
Mildly Constrained Evaluation Policy for Offline Reinforcement Learning

Anonymous Author(s)

Affiliation

Address

email

1 A Experiments

2 A.1 Implementations and hyper-parameters

3 For CQL, we reported the results from the IQL paper [Kostrikov et al., 2022] to show CQL re-
4 sults on "-v2" tasks. For IQL, we use the official implementation [Kostrikov, 2022] to obtain a
5 generally similar performance as the ones reported in their paper. Our implementations of TD3BC,
6 TD3BC-MCEP, AWAC, and AWAC-MCEP are based on [Kostrikov, 2022] framework. In all re-
7 implemented/implemented methods, clipped double Q-learning [Fujimoto et al., 2018] is used. In
8 TD3BC and TD3BC-MCEP, we keep the state normalization proposed in [Fujimoto and Gu, 2021]
9 but other algorithms do not use it.

The hyper-parameters used in the experiments are listed in Table 1.

batch size	BC	IQL	AWAC	AWAC-MCEP	TD3BC	TD3BC-MCEP
actor LR	1e-3	3e-4	3e-5	3e-5	3e-4	3e-4
actor ^e LR		-		3e-5	-	3e-4
critic LR	-	3e-4	3e-4	3e-4	3e-4	3e-4
V LR	-	3e-4			-	
actor/critic network	(256, 256)					
discount factor	0.99					
soft update τ	-	0.005				
dropout	0.1	-				
Policy	TanhNormal				Deterministic	
MuJoCo Locomotion						
τ for IQL	-	0.7			-	
$\lambda/\tilde{\lambda}$	-	$1/\lambda = 3$		1.0		-
λ^e		-		0.6		-
$\alpha/\tilde{\alpha}$			-			2.5
α^e				-		10.0
Adroit Locomotion						
τ for IQL	-	0.7			-	
$\lambda/\tilde{\lambda}$	-	2.0	1.0	10.0		-
λ^e		-		1000.0		-
$\alpha/\tilde{\alpha}$			-			0.1
α^e				-		0.1

Table 1: Hyper-parameters.

11 **A.2 Full results for estimated Q values of the learned evaluation policies**

12 Figure 1 and Figure 2 show the visualization of the estimated Q values in different D4RL MuJoCo locomotion tasks.

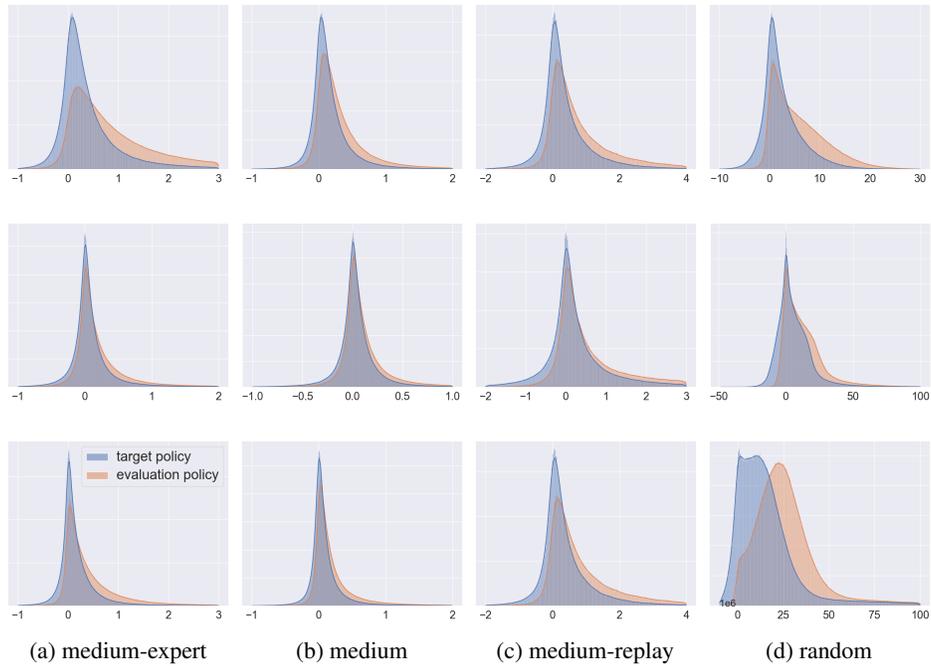


Figure 1: TD3BC-MCEP. **First row:** *halfcheetah*. **Second row:** *hopper*. **Third row:** *walker2d*.

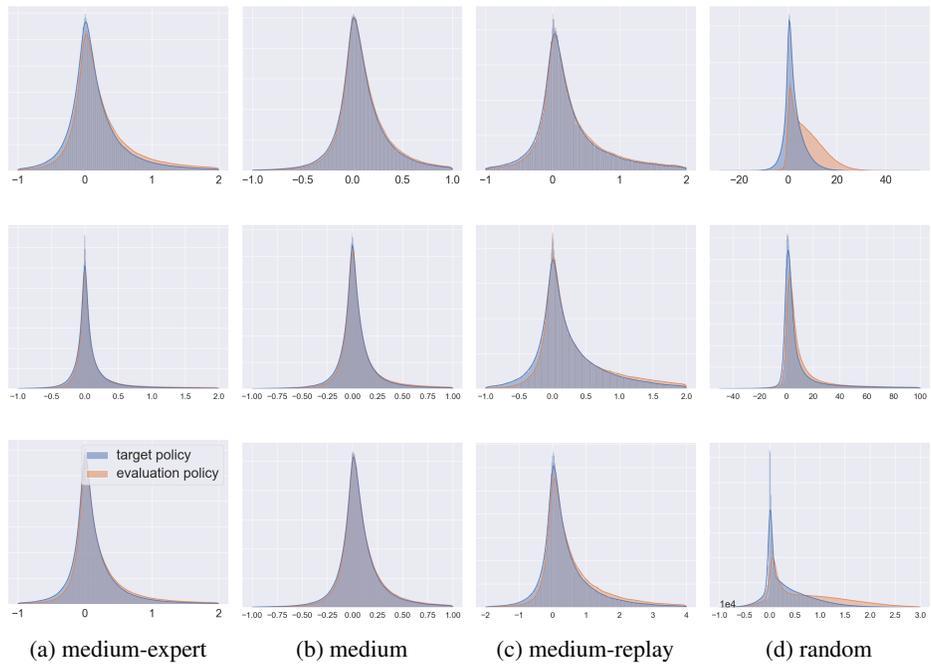


Figure 2: AWAC-MCEP. **First row:** *halfcheetah*. **Second row:** *hopper*. **Third row:** *walker2d*.

Task Name	BC	CQL	IQL	TD3BC	TD3BC -MCEP (ours)	TD3BC-MCEP -BCNorm (ours)	AWAC	AWAC -MCEP (ours)
halfcheetah-r	2.2±0.0	-	10±1.7	11.7±0.4	<u>28.8±1.0</u>	<u>31.6±2.7</u>	9.6±0.4	34.9±0.8
hopper-r	4.7±0.1	-	8.1±0.4	8.3±0.1	8.0±0.4	7.9±0.2	5.3±0.4	9.8±0.5
walker2d-r	1.6±0.0	-	5.6±0.1	1.2±0.0	-0.2±0.1	-0.3±0	5.2±1.0	3.1±0.4
halfcheetah-m	42.4±0.1	44.0	47.4±0.1	48.7±0.2	55.5±0.4	50.1±0	45.1±0	46.6±0
hopper-m	54.1±1.1	58.5	65±3.6	56.1±1.2	91.8±0.9	59.8±2.0	58.9±1.9	91.1±1.5
walker2d-m	71±1.7	72.5	80.4±1.7	85.2±0.9	88.8±0.5	86.8±0.5	79.6±1.5	83.4±0.9
halfcheetah-m-r	37.8±1.1	45.5	43.2±0.8	44.8±0.3	50.6±0.2	51.3±0.1	43.3±0.1	44.9±0.1
hopper-m-r	22.5±3.0	95.0	74.2±5.3	55.2±10.8	100.9±0.4	100.4±0.4	64.8±6.2	101.4±0.2
walker2d-m-r	14.4±2.7	77.2	62.7±1.9	50.9±16.1	86.3±3.2	90.0±0.6	84.1±0.6	84.6±1.3
halfcheetah-m-e	62.3±1.5	91.6	91.2±1.0	87.1±1.4	71.5±3.7	80.6±2.9	77.6±2.6	76.2±5.5
hopper-m-e	52.5±1.4	105.4	110.2±0.3	91.7±10.5	80.1±12.7	<u>95.1±9.1</u>	52.4±8.7	<u>92.5±8.3</u>
walker2d-m-e	107±1.1	108.8	111.1±0.5	110.4±0.5	111.7±0.3	110.8±0.2	109.5±0.2	110.3±0.1
Average	39.3	-	59.0	54.2	64.5	63.6	52.9	64.9

Table 2: Normalized episode returns on D4RL benchmark. The results (except for CQL) are means and standard errors from the last step of 5 runs using different random seeds. Performances that are higher than corresponding baselines are underlined and task-wise best performances are bolded. For each run, we take an average of 10 evaluations.

14 A.3 Behavior cloning normalization for TD3BC-MCEP.

15 In our experiments on D4RL MuJoCo tasks, we observe that the TD3BC-MCEP obtains imbalanced
16 performances on different tasks (See Table 2). Specifically, it shows weaker performances on 2 out of
17 3 *medium-expert* tasks. In these tasks, the proportion of expert data is 50%, which is relatively high.
18 To analyze the trade-off between the policy improvement term (maximizing Q) and the BC term, we
19 visualize the estimated Q and behavior cloning loss for the learned policies on the training data. The
20 first row in Figure 3 presents the TD3BC-MCEP learned policies on *halfcheetah* tasks. Compared to
21 the target policy (orange), the MCEP (blue) generally has larger BC losses in all datasets, including
22 *medium-expert* datasets where the target policy shows superior performances. It shows a consistent
23 pattern of staying less close to the behavior distribution, which is empowered by the milder constraint.
24 However, we want it to adaptively address this tradeoff for different datasets. To achieve this goal,
25 we designed a behavior cloning normalizer.

26 The objective for π_ϕ^e with behavior cloning normalizer is

$$\mathcal{L}_{\pi^e}(\phi) = \mathbb{E}_{(s,a) \sim \mathcal{D}} \left[-\lambda^e Q(s, \pi_\phi^e(s)) + \frac{1}{(a - \tilde{\pi}_\psi(s))^2} (a - \pi_\phi^e(s))^2 \right]. \quad (1)$$

27 Intuitively, this normalizer $1/(a - \tilde{\pi}_\psi(s))^2$ allows the evaluation policy to see how the target policy
28 $\tilde{\pi}_\psi$ performs the behavior cloning. When the target policy stays close to the behavior distribution, the
29 evaluation policy also stays closer to the behavior distribution. We also use the exponential-mean
30 average on the BC normalizer to avoid large value changes during training. For this TD3BC-MCEP-
31 BCNorm agent, we use $\tilde{\alpha} = 2.5$ and $\alpha^e = 20$ after our linear search in $\alpha^e = \{2.5, 5, 10, 15, 20, 25\}$.

32 The second row of Figure 3 shows the learned policies with the BC normalizer. We observe that
33 in *medium-expert* and *medium* datasets where strong behavior cloning benefits the performance,
34 the evaluation policy generally shows similar BC losses to the target policy. In *medium-replay* and
35 *random* datasets where the policy improvement term is essential for the performance, the evaluation
36 policy stays less close to the behavior distribution than the target policy. In Table 2 and Figure 3e, we
37 show that the proposed behavior cloning normalizer (-BCNorm) helps gain a better balance between
38 different tasks and retains the general performance.

39 References

- 40 Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning.
41 *Advances in neural information processing systems*, 34:20132–20145, 2021.
- 42 Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-
43 critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR,
44 2018.

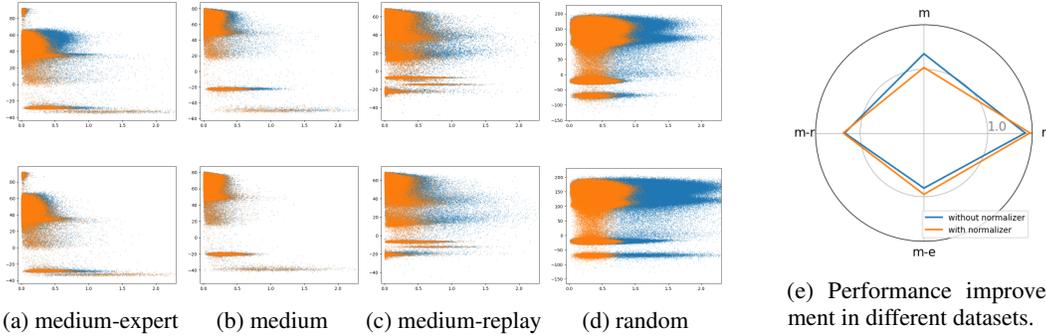


Figure 3: **Left:** We visualize the predicted q value $Q_{\bar{\pi}}(s, \pi)$ (the y axis) and the corresponding behavior-cloning loss $|a - \pi(s)|^2$ (the x axis) for the target policy (orange) and the evaluation policy (lue) on each datapoint in the dataset. **Right:** Average performances in D4RL MoJoCo locomotion tasks. *me*: medium-expert. *m*: medium. *mr*: medium-replay. *r*: random.

- 45 Ilya Kostrikov. JAXRL: Implementations of Reinforcement Learning algorithms in JAX, 10 2022.
 46 URL <https://github.com/ikostrikov/jaxrl2>. v2.
- 47 Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit
 48 q-learning. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=68n2s9ZJWF8>.
 49