Appendices

A Implementation Details

The implementation of our algorithm is based on the original implementation of BCQ: https://github.com/sfujim/BCQ. We train the CVAE first and then train the policy using the fixed decoder. The latent policy is a deterministic policy with tanh activation at the output. The output is then scaled by a hyperparameter max latent action. More discussions on the max latent action is in Appendix C. The perturbation layer is not used by default. We will discuss the effect of perturbation layer in Appendix D.

Hyper-parameters for MuJoCo datasets: The actor, the critic and the CVAE are optimized using Adam. The actor learning rate is 1e-4 and the critic learning rate is 1e-3. The CVAE learning rate is 1e-4. Both the encoder and the decoder have two hidden layers (750, 750) by default. For datasets smaller than 1e6 transitions such as the medium-replay datasets, we use (128, 128) to prevent overfitting. We train the CVAE for 5e5 timesteps with batch size 100. The latent policy, the critic and the perturbation layer have two hidden layers (400, 300). We use $\tau = 0.005$ for the soft target update. $\lambda = 1$ is used to calculate the Q-value target. The policy is trained for 5e5 timesteps with batch size 100.

Hyper-parameters for the robot experiment: The actor and critic learning rates are set to 3e-4 and the CVAE learning rate is 1e-4, with Adam as the optimizer. All of the networks have two hidden layers of size 64, including the actor, the critic, the encoder and the decoder. The smaller network sizes are to prevent overfitting. The CVAE is trained for 15000 iterations. For soft target update we use $\tau = 0.005$. We use $\lambda = 0.75$ for clipped double Q learning and use batch size of 256. The max latent action is set to 2.0.

B D4RL Results

To benchmark the performance of our algorithm, we include the full results for the d4rl MuJoCo datasets here as a reference. The numbers for the baselines are from the d4rl paper [7]. The results are averaged over 3 seeds. “Latent” refers to the latent policy without the perturbation layer. “Latent+P” refers to latent policy + perturbation layer. Our method consistently achieves good performance especially on medium-expert and medium-replay datasets. The other baselines work well on a part of the datasets and fail on the others.

In the current version of the d4rl dataset, hopper-medium-expert is actually a combination of the medium-replay and the expert datasets instead of the medium and the expert datasets. We have verified that the results given in their paper also correspond to the medium-replay + expert dataset. In Table 3 and Table 5 below, we use hopper-medium-expert(a) to refer to the results on this dataset. In addition, we generate the actual hopper-medium-expert by concatenating the medium and the expert datasets, referred as hopper-medium-expert(b) in the table. The results from Figure 4 and the other experiments in the appendix are all based on hopper-medium-expert(b).

Table 2: D4rl Benchmark Results: Average Reward.

Dataset	**BEAR**	**BRAC-v**	**BCQ**	**Latent (Ours)**	**Latent+P (Ours)**
walker2d-medium-expert	1842.7	4926.6	2640.3	4113.2	4465.0
hopper-medium-expert-(a)	3113.5	5.1	3588.5	3593.7	3062.5
hopper-medium-expert-(b)	2648.4	2245.7	2021.7	3592.4	3518.5
halfcheetah-medium-expert	6349.6	4926.6	7750.8	11716.9	12051.4
walker2d-medium-replay	883.8	44.5	688.7	1387.9	658.4
hopper-medium-replay	1076.8	-0.8	1057.8	888.4	1669.6
halfcheetah-medium-replay	4517.9	5640.6	4463.9	5172.6	5397.4
walker2d-medium	2717.0	3725.8	2441.0	2047.0	3072.4
hopper-medium	1674.5	990.4	1752.4	1050.4	1182.1
halfcheetah-medium	4897.0	5473.8	4767.9	4602.6	4964.6
walker2d-random	336.3	87.4	228.0	104.0	311.6
hopper-random	349.9	376.3	323.9	320.5	412.2
halfcheetah-random	2831.4	3590.1	-1.3	2922	3235.8

Table 3: D4rl Benchmark Results: Normalized Score

Dataset	**BEAR**	**BRAC-v**	**BCQ**	**Latent (Ours)**	**Latent+P (Ours)**
walker2d-medium-expert	40.1	81.6	57.5	89.6	97.2
hopper-medium-expert-(a)	96.3	0.8	110.9	111.0	94.7
hopper-medium-expert-(b)	82.0	69.6	62.7	111.0	108.7
halfcheetah-medium-expert	53.4	41.9	64.7	96.6	99.3
walker2d-medium-replay	19.2	0.9	15	30.2	14.3
hopper-medium-replay	33.7	0.6	33.1	27.9	51.9
halfcheetah-medium-replay	38.6	47.7	38.2	43.9	45.7
walker2d-medium	59.1	81.1	53.1	44.6	66.9
hopper-medium	52.1	31.1	54.5	32.9	36.9
halfcheetah-medium	41.7	46.3	40.7	39.3	42.2
walker2d-random	7.3	1.9	4.9	3.1	6.8
hopper-random	11.4	12.2	10.6	10.5	13.3
halfcheetah-random	25.1	31.2	2.2	25.8	28.3

Table 4: D4rl Results on More Datasets: Average Reward. For these datasets, we searched over $0.5, 1, 2$ for max latent action and report the best results.

Dataset	**BC**	**SAC-off**	**BEAR**	**BRAC-v**	**BCQ**	**Latent (Ours)**
maze2d-umaze	29.0	145.6	28.6	1.7	41.5	102.6
maze2d-medium	93.2	82.0	89.8	102.4	35.0	109.6
maze2d-large	20.1	1.5	19.0	115.2	23.2	334.6
antmaze-umaze	0.7	0.0	0.7	0.7	0.6	0.7
antmaze-umaze-diverse	0.6	0.0	0.6	0.7	0.7	0.5
antmaze-medium-play	0.0	0.0	0.0	0.0	0.0	0.2
antmaze-medium-diverse	0.0	0.0	0.1	0.0	0.0	0.0
antmaze-large-play	0.0	0.0	0.0	0.0	0.0	0.0
antmaze-large-diverse	0.0	0.0	0.0	0.0	0.0	0.0
pen-human	1121.9	284.8	66.3	114.7	2149.0	2101.0
hammer-human	-82.4	-214.2	-242.0	-243.8	-210.5	324.7
door-human	-41.7	57.2	-66.4	-66.4	-56.6	73.3
relocate-human	-5.6	-4.5	-18.9	-19.7	-8.6	7.1
pen-cloned	1791.8	797.6	885.4	22.2	1407.8	1558.0
hammer-cloned	-175.1	-244.1	-241.1	-236.9	-224.4	-142.9
door-cloned	-60.7	-56.3	-60.9	-59.0	-56.3	41.2
relocate-cloned	-10.1	-16.1	-17.6	-19.4	-17.5	-16.7
pen-expert	2633.7	277.4	3253.1	6.4	3521.3	3693.3
hammer-expert	16140.8	3019.5	16359.7	-241.1	13731.5	16333.5
door-expert	969.4	163.8	2980.1	-66.6	2850.7	3004.0
relocate-expert	4289.3	-18.2	4173.8	-21.4	1759.6	4528.5
kitchen-complete	1.4	0.6	0.0	0.0	0.3	1.4
kitchen-partial	1.4	0.0	0.5	0.0	0.8	1.8
kitchen-mixed	1.9	0.1	1.9	0.0	0.3	1.6

Table 5: D4rl Results on More Datasets: Normalized Score

Dataset	**BC**	**SAC-off**	**BEAR**	**BRAC-v**	**BCQ**	**Latent (Ours)**
maze2d-umaze	3.8	88.2	3.4	-16.0	12.8	57.0
maze2d-medium	30.3	26.1	29.0	33.8	8.3	36.5
maze2d-large	5.0	-1.9	4.6	40.6	6.2	122.7
antmaze-umaze	65.0	0.0	73.0	70.0	78.9	70.7
antmaze-umaze-diverse	55.0	0.0	61.0	70.0	55.0	45.3
antmaze-medium-play	0.0	0.0	0.0	0.0	0.0	16.0
antmaze-medium-diverse	0.0	0.0	8.0	0.0	0.0	0.7
antmaze-large-play	0.0	0.0	0.0	0.0	6.7	0.7
antmaze-large-diverse	0.0	0.0	0.0	0.0	2.2	0.3
pen-human	34.4	6.3	-1.0	0.6	68.9	67.3
hammer-human	1.5	0.5	0.3	0.2	0.5	4.6
door-human	0.5	3.9	-0.3	-0.3	0.0	4.4
relocate-human	0.0	0.0	-0.3	-0.3	-0.1	0.3
pen-cloned	56.9	23.5	26.5	-2.5	44.0	49.0
hammer-cloned	0.8	0.2	0.3	0.3	0.4	1.0
door-cloned	-0.1	0.0	-0.1	-0.1	0.0	3.3
relocate-cloned	-0.1	-0.2	-0.3	-0.3	-0.3	-0.2
pen-expert	85.1	6.1	105.9	-3.0	114.9	120.7
hammer-expert	125.6	25.2	127.3	0.3	107.2	127.1
door-expert	34.9	7.5	103.4	-0.3	99.0	104.2
relocate-expert	101.3	-0.3	98.6	-0.4	41.6	106.9
kitchen-complete	33.8	15.0	0.0	0.0	8.1	34.8
kitchen-partial	33.8	0.0	13.1	0.0	18.9	43.9
kitchen-mixed	47.5	2.5	47.2	0.0	8.1	40.8

C Sensitivity Analysis: Max Latent Action

The max latent action limits the range of output for the latent policy to ensure that the output has a high probability under the latent variable prior of the CVAE. As mentioned in Section 5.1, if the output of the latent policy has a high probability under the distribution of the latent variable prior, then the decoded output has a high probability to be within the distribution of the behavior policy. Larger max latent action may result in out-of-distribution actions. On the other hand, smaller max latent action will make the action selection more restrictive. We evaluated the effect of the max latent action from $\{0.5, 1, 2, 3\}$ over the MuJoCo datasets in d4rl as shown in Figure 6. In hopper-medium-replay and halfcheetah-medium-expert, 0.5 works the best. In most cases, 2 works well. Thus, we use 0.5 for hopper-medium-replay and halfcheetah-medium-expert and 2 by default for all the other environments for simplicity. Note that all the experiments for walker2d-random are unstable, thus the comparison across different parameters might not be valuable since we only average across 3 seeds.

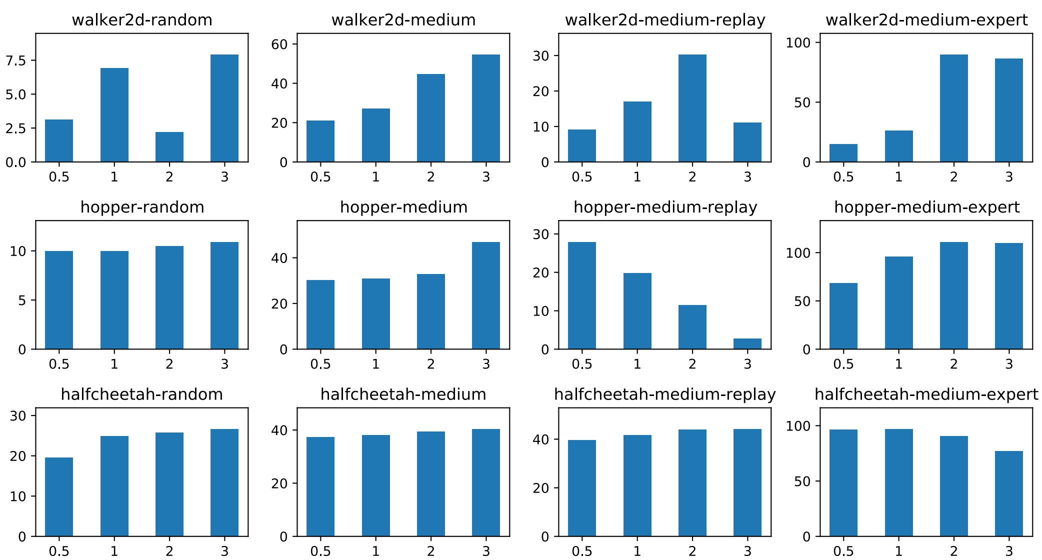

Figure 6: Sensitivity analysis on the max latent action for the latent policy: X-axis is the max latent action value. Y-axis is the normalized score.

D Ablation Study: Perturbation Layer

We provide a full comparison of the perturbation layer on MuJoCo datasets in this section. We summarize the results with max perturbation $\epsilon \in \{0, 0.01, 0.05, 0.1, 0.2, 0.5\}$ in Figure 7. $\epsilon = 0$ is only using the Latent Policy without the perturbation layer. As mentioned above, the walker2d-random experiments are not stable, thus the comparison might not be valuable. In most cases, the addition of perturbation layer sometimes improves the performance, but not significant. With a large ϵ higher than a certain value, the performance usually drops. Thus, we make the perturbation layer an optional component in our method.

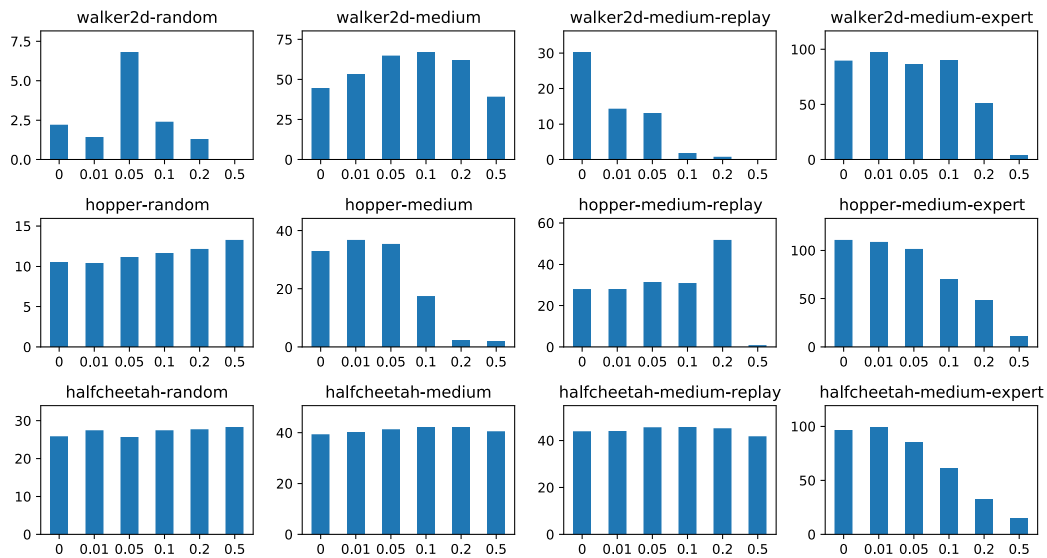

Figure 7: Ablation study on the perturbation layer: X-axis is the max perturbation. Y-axis is the normalized score.

E Empirical Analysis on MMD Constraint

To understand the limitation of using sampled MMD constraint to limit out-of-distribution actions, we simulate the MMD loss in different scenarios. In the first experiment, we construct a one-dimensional behavior policy sampling from $N(0,1)$ and an agent policy sampling from $N(0,x)$, where x is a variable. In Figure 8 below, we plot the MMD loss for this agent policy with different x as the x-axis with various kernel parameters. Ideally, the loss should be smaller than a threshold for any $x \leq 1$ to allow the agent policy to select the best action within the support with a higher probability. However, as shown in Figure 8, this is only roughly satisfied with the Gaussian kernel and large sigma. Sampled MMD constraint aims to match the entire support of two distributions and could be overly restrictive.

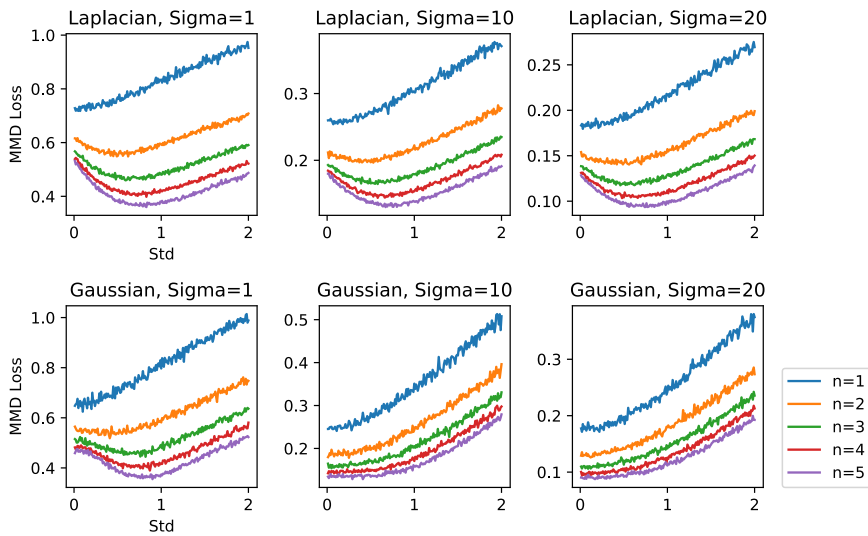

Figure 8: Simulated MMD loss with N(0,1) as the behavior policy.

In Figure 9, we further demonstrate the limitation of MMD constraint on multimodal distributions. We assume a behavior policy sampling uniformly from $[-2,-1]\bigcup[1,2]$ and an agent policy sampling from $N(x, 0.5)$. We vary the mean value x in the x-axis of the figures. In this case, we expect the minimum loss to happen at $x=-1.5$ and $x=1.5$ to prevent out of distribution actions. However, the simulation results show that this is not the case for any of the curves. With large sigma, the minimum MMD loss occurs at $x=0$, which lies in the "hole" of the behavior policy distribution.

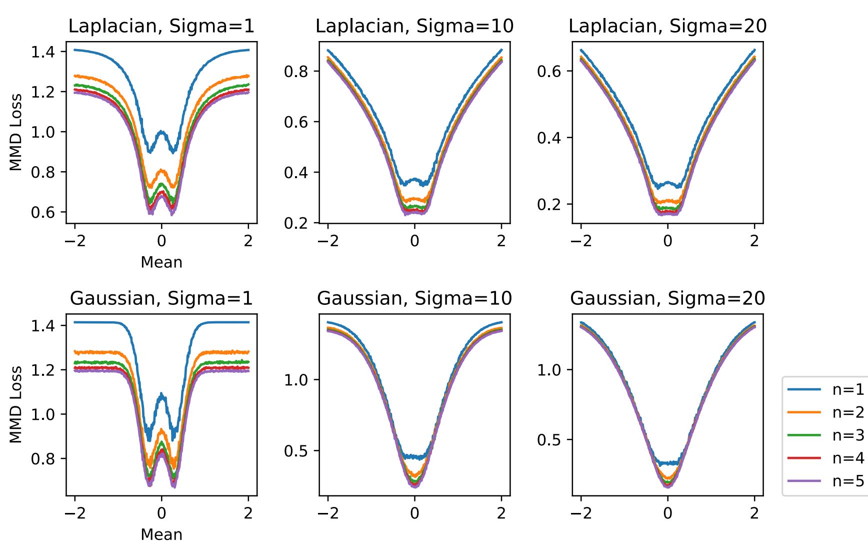

Figure 9: Simulated MMD loss with N(0,1) as the behavior policy.

F Robot Experiment

For the real robot experiments, we use a Sawyer robot equipped with a WSG 32 gripper and a WSG DSA tactile sensor finger. The physical setup involves a cloth with one corner clamped on a fixture. The task is to slide along the cloth as far as possible, ideally until the other corner is reached.

The movement of the end-effector is constrained to a vertical plane from the fixture. The observation space of the RL environment consists of the tactile sensor readings, end-effector force, and end-effector pose (z position and angle). The observations are thus in the form of a 89-d vector. The action space consists of horizontal and vertical delta position actions.

For each timestep, if the gripper is sliding along the cloth, it receives a reward equal to the horizontal action. This is to encourage faster sliding. However, if the edge is lost from the gripper, the reward will be zero for that timestep and the episode ends. This failure condition is detected based on the gripper width adjustment procedure discussed below. In addition, the maximum episode length is 70 timesteps.

We use a hard-coded procedure to adjust the gripper width. In general, we want to get clearer tactile readings of the cloth by grasping tightly but we also want to reduce the friction force such that the gripper can slide easily. The adjustment is based on the coverage and the mean value of the tactile readings as well as the end-effector force readings. When the gripper width is at the minimum value and there is still no tactile reading or force reading, we consider it as a failure and the episode ends.