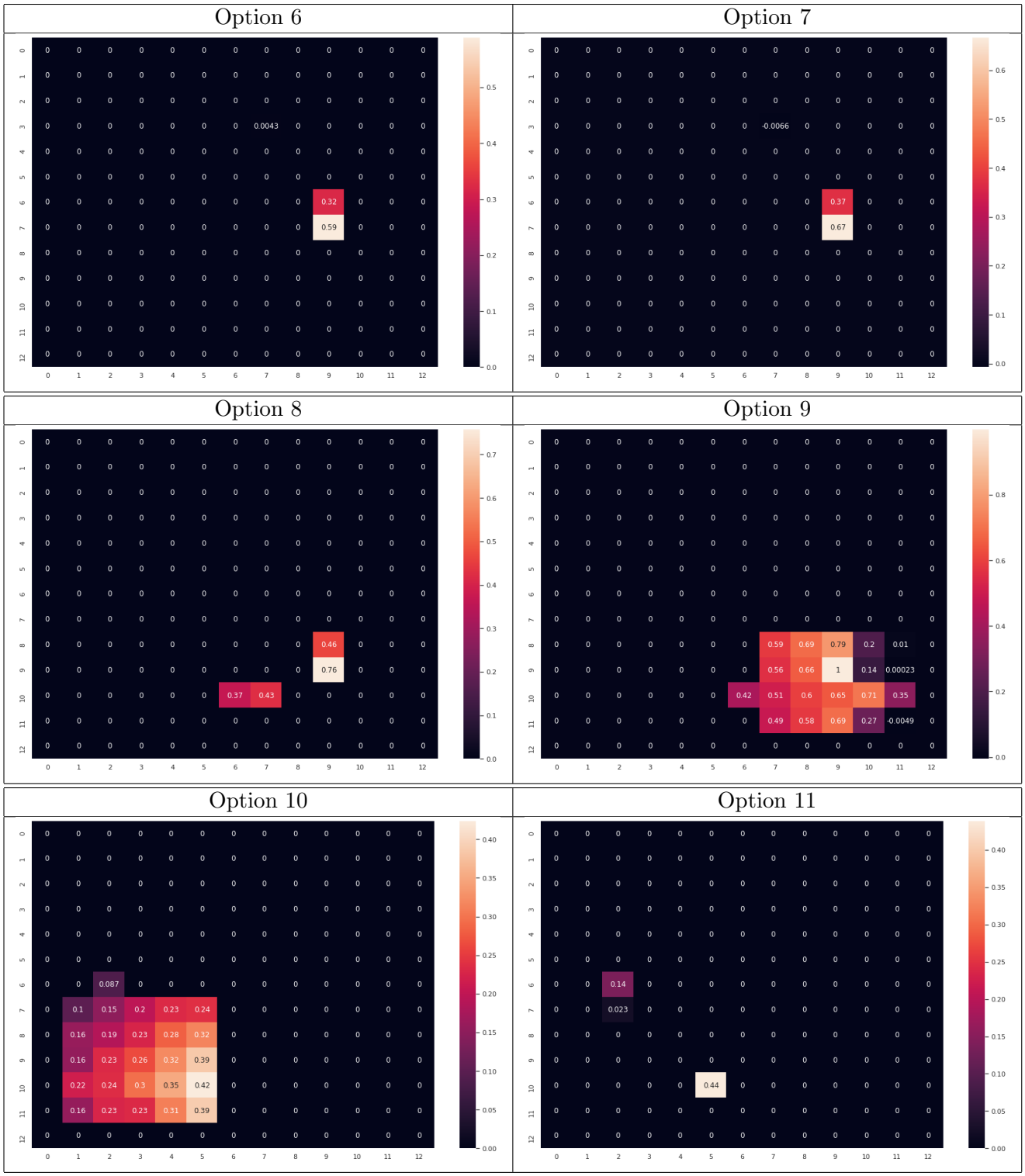


Option 11



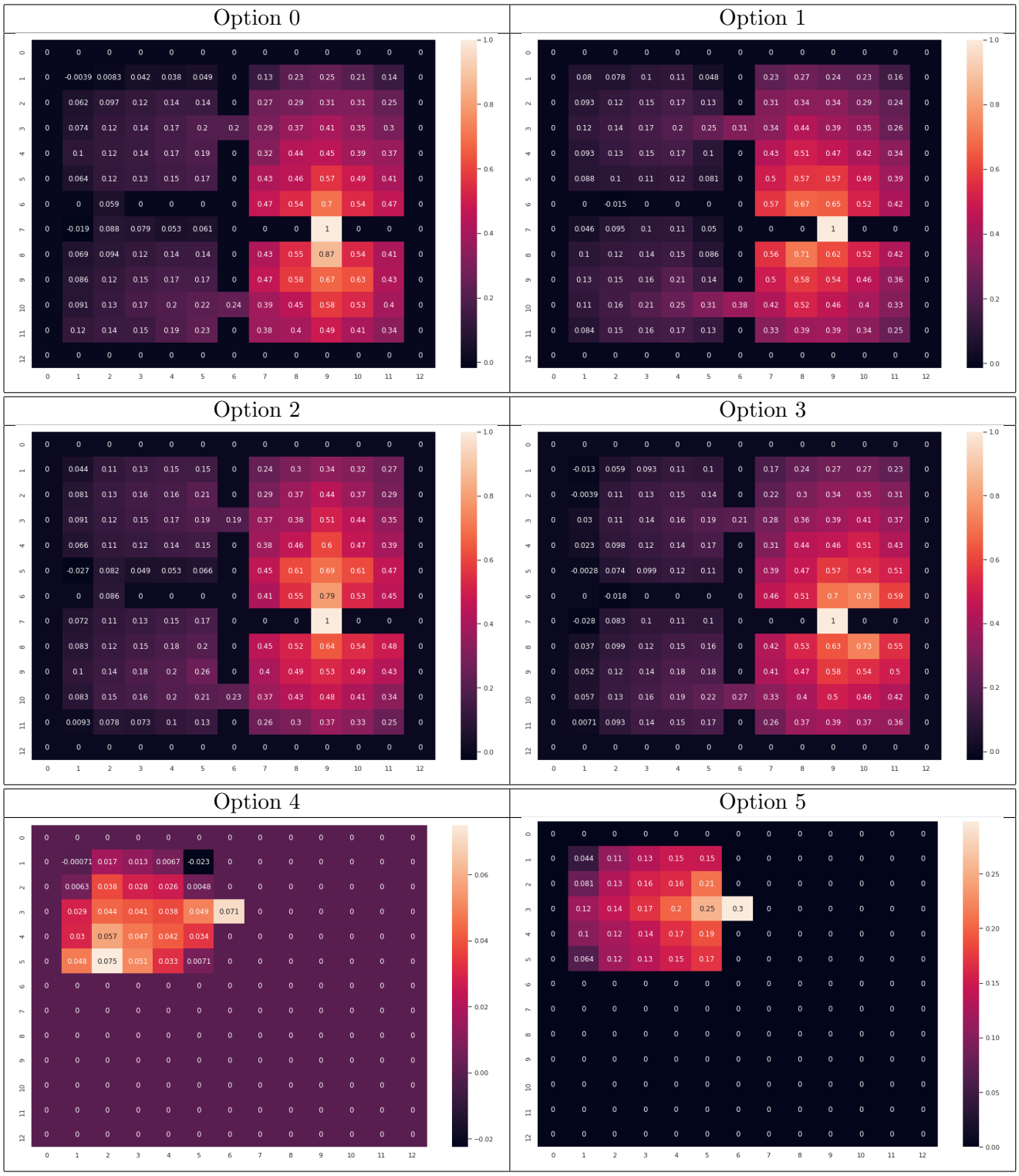
1.1.4 Related plots with fixed start point with goal 2



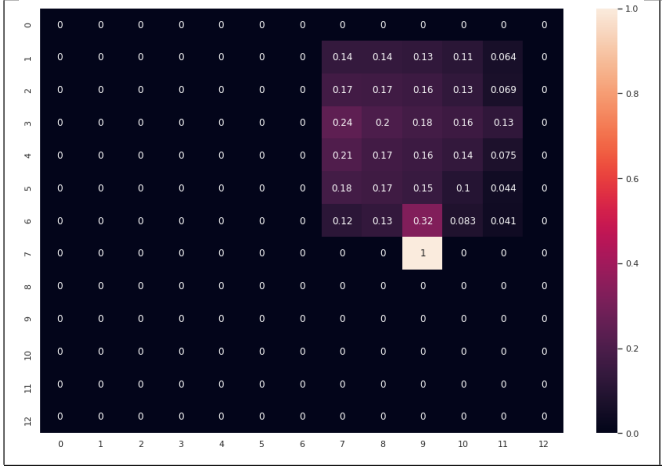

Option 6
Option 7
Option 8
Option 9
Option 10
Option 11

1.2 Intra-Option Q-Learning

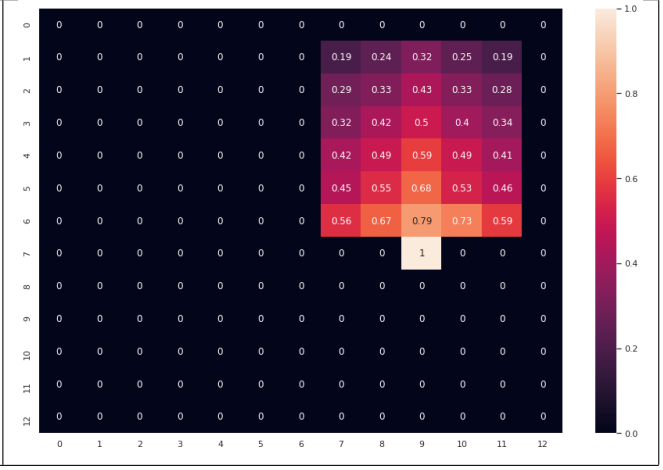
1.2.1 Related Q-value plots with random start point with goal 1



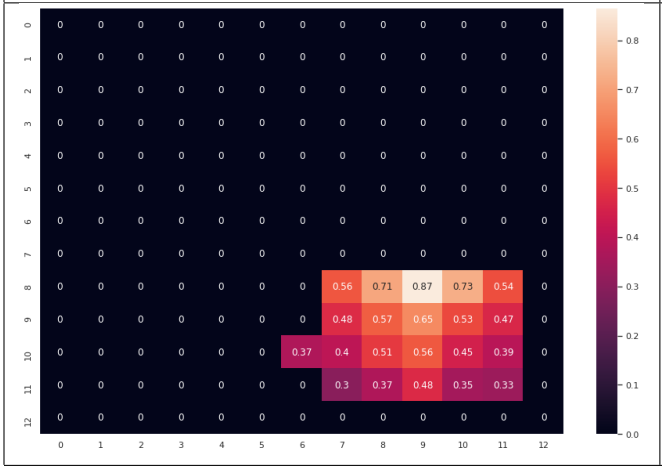
Option 6



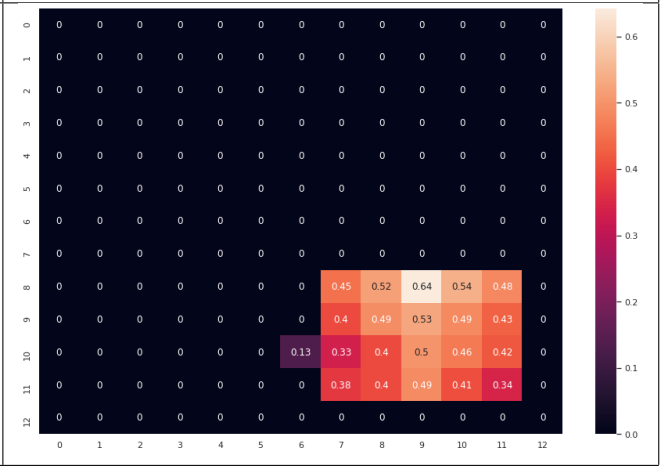
Option 7



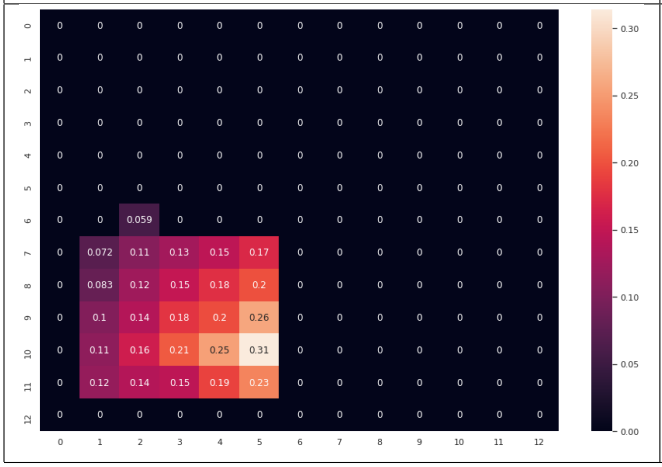
Option 8



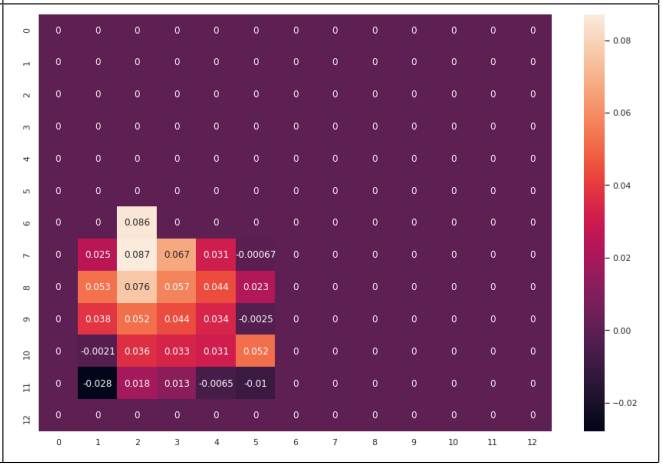
Option 9



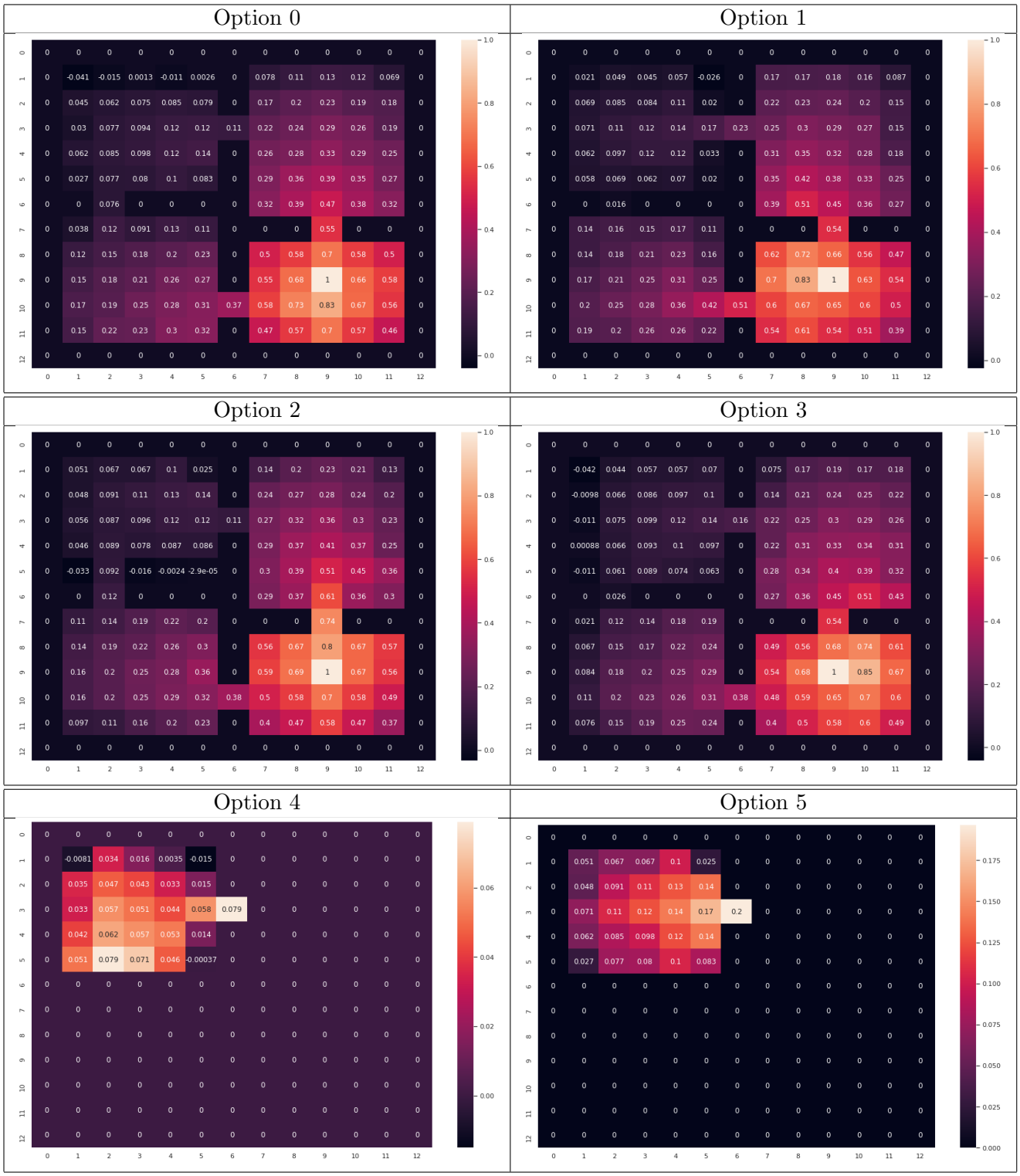
Option 10

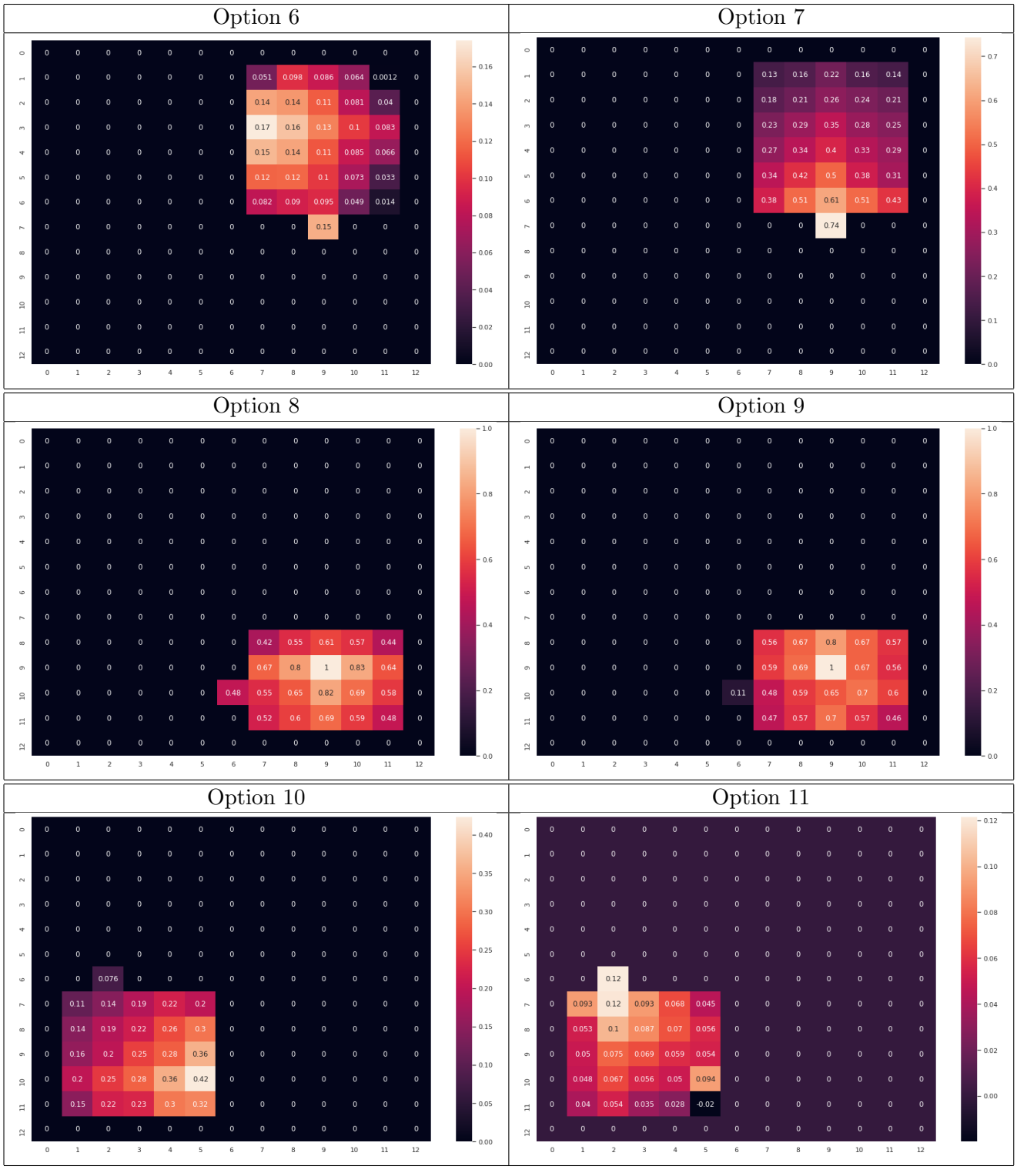


Option 11

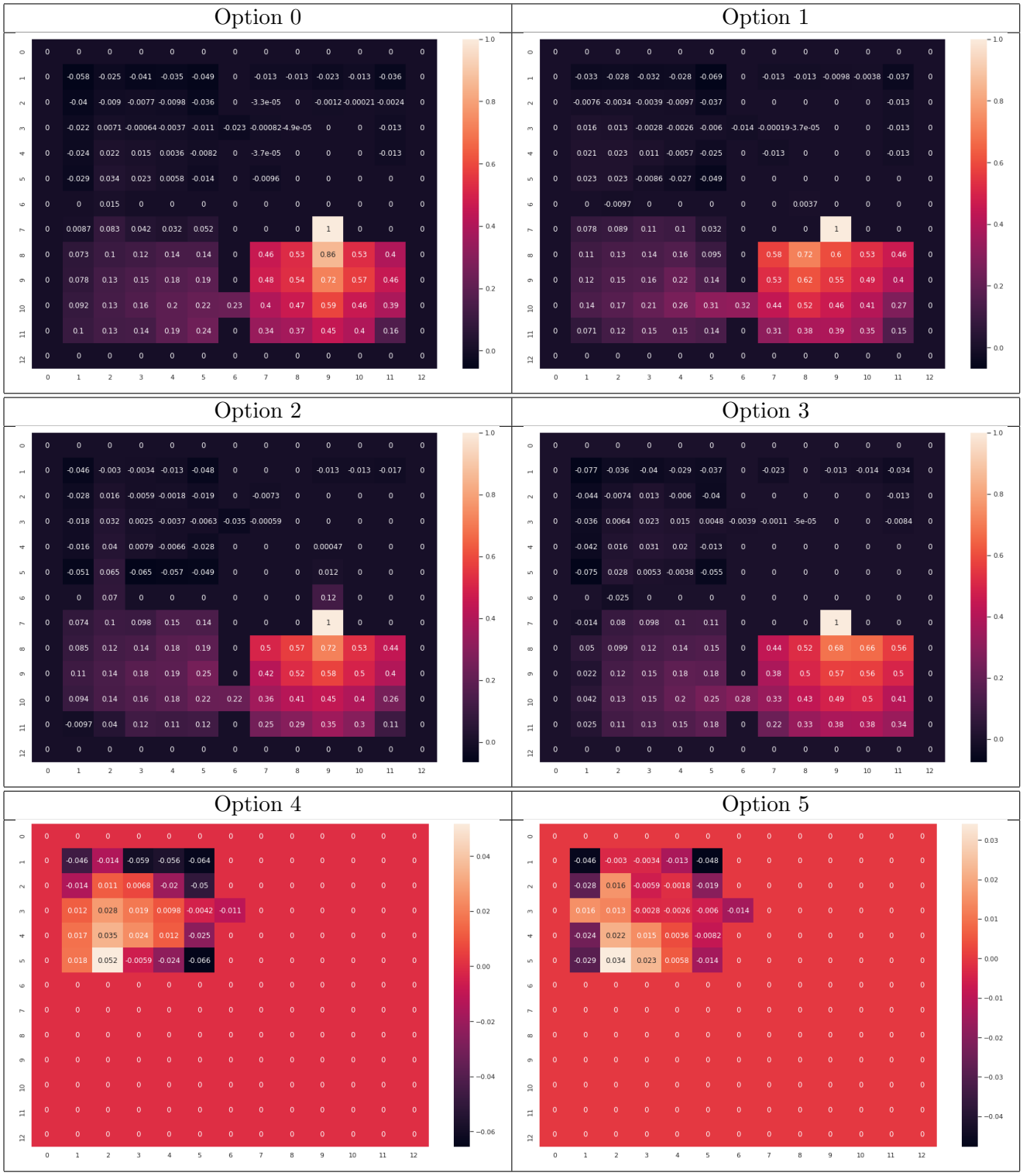


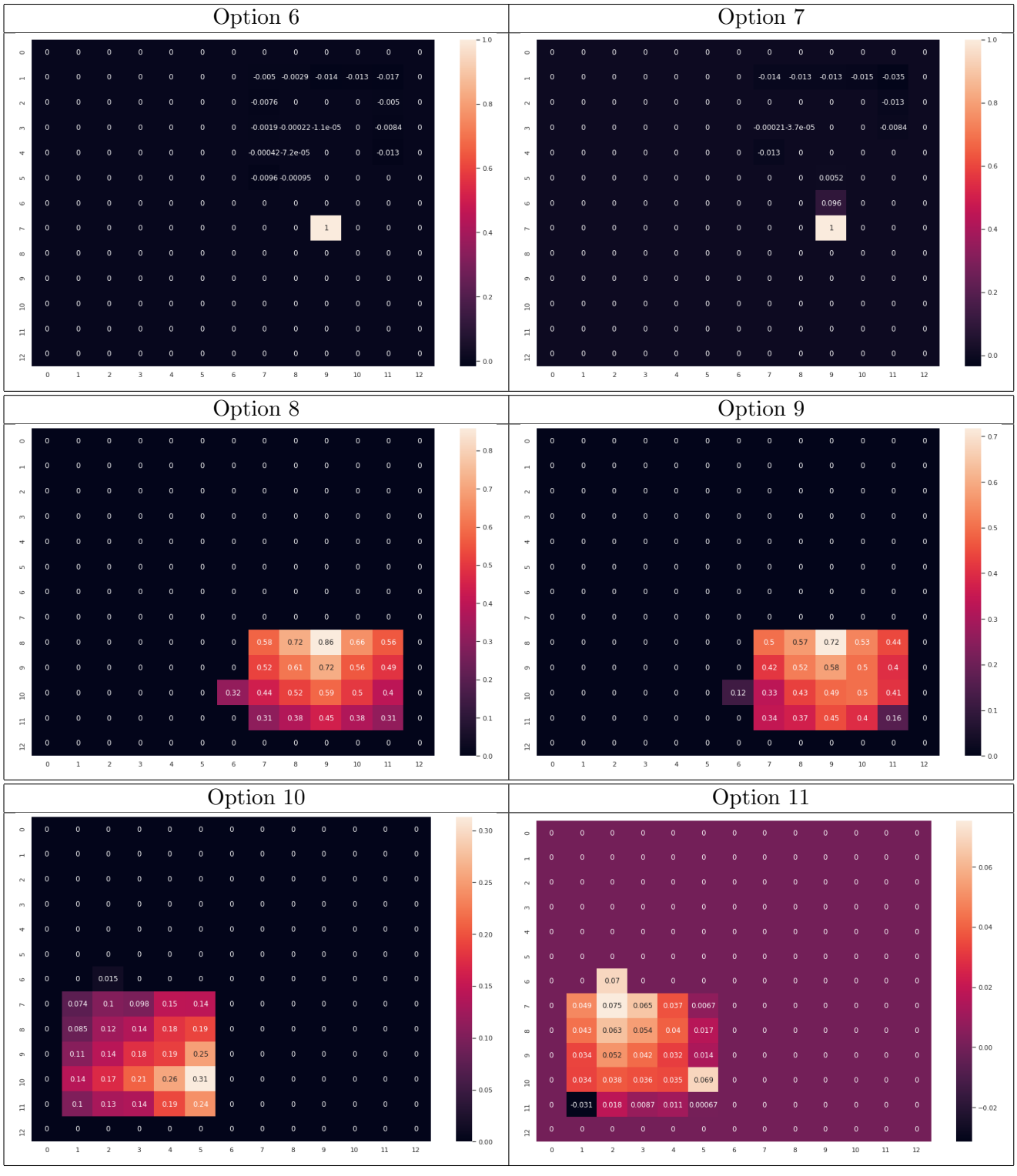
1.2.2 Related Q-value plots with random start point with goal 2



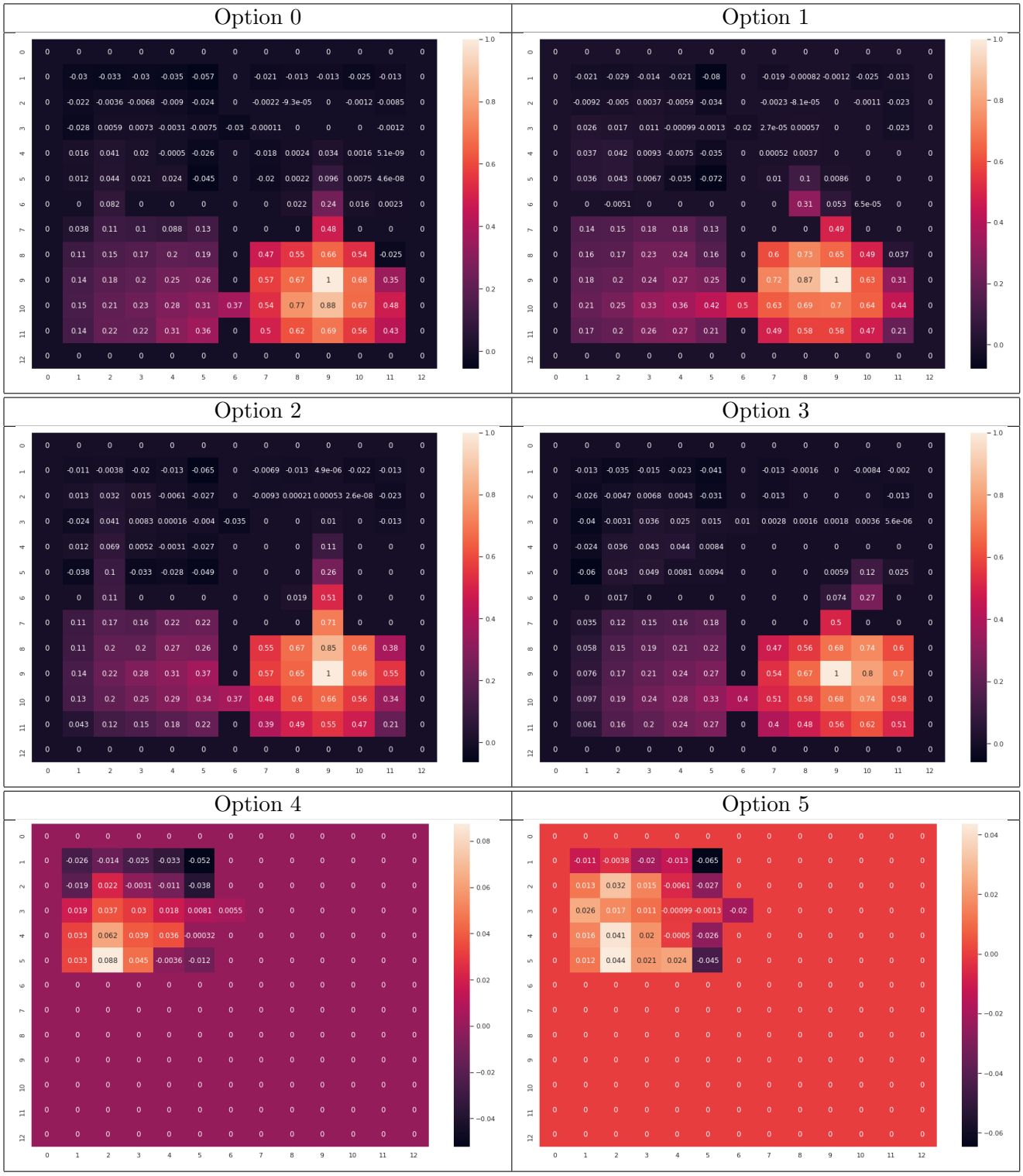

Option 6
Option 7
Option 8
Option 9
Option 10
Option 11

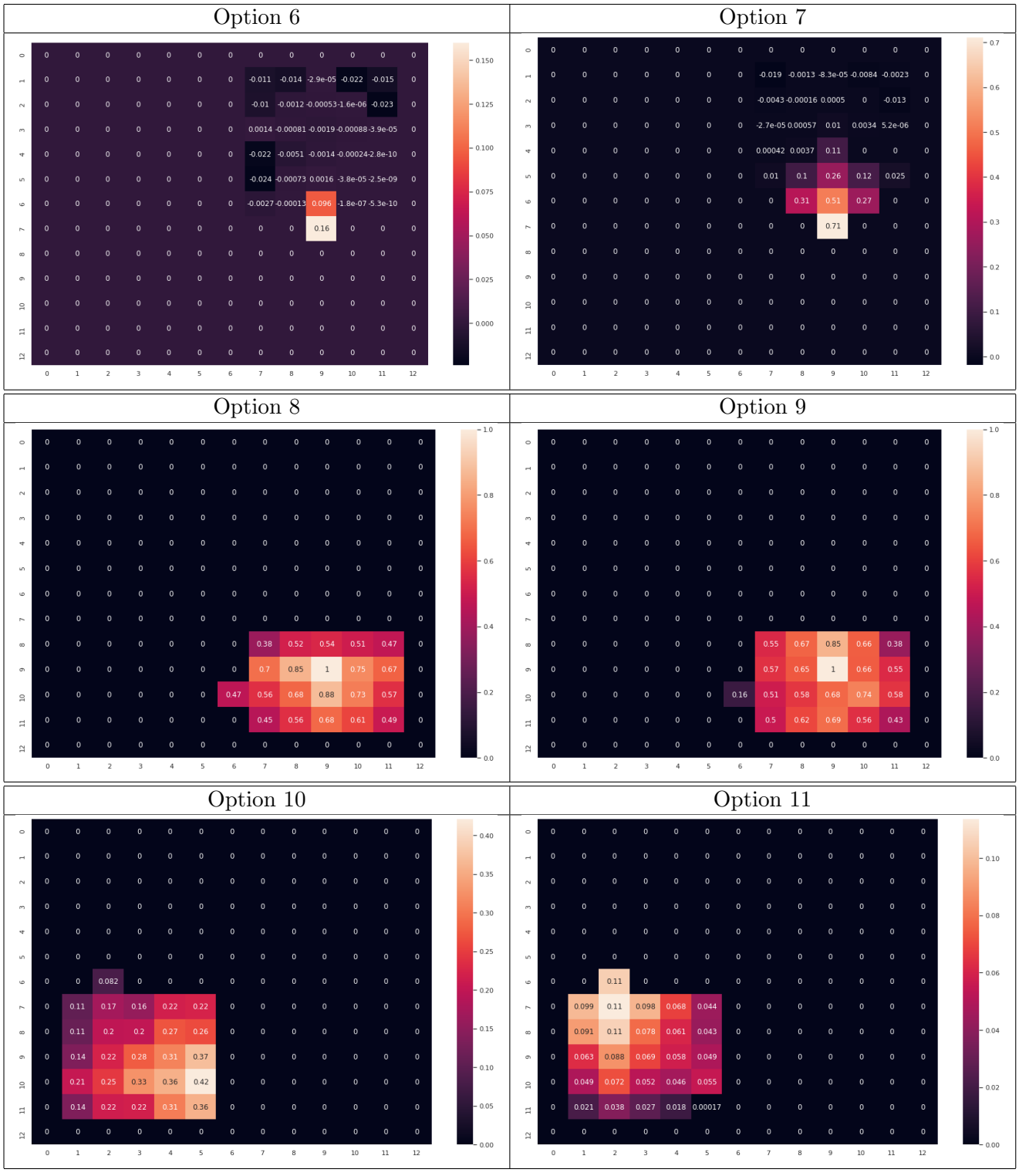
1.2.3 Related plots with fixed start point with goal 1




Option 6
Option 7
Option 8
Option 9
Option 10
Option 11

1.2.4 Related plots with fixed start point with goal 2

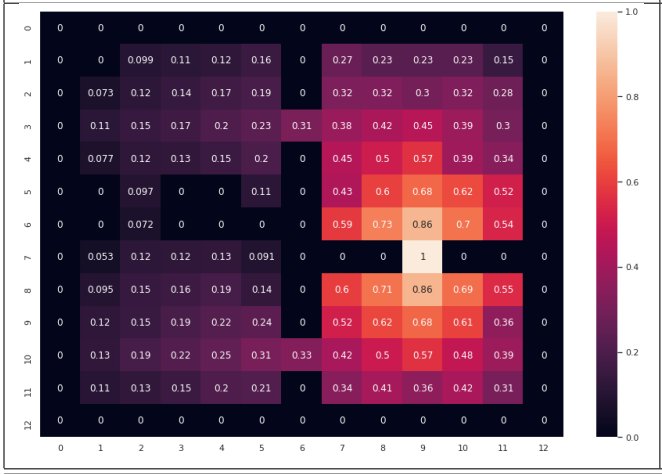



Option 6
Option 7
Option 8
Option 9
Option 10
Option 11

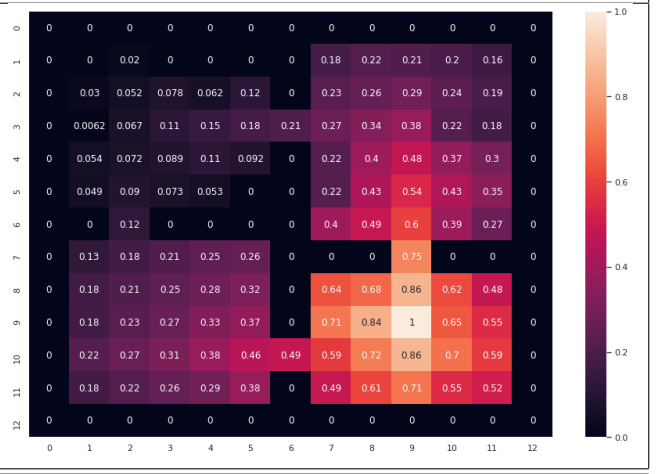
1.3 V-value plots

SMDP-Q Learning(V-values)

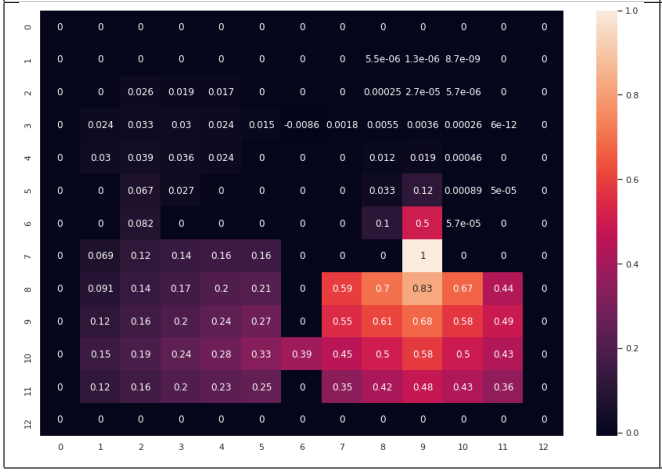
Random initial state with g1



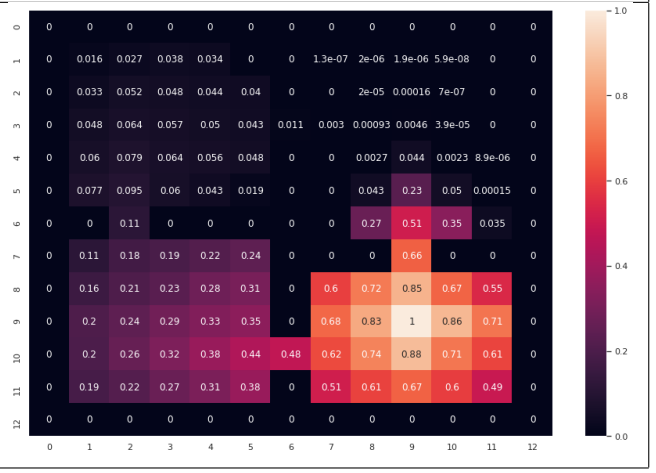
Random initial state with g2



Fixed initial state with g1

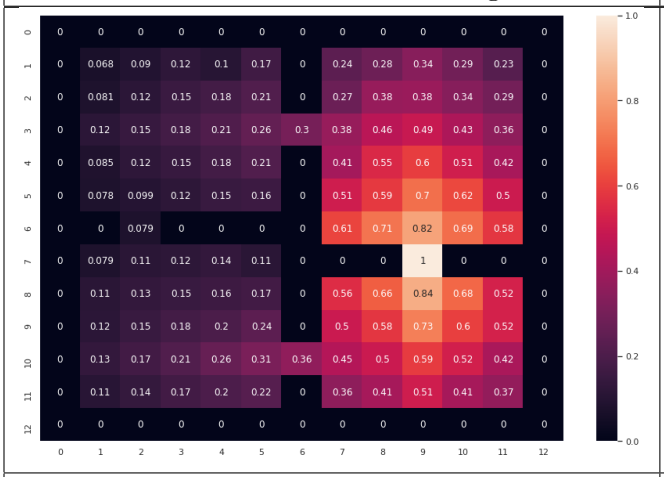


Fixed initial state with g2

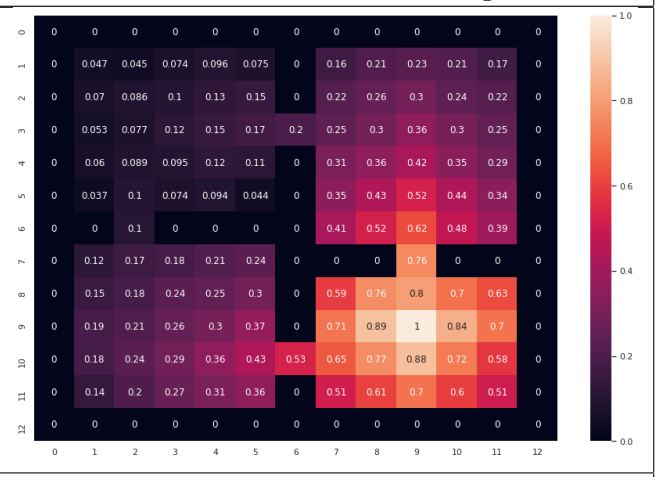


Intra-Option Q Learning(V-values)

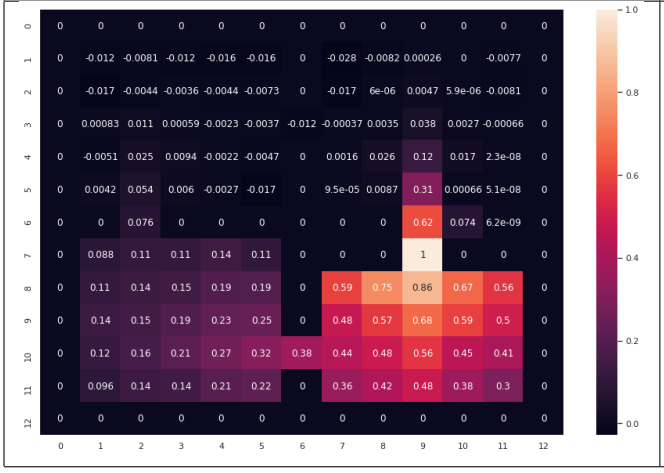
Random initial state with g1



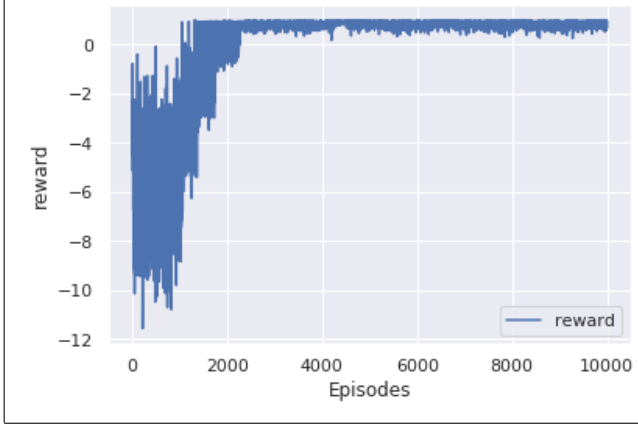
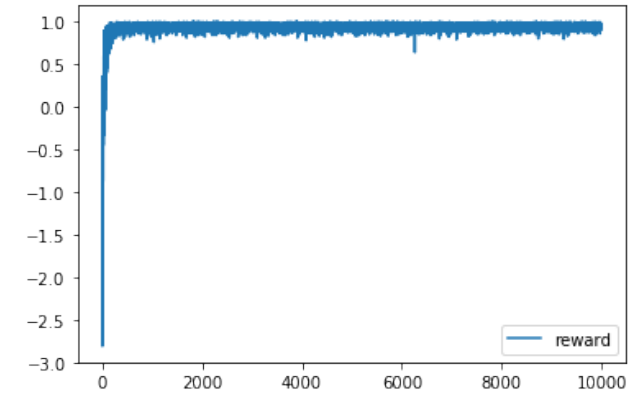
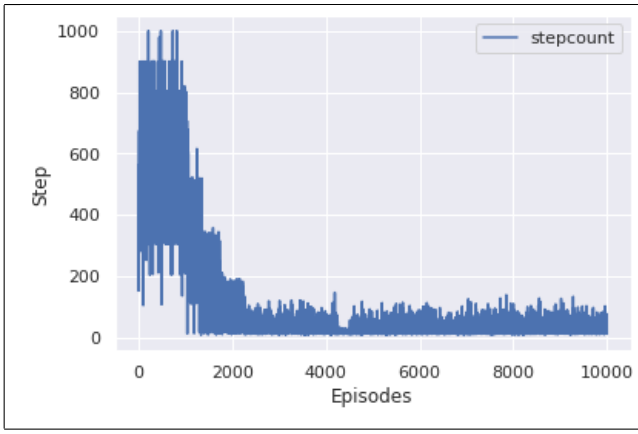
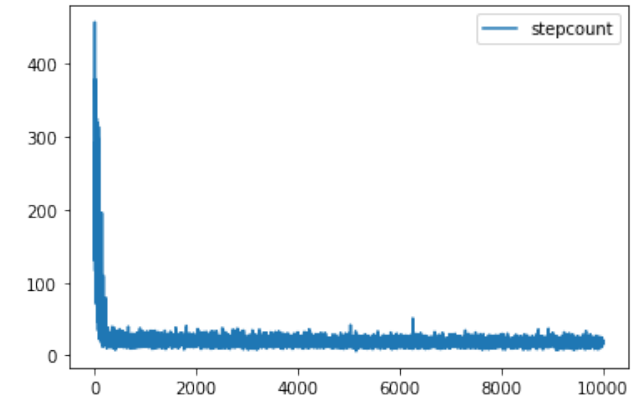
Random initial state with g2



Fixed initial state with g1



Fixed initial state with g2

Learning curve comparison)	
SMDP Q-Learning	Intra-option Q-Learning
	
	

1.4 Observation and inference:-

- We have plotted heat map for both Q-values and V-values for Smdp Q-learning and Intra-option Q-Learning for fixed start and random start.
- As we can observe from both Q-value and V-value plots,After fixing the start state, every episode start with same initial state, agent does not explored much those states which are far from initial state. This happened because, I kept epsilon value .1 so at each step it select best action with .9 an this effect like this- Suppose it reached goal some number of times, then all the states for those paths from start to end will be updated. since the path from room 4 to 3 is smaller then path $4- > 1- > 2- > 3$, ones the smaller paths states updated then the q-values on that states will be high for those option which leads to goal via path $4- > 3$. In this way if we consider a path $4- > 1- > 2- > 3$, states around initial state will have high values for the action which leads to goal via path $4- > 3$ and it is exploring with only .1 so it is highly likely that it will follow smallest path most often.
- If we compare V-values heatmap we cant find much difference in both the case fixed initial state or random initial state but if we look at Q-values there is difference. As we know that Intra-option learning in real sense uses Q-learning. So every state gets more updates.

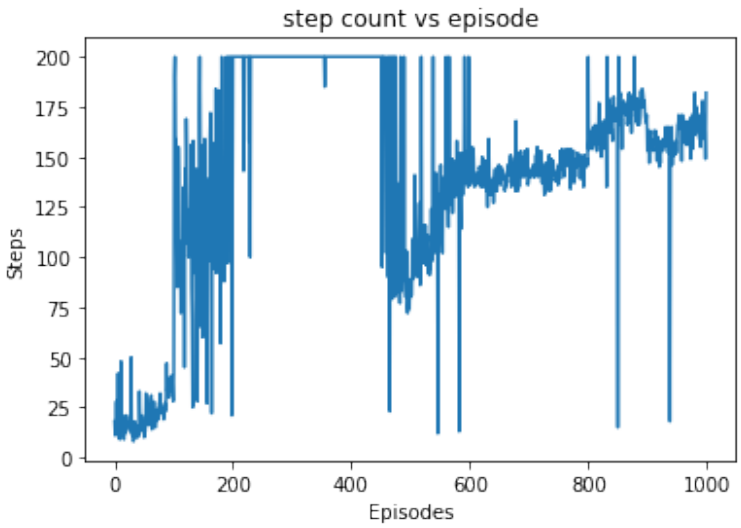
If we look at option 4,6,7,11 they are not explored much because if agent picks any option from state then only that state gets update not the states along the path of the option, In this after finding options that leads to goal in less time it select that option with .9 probability which results in other sates get less update but in case of Intra-option all options are fully explored.In case of intra option information of one transition is used to update all the relevant transition.

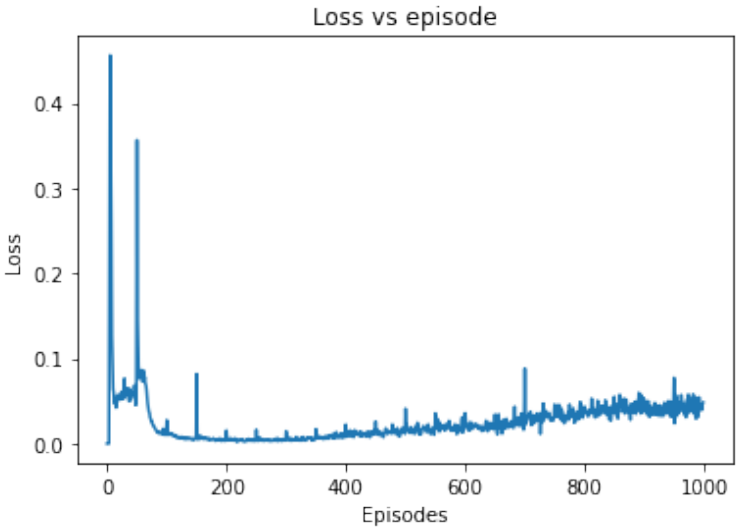
From the above learning curve we can see that, both curves approaches best values in less number of episodes. Because in each episode we are generating a lot of data and updating a lot number of state-action pair q values.

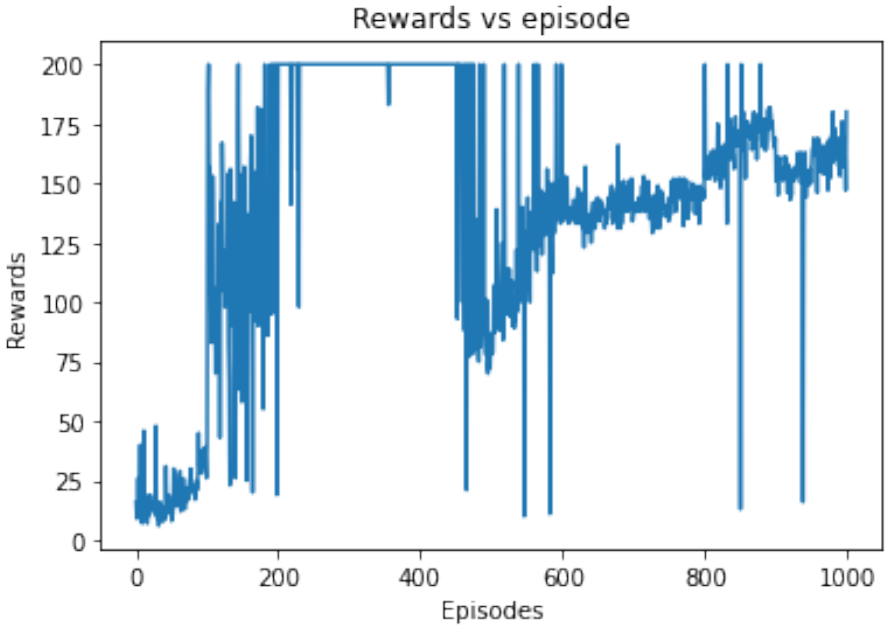
2 DQN

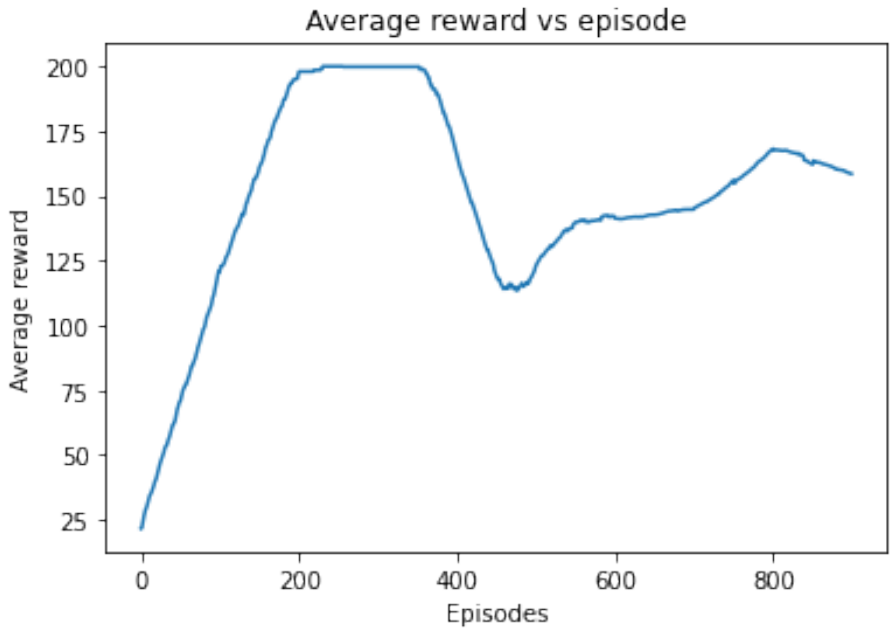
2.1 Implementation of DQN and related plots

2.1.1 Related plots




Loss vs episode
Loss
0.0
0.1
0.2
0.3
0.4
0
200
400
600
800
1000
Episodes


Rewards vs episode
Rewards
0
25
50
75
100
125
150
175
200
0
200
400
600
800
1000
Episodes

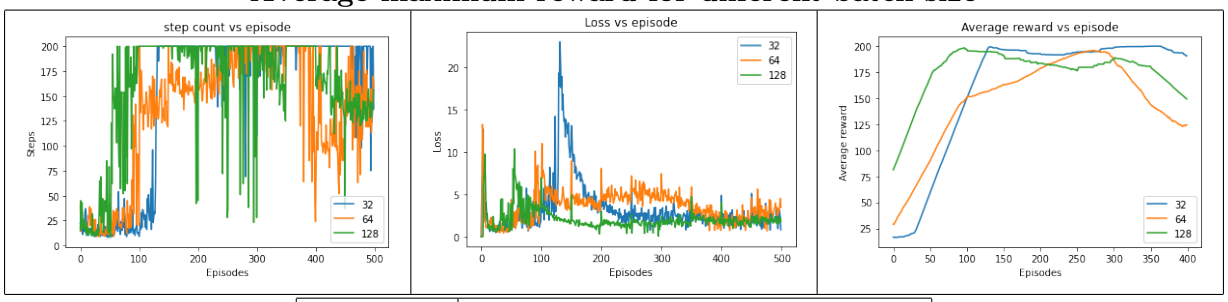


2.1.2 Best hyperparameters

Parameters list	
Parameter Name	Parameter value
Number of episodes	1000
Epsilon start value	1
Epsilon end value	.05
Decay rate	500
Gamma	.99
Learning rate	1e-4
Replay memory size	100000
Target network update	50(episodes)

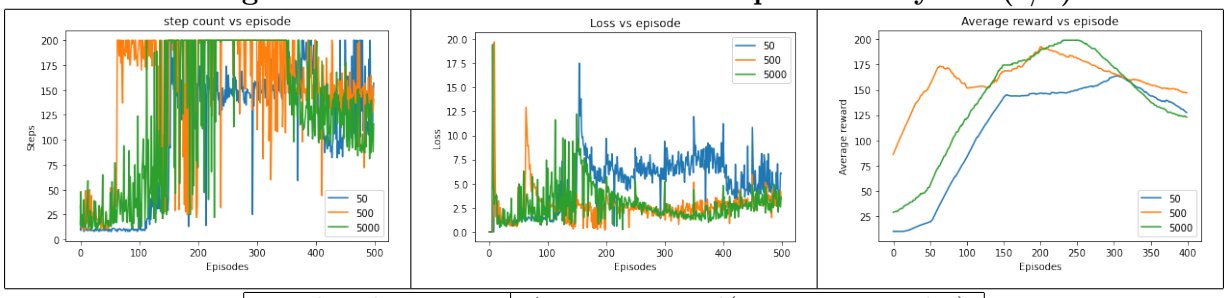
2.1.3 Observations and inference

Average maximum reward for different batch size



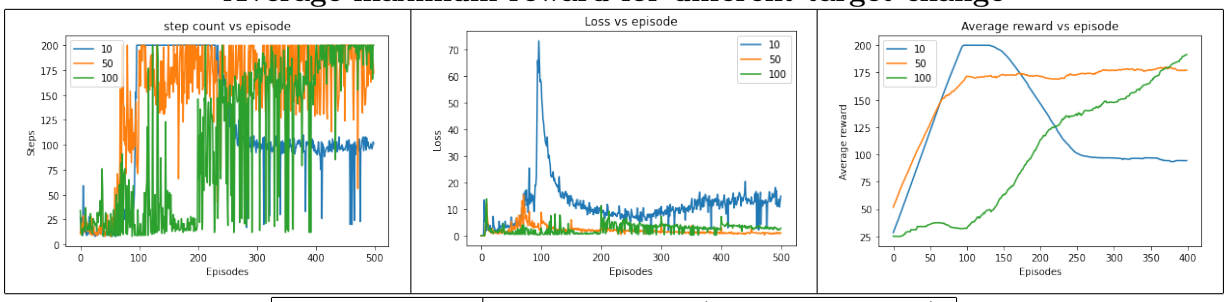
Batch size	Average reward(over 100 episodes)
32	195.84
64	198.67
128	196.2

Average maximum reward for different epsilon decay rate($1/x$)



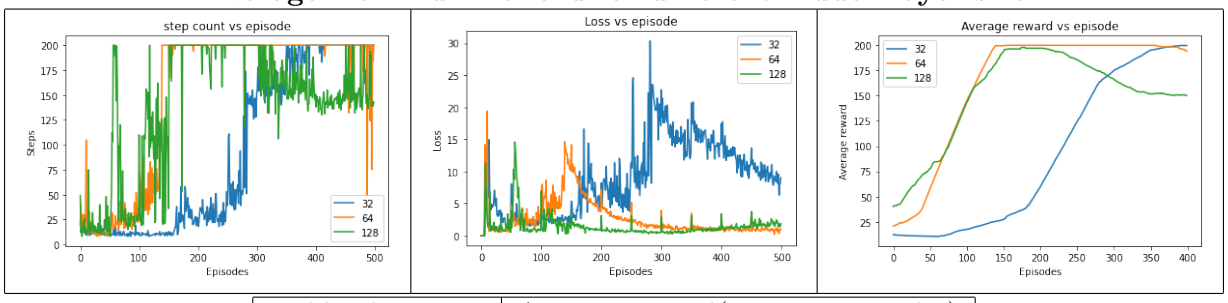
Epsilon decay rate	Average reward(over 100 episodes)
x = 50	163.84
x= 500	192.67
x = 5000	199.2

Average maximum reward for different target change



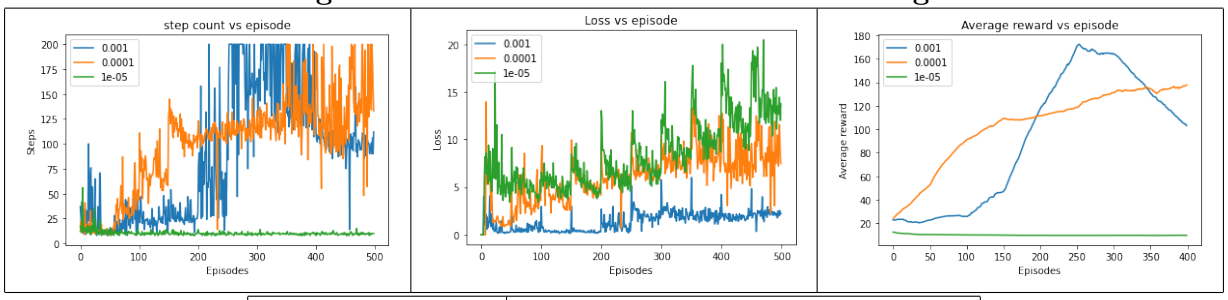
Target change	Average reward(over 100 episodes)
10	200.
50	180.67
100	191.2

Average maximum reward for different hidden layer size



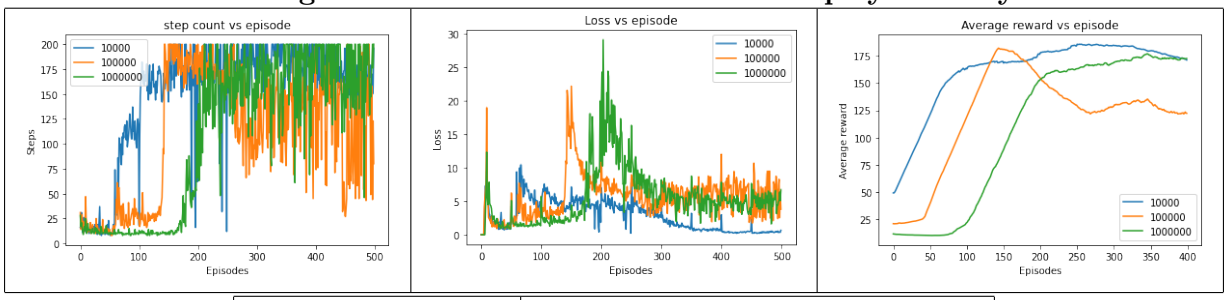
Hidden layer size	Average reward(over 100 episodes)
32	199.
64	1990.67
128	197.2

Average maximum reward for different learning rate



Learning rate size	Average reward(over 100 episodes)
1e-3	172.
1e-4	137.67
1e-5	12.2

Average maximum reward for different replay memory

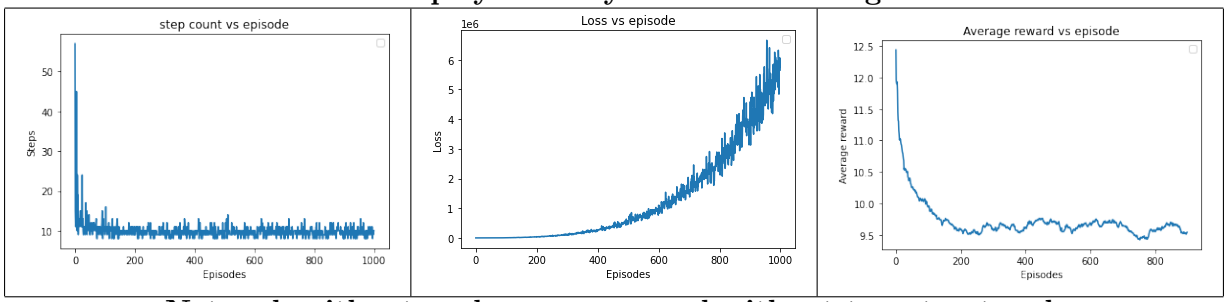


Replay memory size	Average reward(over 100 episodes)
10000	185.
100000	181.67
1000000	176.2

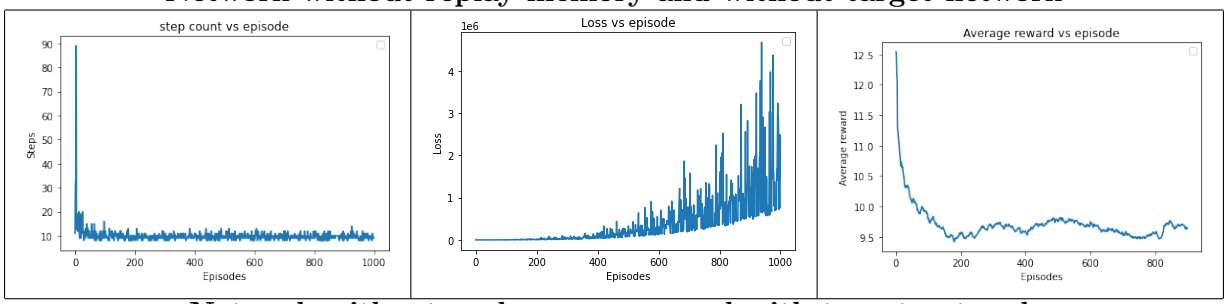
- **Batch size:-** As we can see from average reward graph maximum reward over 100 episodes in all cases are approximately same. It is evident from the graph as we increase batch size learning becomes faster because state are getting more updates.
- **Epsilon decay:-** As we can see from average reward graph maximum reward over 100 episodes, as the decay rate decreasing maximum average reward keep increasing. As the rate decreases agent gets more chance to explore and in results average reward increased. It is also visible in step graph where green shows more variance as it is exploring more.
- **Target change:-** As we can see if the target update frequency is high then it attends maximum average reward fast and after that again it decreases, but in case of low frequency maximum average reward increase slowly and it show stable learning.
- **Hidden layer size:-** As we can see all are achieving maximum average reward. It can be seen that as the hidden layer size increases leaning becomes fast but after some size further increase does not increase efficiency of learning.We can choose the hidden layer depending on the complexity of state and action space.Here er have choose 3 layers with size 128 each.
- **Learning rate:-** As it can be seen that too small and not too big learning rate is not good. SO for choosing good learning rate we have to do multiple experiment.
- **Replay memory size:-** As the memory size increase it can keep more number of transitions it means it can recall more old information which leads to more stable update and it is also visible from the graph. Although network with less memory size learns faster but it is not stable as the more information comes it shows more variation.
- We see spikes in the loss because target suddenly updated by new parameter which cause sudden increase in loss.

2.1.4 playing with Replay memory and Target network

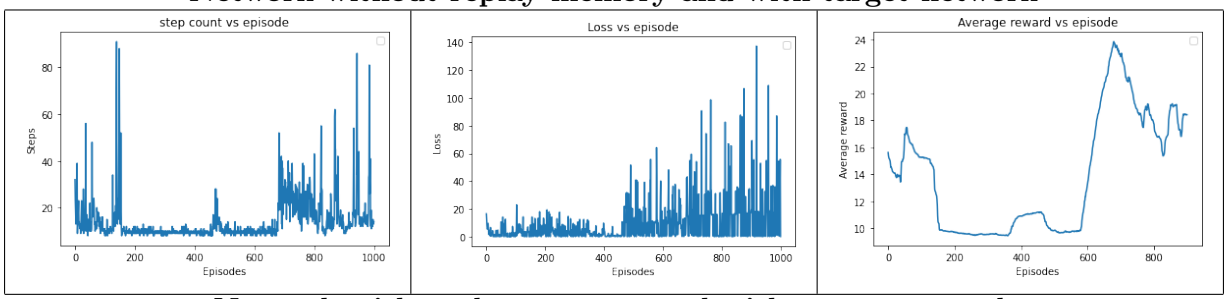
Network with replay memory and without target network



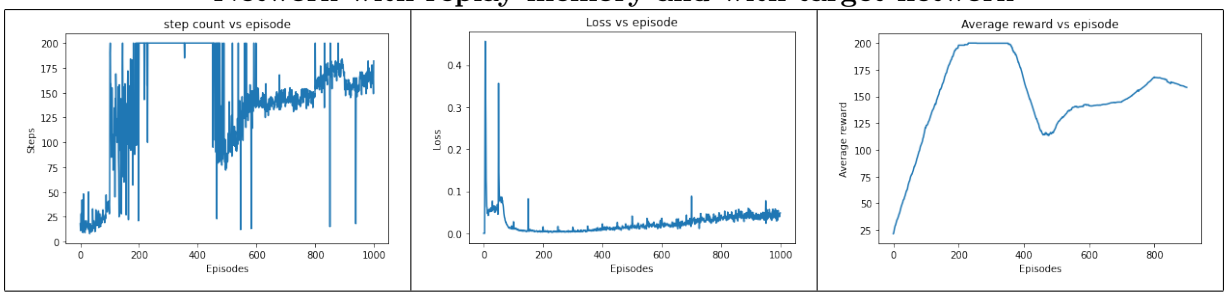
Network without replay memory and without target network



Network without replay memory and with target network



Network with replay memory and with target network



- **Network with replay memory and without target network:-** As we can see from Loss vs episode graph, as there is no target network loss keeps on increasing and average reward and steps are much less. It is happening because as we have more data in replay memory which means the diversity of state space increases. but when we train it with batch we update the parameter to minimise the current loss and that parameter updates the target network also,which means target network move towards current batch. Next time when we sample new batch they produce more loss.

- **Network without replay memory and with target network:-**Since there is no memory replay, it means biased sampling of the input space , correlation between the sets of inputs.Without replay memory there will be stochastic gradient descent. which means that loss curve will be more fluctuating and it is also visible from loss curve and effect of having no target network is same as above.
- **Network without replay memory and with target network:-** Here we target network. Since there is no memory replay, it means biased sampling of the input space , correlation between the sets of inputs. And the loss is increasing because we are keep updating the current network but the target network have parameters trained by previous episodes and the current episode have the path very different from previous ones. Basically we are using the parameters of some network which was biased trained, to get the target of current transition, it might happen that similar states might not have been seen by network earlier not able to generalize well which results in big loss.
- **Network with replay memory and with target network:-** As both are here so variation in loss is less because of replay memory and loss is not exponentially increasing because of target network.