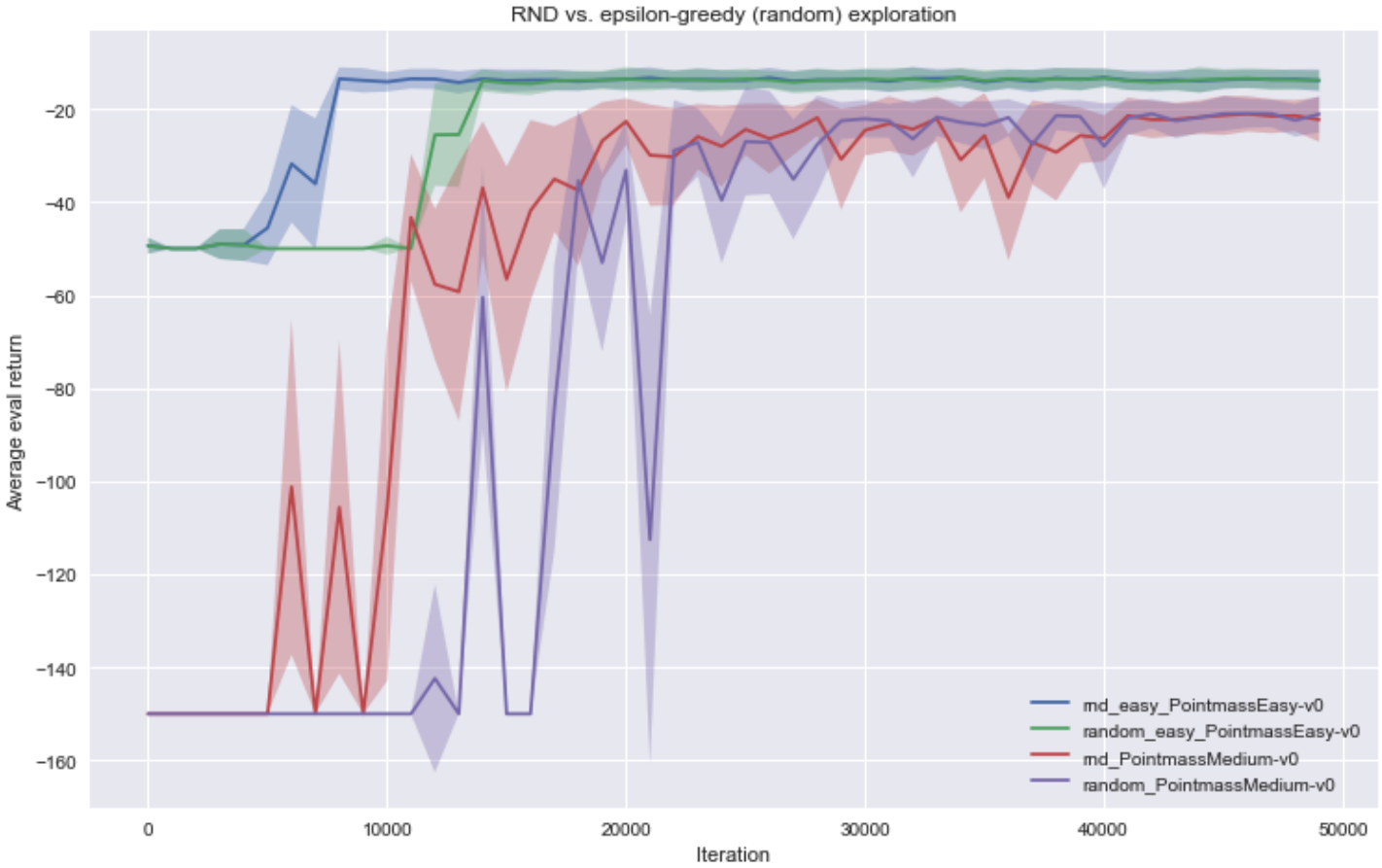
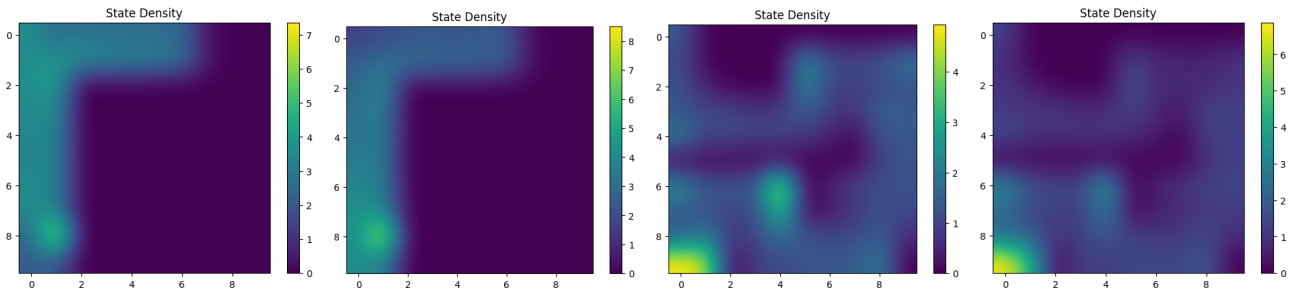
CS 285 Homework 5: Exploration and Offline Reinforcement Learning
November 21, 2021
Gautam Mittal
gbm@berkeley.edu

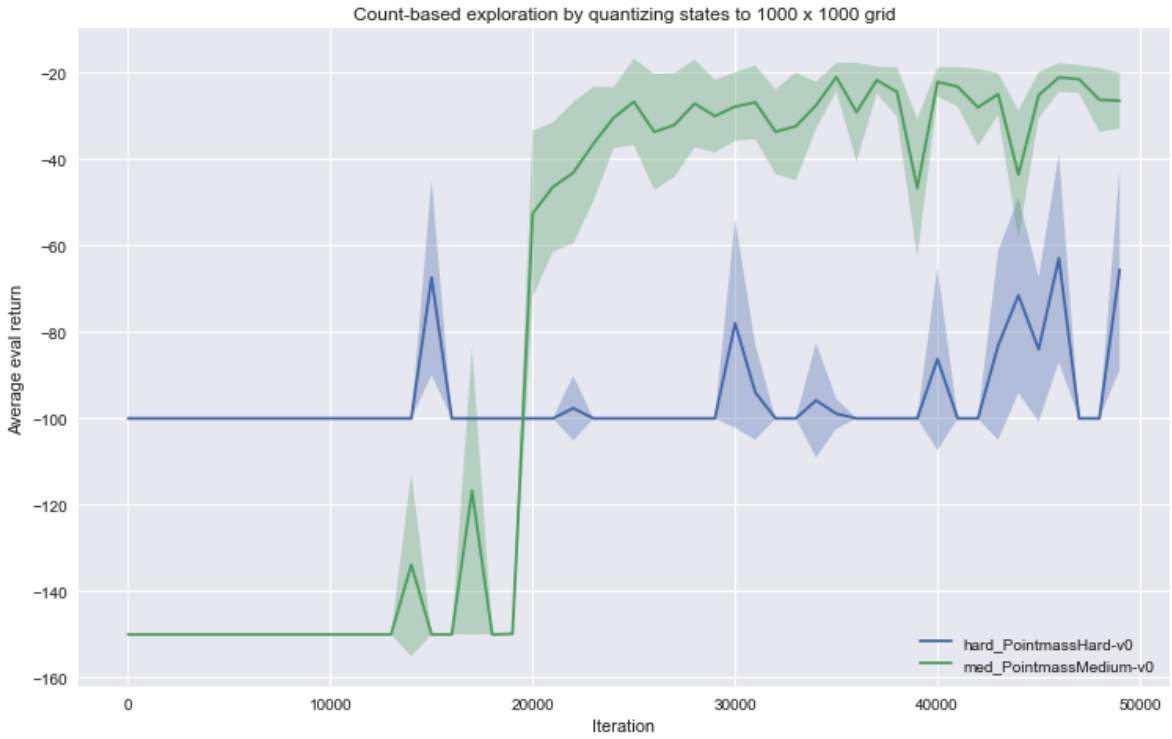
1. RND and Exploration Performance



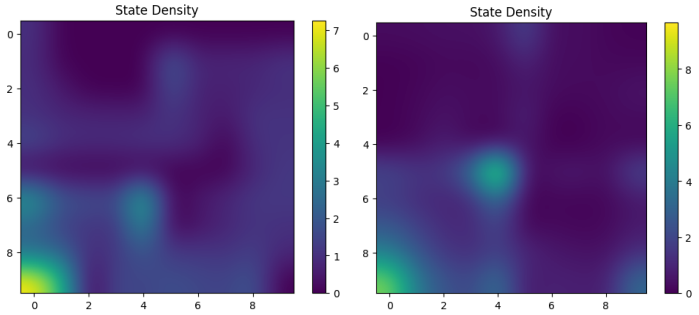
RND vs. epsilon greedy: As shown from the state density heat maps below, the RND strategy has a much more uniform coverage the environment than in the epsilon-greedy case where most of the density is around the initial starting state. This increased coverage of states as a result of the exploration is reflected in the performance curves above, as the RND strategy allows the agent to more quickly learn how to reach the goal with much lower variance.



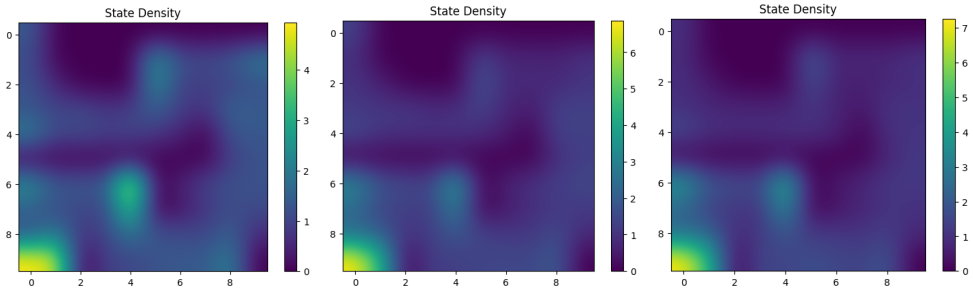
Easy RND **Easy Random** **Medium RND** **Medium Random**



Count-based exploration: We implemented a count-based approach where each continuous state is quantized to a discrete point on a 1000 x 1000 grid and we determine counts in terms of this discretized grid. To generate the exploration bonus, we use UCB (sqrt(2 * log n / N(s)) given the current state's count on the discretized grid. The approach yielded similar results to epsilon-greedy exploration, both in terms of average evaluation return on the medium environment and in terms of the state density. This is likely because even with the discretization to a 1000x1000 grid, many of the states will remain at a count of zero, resulting in a fairly uniform exploration bonus distribution which will yield information similar to an epsilon-greedy strategy. This could possibly be improved using pseudo-counts (a generative model instead of quantization) and/or quantizing to a smaller grid (e.g. 20x20 or 100x100).

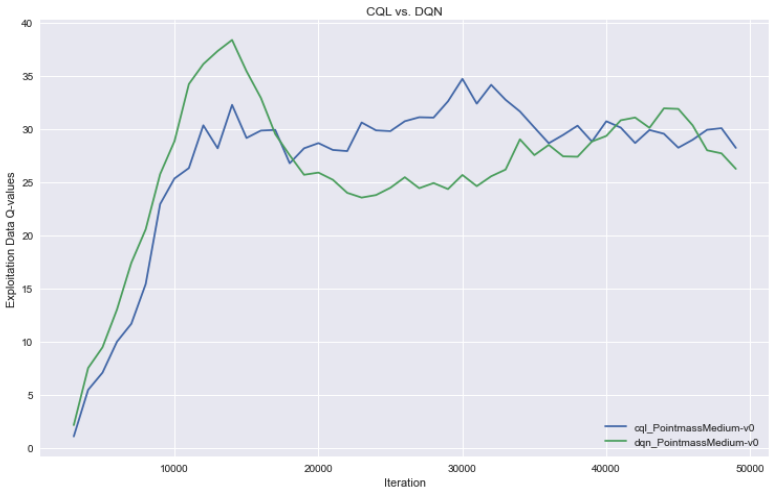


Visualization of count-based exploration method on medium (left) and hard (right) environments.

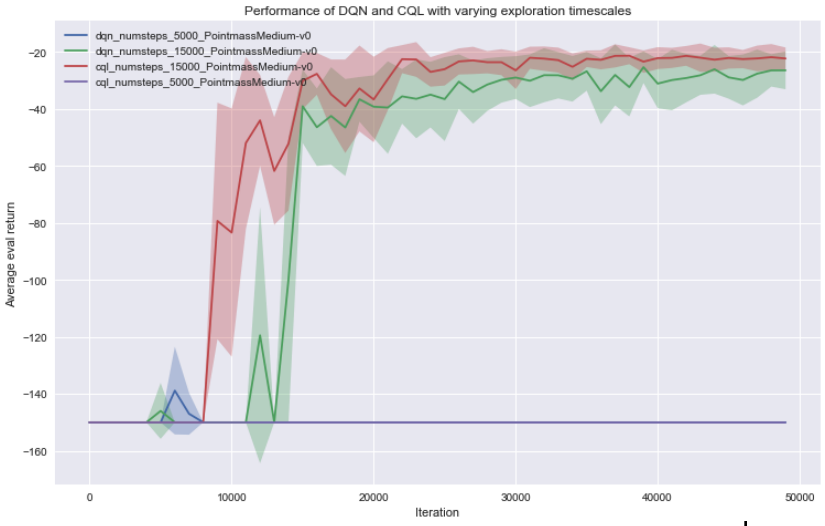


RND (left), random (middle), count-based (right) heatmaps on medium environment.

2. Offline learning on exploration data

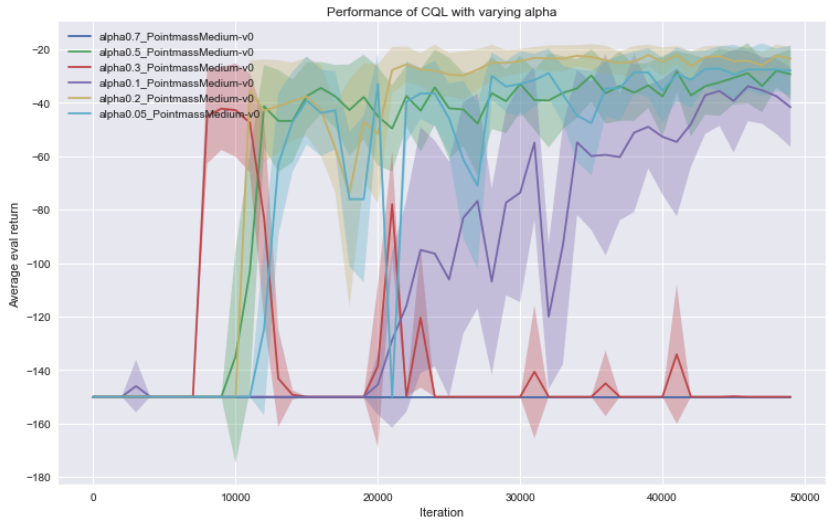


(1) **CQL vs. DQN:** As shown, CQL underestimates the states in general, leading to more stable learning performance. The reward shift was 1 and the reward scale was 100 for all experiments. CQL underestimates Q-values due to the regularizer which reduces the soft-maximum on Q values, but the overall loss still attempts to maximize the Q-values for (s, a) seen in the offline dataset. This means that performance is much more stable early on training as CQL assigns conservative Q-values to out-of-distribution states.



(2) **Exploration timescale:** As shown from the graph, the performance suffers significantly without enough exploration. With 5000 iterations, both the DQN and CQL agents are unable to learn any meaningful policies, while with 15000 they are able to perform quite well, with CQL improving over DQN (as expected).

Config	Final Return
(5000, DQN)	-150
(5000, CQL)	-150
(15000, DQN)	-26.43
(15000, CQL)	-22.28

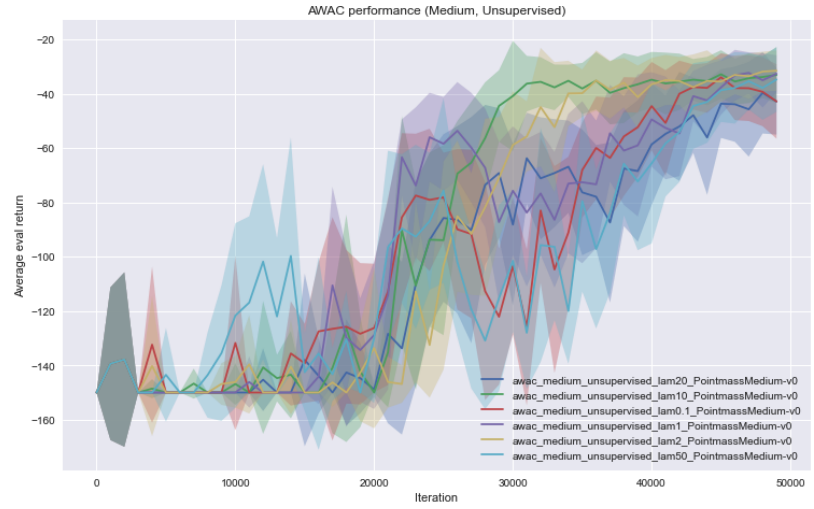


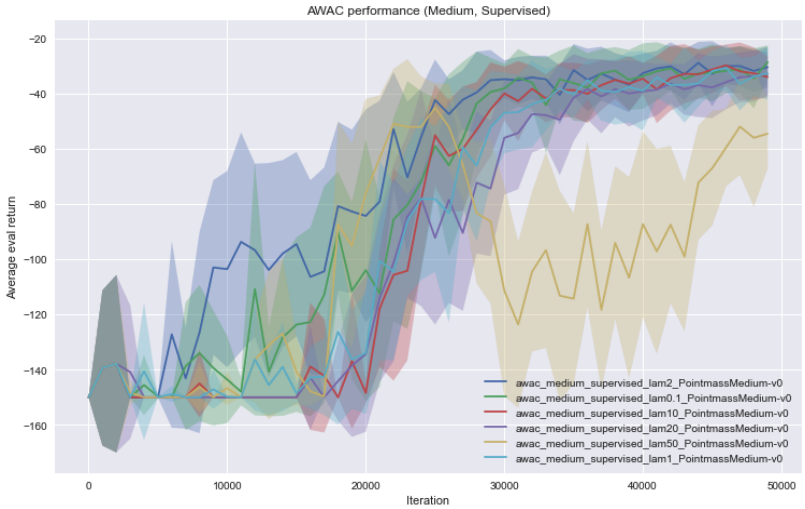
(3) **Effect of CQL alpha:** The final performance in the plot in ascending order: 0.7, 0.3, 0.1, 0.5, 0.05, 0.2. Interestingly, CQL is relatively sensitive to choice of alpha, with 0.05 (agent very similar to DQN) coming close to the performance of the other 3 best agents and there is no obvious trend. Additionally, over-regularizing the Q-values hurts performance significantly, with alpha=0.7 unable to make any forward progress, which aligns with expectations. The variance of each algorithm is also interesting to visualize, since early learning affects downstream Q-value estimates.

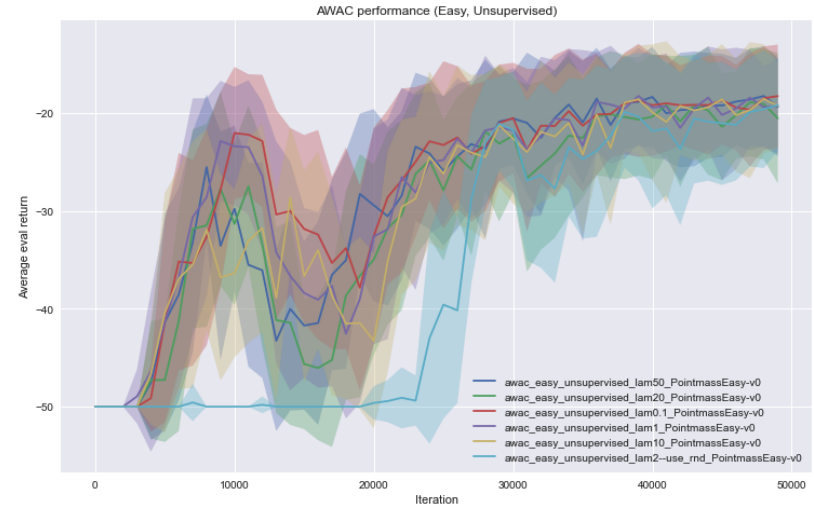
3. “Supervised” exploration with mixed bonuses

Mixed rewards: Compared to Part 2, the DQN performance is slightly improved with lower variance improvement over time and the CQL performance is worse in the early steps of learning, possibly due to the CQL regularization causing over underestimation of out-of-distribution states during learning due to the wider coverage of states because of the RND exploration strategy. Compared to Part 1, the performance is improved, likely due to the fact that RND is very important in the early stages of training but ultimately is harmful to continue exploration in later stages of training. As a result, the mixed reward shifts its weight towards environment reward, allowing the agent to stabilize and achieve improved performance as opposed to taking on exploration bonuses which may not be informative in the late stages of training.

4. Offline Learning with AWAC







AWAC: In most experiments, lambda had a minimal effect on final agent performance, both in unsupervised and supervised RND scenarios. However, in the supervised medium environment a larger lambda did result in worse agent performance, likely due to downweighting of the advantage too much. This results in an agent that is similar to behavioral cloning and is more susceptible to issues surrounding out-of-distribution states late in learning, resulting in unstable learning. Compared to CQL, AWAC is less sensitive to hyperparameters and has overall improved performance with lower variance in the initial stages of learning.

All commands

```
### Part 1
CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_expl.py --env_name PointmassEasy-v0 --use_rnd
--unsupervised_exploration --exp_name q1_env1_rnd_easy

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_expl.py --env_name PointmassEasy-v0
--unsupervised_exploration --exp_name q1_env1_random_easy

CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--unsupervised_exploration --exp_name q1_env1_rnd

CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0
--unsupervised_exploration --exp_name q1_env1_random

CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_expl.py --env_name PointmassHard-v0 --use_rnd
--unsupervised_exploration --exp_name q1_env2_rnd

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_expl.py --env_name PointmassHard-v0
--unsupervised_exploration --exp_name q1_env2_random



### Part 1.2
CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0
--unsupervised_exploration --pc_expl --exp_name q1_alg_med

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_expl.py --env_name PointmassHard-v0
--unsupervised_exploration --pc_expl --exp_name q1_alg_hard

### Part 2
CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --exp_name q2_dqn
--use_rnd --unsupervised_exploration --offline_exploitation --cql_alpha=0 --exploit_rew_shift=1
--exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --exp_name q2_cql
--use_rnd --unsupervised_exploration --offline_exploitation --cql_alpha=0.1 --exploit_rew_shift=1
--exploit_rew_scale=100

### Part 2.2
CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=5000 --offline_exploitation --cql_alpha=0.1 --unsupervised_exploration --exp_name
q2_cql_numsteps_5000  --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=15000 --offline_exploitation --cql_alpha=0.1 --unsupervised_exploration --exp_name
q2_cql_numsteps_15000  --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=5000 --offline_exploitation --cql_alpha=0.0 --unsupervised_exploration --exp_name
q2_dqn_numsteps_5000  --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=15000 --offline_exploitation --cql_alpha=0.0 --unsupervised_exploration --exp_name
q2_dqn_numsteps_15000  --exploit_rew_shift=1 --exploit_rew_scale=100

### Part 2.3
CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
```

```
--unsupervised_exploration --offline_exploitation --cql_alpha=0.05 --exp_name q2_alpha0.05
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--unsupervised_exploration --offline_exploitation --cql_alpha=0.1 --exp_name q2_alpha0.1
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--unsupervised_exploration --offline_exploitation --cql_alpha=0.2 --exp_name q2_alpha0.2
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--unsupervised_exploration --offline_exploitation --cql_alpha=0.3 --exp_name q2_alpha0.3
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--unsupervised_exploration --offline_exploitation --cql_alpha=0.5 --exp_name q2_alpha0.5
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--unsupervised_exploration --offline_exploitation --cql_alpha=0.7 --exp_name q2_alpha0.7
--exploit_rew_shift=1 --exploit_rew_scale=100

### Part 3
CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=20000 --cql_alpha=0.0 --exp_name q3_medium_dqn --exploit_rew_shift=1
--exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_expl.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=20000 --cql_alpha=1.0 --exp_name q3_medium_cql --exploit_rew_shift=1
--exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_expl.py --env_name PointmassHard-v0 --use_rnd
--num_exploration_steps=20000 --cql_alpha=0.0 --exp_name q3_hard_dqn --exploit_rew_shift=1
--exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_expl.py --env_name PointmassHard-v0 --use_rnd
--num_exploration_steps=20000 --cql_alpha=1.0 --exp_name q3_hard_cql --exploit_rew_shift=1
--exploit_rew_scale=100

### Part 4
CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --exp_name
q4_awac_medium_unsupervised_lam0.1 --use_rnd --unsupervised_exploration --awac_lambda=0.1
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --exp_name
q4_awac_medium_unsupervised_lam1 --use_rnd --unsupervised_exploration --awac_lambda=1
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --exp_name
q4_awac_medium_unsupervised_lam2 --use_rnd --unsupervised_exploration --awac_lambda=2
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --exp_name
q4_awac_medium_unsupervised_lam10 --use_rnd --unsupervised_exploration --awac_lambda=10
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --exp_name
```

```
q4_awac_medium_unsupervised_lam20 --use_rnd --unsupervised_exploration --awac_lambda=20
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=0 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --exp_name
q4_awac_medium_unsupervised_lam50 --use_rnd --unsupervised_exploration --awac_lambda=50
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100



CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=0.1 --exp_name q4_awac_medium_supervised_lam0.1
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=1 --exp_name q4_awac_medium_supervised_lam1
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=2 --exp_name q4_awac_medium_supervised_lam2
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=10 --exp_name q4_awac_medium_supervised_lam10
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=20 --exp_name q4_awac_medium_supervised_lam20
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=1 python cs285/scripts/run_hw5_awac.py --env_name PointmassMedium-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=50 --exp_name q4_awac_medium_supervised_lam50
--exploit_rew_shift=1 --exploit_rew_scale=100



CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --exp_name
q4_awac_easy_unsupervised_lam0.1 --use_rnd --unsupervised_exploration --awac_lambda=0.1
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --exp_name
q4_awac_easy_unsupervised_lam1 --use_rnd --unsupervised_exploration --awac_lambda=1
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --exp_name
q4_awac_easy_unsupervised_lam2--use_rnd --unsupervised_exploration --awac_lambda=2
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --exp_name
q4_awac_easy_unsupervised_lam10 --use_rnd --unsupervised_exploration --awac_lambda=10
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --exp_name
q4_awac_easy_unsupervised_lam20 --use_rnd --unsupervised_exploration --awac_lambda=20
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=2 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --exp_name
q4_awac_easy_unsupervised_lam50 --use_rnd --unsupervised_exploration --awac_lambda=50
--num_exploration_steps=20000 --exploit_rew_shift=1 --exploit_rew_scale=100
```

```
CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=0.1 --exp_name q4_awac_easy_supervised_lam0.1
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=1 --exp_name q4_awac_easy_supervised_lam1 --exploit_rew_shift=1
--exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=2 --exp_name q4_awac_easy_supervised_lam2 --exploit_rew_shift=1
--exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=10 --exp_name q4_awac_easy_supervised_lam10
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=20 --exp_name q4_awac_easy_supervised_lam20
--exploit_rew_shift=1 --exploit_rew_scale=100

CUDA_VISIBLE_DEVICES=3 python cs285/scripts/run_hw5_awac.py --env_name PointmassEasy-v0 --use_rnd
--num_exploration_steps=20000 --awac_lambda=50 --exp_name q4_awac_easy_supervised_lam50
--exploit_rew_shift=1 --exploit_rew_scale=100
```