Uncertainty-Aware Data Augmentation for Offline Reinforcement Learning

Yunjie Su, Yilun Kong, Xueqian Wang *
Tsinghua Shenzhen international graduate school, Tsinghua University
syj20@mails.tsinghua.edu.cn

Abstract

Data augmentation is commonly used to solve the short coverage of the full state-action space problem in Offline RL. However, the existing data augmentation methods for proprioceptive information meets a dilemma where the data coverage is limited by tight constraints, otherwise too aggressive method will hurt the performance. We aim to address the problem by our proposed algorithm **Uncertainty-Aware Data Augmentation (UADA)**, an effective and implementation-wise method. We extend the static offline datasets during training by adding gradient-based perturbation to the state and utilizing the estimated uncertainty of the value function to constrain the range of the gradient. The predictive uncertainty of the value function works as a guidance to adjust the range of augmentation automatically, ensuring the state perturbation adaptive and convincing. We plugged our method into standard offline RL algorithms and evaluated it on several offline reinforcement learning tasks. Empirically, we observe that UADA substantially improves the performance and achieves better model stability.

Introduction

Offline RL algorithms are proposed to learn policies from previously collected and static datasets without relying on environment interactions [Levine et al.2020, Janner, Li, and Levine2021,Chen et al.2019,Levine et al.2021]. These algorithms adopt great promises to solve many real-world problems when online interaction is costly or dangerous yet historical data is easily accessible, such as self-driving or industrial robotics.

Though such approaches achieve some degree of experimental success, they still face challenges in regard to the static dataset, which typically does not cover the full state-action space. When encountering an action or state unseen within the training set, the value estimation on out-of-distribution (OOD) actions or states can be arbitrary, resulting in destructive estimation errors that propagates through the Bellman loss and slows the learning and brings instability.

**Corresponding author

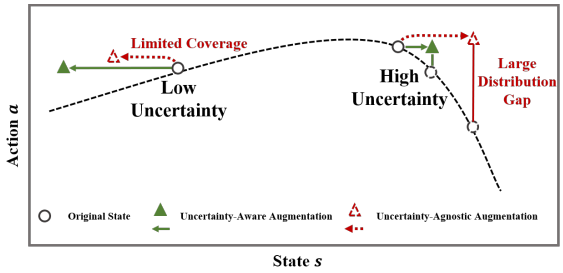


Figure 1: **Benefits of UADA.** On the one hand, by detecting the uncertainty, we can exert looser perturbation constraint on the original tuple to acquire more coverage of state-action pair. On the other hand, we can avoid large distribution gap during data augmentation in high uncertainty zone.

The prior attempt for OOD problem is data augmentation, which serves as a simple technique to achieve local exploration. Since it can smooth out the state space by "visiting" the local regions and ensure the learned value estimations are similar. However, we observe that the most effective proprioceptive information augmentation method [Sinha, Mandlekar, and Garg2022]: the adversarial state training, choosing the augmentation direction where the value function deviates the most, is sensitive to the size of the gradient constraint. When the constraint is tight, the generalization ability to unseen data is little. Otherwise, it will end up hurting the RL agent, since the reward for the original state may not coincide with the reward obtained from the augmented state.

In this paper, we adopt the uncertainty of the value prediction to adversarial data augmentation. On the one hand, the uncertainty estimation of the value function considers the errors of neural network's approximation and prevents aggressive augmentation from data points that induce high uncertainty scores; on the other hand, adversarial augmentation method can leverage active computation of physically-plausible adversarial examples during training to enable robust policy learning. In conclusion, we put forward a neighbourhood perturbation around the given state where the behavioral policy is likely to choose the same action without changing the semantics based on the adaptive constraint.

The proposed framework works as a plugin to be added to a number of off-the-shelf offline RL algorithms. Experimentally, we conduct our approach on several challenging benchmark datasets from D4RL [Fu et al.2020], and we are able to artificially increase the amount of data available during training, thereby improving the generalization ability.

Related Work

The proposed model-free offline RL algorithms [Levine et al.2020, Fujimoto, Meger, and Precup2018, Wu, Tucker, and Nachum2019] can be divided into several parts to solve poor out-of-distribution problem:1) Policy Constraint [Fujimoto, Meger, and Precup2018, Kumar et al.2019, Siegel et al.2020]: directly constraining learned policy to stay inside distribution, or with the support of dataset. 2) Value Regularization [Kumar et al.2020, Kostrikov et al.2021]: regularizing value function by assigning low values to out-of-distribution (OOD) actions. 3) Uncertainty Estimation [Agarwal, Schuurmans, and Norouzi2019, Wu et al.2021]: conducting a proper estimation and usage of uncertainty. However, these methods share similar traits of being conservative or omitting evaluation on OOD data, which brings benefits of minimizing model exploitation error, but at the expense of poor generalization of learned policy in OOD regions.

Data augmentation is a useful way to obtain the same datapoint from multiple viewpoints in computer vision research [Wu et al.2021, Chen et al.2021, Peng et al.2021] and also has become a widely used technique in visual RL for acquiring sample-efficient and generalizable policies [Ma et al.2022]. As for proprioceptive information, S4RL [Sinha, Mandlekar, and Garg2022] tested several data augmentation schemes to investigate the role of data augmentations, which also reflects some techniques may omit important information about the state of the robot and lead to worse performance, such as Dimension-Dropout and State-Switch, which are popular computer vision data augmentation algorithms. Conclusively, the aggressive adversarial perturbation strength hurts the performance since the new states may be semantically different from the original state.

Another attribution for OOD problem is the function approximation of the neural networks. Exploiting the prevalence of uncertainty in the underlying DRL algorithm leverages information about their distributions to improve the learning process. [Osband, Aslanides, and Cassirer2018]. Uncertainty estimation has been implemented in model-free RL for safety and risk estimation or exploration [Hoel, Tram, and Sjöberg2020]. In the offline RL setting, where the dataset is limited, uncertainty-weighted actor-critic uses inverse-variance weighting to discard out-of-distribution state-action pairs [Wu et al.2021]. To our limited acknowledgement, our method is the first to combine data augmentation with uncertainty estimation in offline RL.

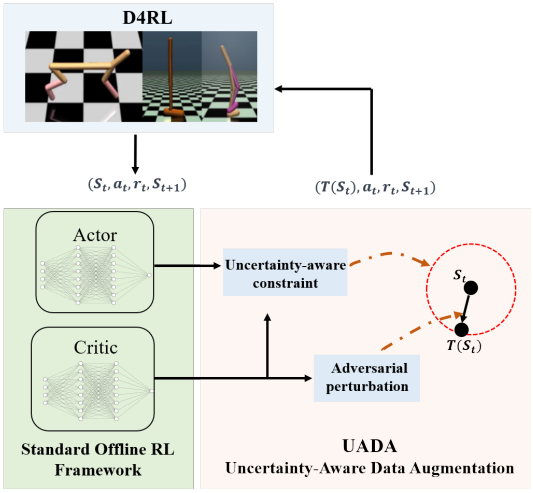


Figure 2: **Overview of UADA**. We perform state perturbations in two phases: we utilize the loss function to choose the direction where value function deviates the most and apply Monte Carlo dropout for uncertainty estimation to constrain the size of the gradient adaptively.

Uncertainty-Aware Data Augmentation Framework

In offline RL, the agent aims to learn an optimal policy by sampling experiences from the given datasets $D = (s_t, a_t, s_{t+1}, R_t)$. Since our implementation is applied to actor-critic based methods, the Actor function approximates the policy and the Critic function evaluates the value of state-action pair, aiming to maximize the expected γ-discounted cumulative reward: $\mathbb{E}_\pi[\sum_{t=0}^{T} \gamma^t r_\pi(s_t, a_t)]$. The policy is optimized to maximize the value function via policy improvement:

$$\pi_{i+1} \leftarrow \arg\max_{\pi} \mathbb{E}_{s_t \sim \mathcal{D}}[\hat{Q}\left(s_t, \pi(a_t|s_t)\right)]. \tag{1}$$

Data augmentations from states should promise the output of a small transformation on an input state to be physically realizable. There exists assumptions that the local transformation to perturb the state will not change the semantics and the reward function is smooth. We denote a data augmentation transformation as $T(s_t)$, where $s_t \in D$ and T is the transformation function. The augmented tuple will be $(\, T(s_t), a_t, s_{t+1}, R_t \,)$ which shares the same action a_t, next state s_{t+1}, and reward R_t as in the original tuple.

Adversarial Perturbation. Deep learning models are highly vulnerable to adversarial examples and are often fooled by the same adversarial example. In reinforcement learning, the objective is to misguide the policy to output incorrect actions to better encounter changes in the input state. In data augmentation, to generate a perturbation on a state, we use an isometrically scaled version of the full gradient:

$$\delta = \epsilon \nabla_{S_t} \mathbb{J}_Q(Q(S_t, a_t)) \tag{2}$$

where $\nabla_{S_t} \mathbb{J}_Q(Q(S_t, a_t))$ is a loss function over the policy update as in Eq. 1 where the value deviates the most [Man-

dlekar et al.2017], and ϵ corresponds to the perturbation strength, which is related to the uncertainty estimation.

Algorithm 1 Uncertainty-Aware Data Augmentation

Input: Dataset $\mathcal{D}$, target network update rate τ, mini-batch size N, sampled actions for MMD ($n = 10$), sample numbers stochastic forward passes ($T = 100$), hyperparameters:λ, α, β

1: Initialize Q networks $\{Q_{\theta_1}, Q_{\theta_2}\}$ and target network $\{Q_{\theta'_1}, Q_{\theta'_2}\}$ with MC Dropout;
Initialize actor $\{\pi_{\phi_1}, \pi_{\phi_2}\}$ and target actor $\{\pi_{\phi'_1}, \pi_{\phi'_2}\}$
2: **for** $t \leftarrow 1$ to N **do**
3: Sample mini-batch of transitions $(s, a, r, s') \sim \mathcal{D}$ added with the previous enhanced record
4: **Determine action candidates**
$\mathcal{M}_k = \{a_i\}_{i=1}^{k}\ a_i \sim \mathcal{N}(\mu(s'; \phi_i), \sigma)\ (i = 1, 2)$
$a^*_{K1} = \arg\max_{a_i \in \mathcal{M}_k} Q(s', a_i; \phi'_1)$
$a^*_{K2} = \arg\max_{a_j \in \mathcal{M}_k} Q(s', a_j; \phi'_2)$
5: Calculate $Q(s, a) = r +$
$\gamma \max_{a_i}[\lambda \min_{j=1,2}\{Q_{\theta'_1}(s', \mu(s'; \phi'_1)), Q_{\theta'_2}(s', a^*_{K1})\}$
$+(1-\lambda) \max_{j=1,2}\{Q_{\theta'_2}(s', \mu(s'; \phi'_2)), Q_{\theta'_1}(s', a^*_{K2})\}]$
6: Calculate variance of the $y(s, a)$ through variance of T stochastic samples from $Q_{\theta'_1}, Q_{\theta'_2}$ according to Eq.3 as the **MC dropout uncertainty estimation** and finally reach the **Constraint** ϵ in Eq. 4
7: **Policy-update** with uncertainty
$\pi_{i+1} \leftarrow \arg\max_\pi \mathbb{E}_{s_t \sim \mathcal{D}}[\frac{\beta}{Var[y(s,a)]} Q\left(s_t, \pi(a_t|s_t)\right)]$
and calculate the perturbation direction as in Eq. 2
8: **Generate Augmented state**
$T(S_t) \leftarrow S_t + \epsilon \nabla_{S_t} \mathbb{J}_Q(Q(S_t, a_t))$ and produce a new transition $(T(s), a, s', r)$ for future training
9: Update networks as in [Wu et al.2021]
10: **end for**

Uncertainty-estimation Constraint. Our augmentation method depends on the gradient with respect to the value function, while the neural network may not always have the accurate approximation. We focus on uncertainty estimation to constrain the range of the perturbation to mitigate the impact of unreliable value prediction. The time-wise or trajectory-wise uncertainty estimation methods are incompatible with the offline RL problem, since the dataset dose not contain trajectories. Therefore, we decide to detect the uncertainty of the state-action pair by dropout uncertainty estimation. We denote $X = (s, a)$ and $Y = Q(s, a)$. We draw inspiration from a Bayesian formulation for the value function in RL parameterized by θ, and maximize $p(\theta|X, Y) = p(Y|X, \theta)p(\theta)/p(Y|X)$ as our objective. Since $p(Y|X)$ is intractable, we approximate the above inference process through dropout variational inference by training with dropout before every weight layer [Gal and Ghahramani2016]. We will capture the uncertainty through the approximate predictive variance with respect to the estimated $\hat{Q}$ for T stohastic forward passes:

$$Var[Q(s,a)] \approx \sigma^2 + \frac{1}{T}\sum_{t=1}^{T} \hat{Q}_t(s,a)^T \hat{Q}_t(s,a) - E[\hat{Q}(s,a)]^T E[\hat{Q}(s,a)] \quad (3)$$

with σ^2 representing the inherent noise in the data, the second term representing how much the model is uncertain about its prediction, and $E[\hat{Q}(s, a)]$ for predictive mean. Capturing the model uncertainty, we finally define the ϵ as follows:

$$\epsilon = \frac{\lambda}{|Var[Q(s,a)|} \quad (4)$$

λ is the hyperparameter. The resulting ϵ intuitively discourages from exerting aggressive augmentation on state-action pair with high uncertainty of value function estimation.

Empirical evaluation

In this section, we will first conduct experiments on the popular D4RL benchmark for offline RL. The benchmark covers various different tasks such as locomotion tasks with Mujoco Gym, Antmaze, and other robotics tasks such as kitchen and adroit require hierarchical planning. We compare UADA to the two best performing augmentation variants from S4RL [Sinha, Mandlekar, and Garg2022], adding augmentation with Gaussian noise(S4RL($\mathcal{N}$)) and normal adversarial training(S4RL(*Adv*)). We use the same hyperparameters as proposed in the respective papers.

Task Name	CQL	S4RL($\mathcal{N}$)	S4RL(*Adv*)	CQL+UADA
antmaze-u	74.0	91.3	94.1	**99.6**
antmaze-u-d	84.0	87.8	88.0	**91.3**
antmaze-m-p	61.2	**61.9**	61.6	61.8
antmaze-m-d	53.7	78.1	82.3	**85.9**
antmaze-l-p	15.8	24.4	**25.1**	**25.1**
antmaze-l-d	14.9	27.0	26.2	**32.1**
cheetah-r	35.4	52.3	53.9	**54.1**
cheetah-m	44.4	48.8	48.6	**59.0**
cheetah-m-r	42.0	51.4	51.7	**58.9**
cheetah-m-e	62.4	79.0	78.1	**79.2**
hopper-r	**10.8**	**10.8**	10.7	**10.8**
hopper-m	58.0	78.9	81.3	**92.7**
hopper-m-r	29.5	35.4	36.8	**49.0**
hopper-m-e	111.0	113.5	117.9	**123.5**
walker-r	7.0	24.9	**25.1**	**25.0**
walker-m	79.2	93.6	93.1	**104.4**
walker-m-r	21.1	30.3	35.0	**48.2**
walker-m-e	98.7	112.2	107.1	**115.9**

Table 1: **Experiments on on the OpenAI Gym subset of the D4RL tasks**.We conduct the baseline CQL results directly from [Kumar et al.2020] and report the mean normalized episodic returns over 5 random seeds using the same protocol as [Sinha, Mandlekar, and Garg2022].

We summarize the overall results in Table 1. Our method is consistently outperform both the baseline CQL and S4RL across multiple tasks and data distributions. Outperforming

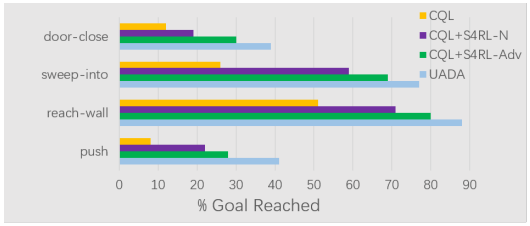
Dataset	BCQ	BCQ+UADA	BEAR	BEAR+UADA	TD3+BC	TD3+BC+UADA
hopper-r	**10.6**	10.4	11.4	**11.9**	**9.8**	9.2
halfcheetah-r	2.2	**2.3**	**25.1**	**25.1**	2.1	**2.3**
walker2d-r	**4.9**	4.6	7.3	**7.5**	**1.6**	1.4
hopper-m	54.5	**55.2**	52.1	**54.3**	29.0	**32.5**
halfcheetah-m	40.7	**42.8**	41.7	**43.8**	36.1	**40.2**
walker2d-m	53.1	**58.3**	59.1	**60.2**	6.6	**7.6**
hopper-m-r	33.1	**33.5**	33.7	**35.1**	11.8	**12.2**
halfcheetah-m-r	38.2	**38.9**	38.6	**39.7**	**38.4**	35.2
walker2d-m-r	15.0	**19.7**	19.2	**20.9**	11.3	**13.8**
hopper-m-e	110.9	**114.5**	96.3	**100.6**	111.9	**113.8**
halfcheetah-m-e	64.7	**68.4**	53.4	**59.2**	35.8	**40.4**
walker2d-m-e	57.5	**60.3**	40.1	**47.3**	6.4	**9.8**
locomation total	485.4	508.9	478	505.6	300.8	318.4
antmaze-u	68.7	**70.2**	56.7	**62.4**	78.6	**80.5**
antmaze-u-d	61.2	**70.5**	49.3	**50.9**	71.4	**80.3**
antmaze-m-p	35.3	**38.7**	0.0	0.0	10.6	**15.9**
antmaze-m-d	27.3	**30.8**	**0.7**	0.5	3.0	**9.6**
antmaze-l-p	2.2	**2.5**	0.0	0.0	1.0	1.0
antmaze-l-d	41.2	**45.8**	1.0	**1.4**	0.0	0.0
antmaze-total	235.9	258.5	107.7	115.2	164.6	187.3

Table 2: Normalized average scores comparison of baseline methods vs.**UADA** + baselines over 3 seeds on benchmarks tasks.

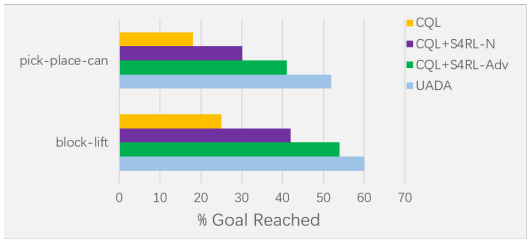
S4RL-variants on various different types of environments suggests that UADA fundamentally improves the data augmentation strategies discussed in S4RL.
We also experimentally employ our UADA methods to several recent strong baseline offline RL methods, such as CQL [Kumar et al.2020], BEAR [Kumar et al.2019] and TD3+BC [Fujimoto and Gu2021] on D4RL benchmarks.The results of CQL and BEAR are obtained by running their official codes, and we take the results of TD3+BC from its original paper. Results are shown in Table 2. With our proposed UADA, in most tasks, there exists improvements comparing with the original methods, reflecting our method can work as an accelerator for the existed offline RL methods.

We also note that S4RL-variants outperform UADA on the "-random" split of the data distributions, which is expected as UADA depends on uncertainty estimation to guide the data augmentation strategy. Since the "-random" split consists of random actions in the environment, tuples will be assigned tight constraint of the augmentation and the augmentation strategy will not bring more data coverage compared with the original ones.

We finally demonstrate the benefits of UADA by combining it with CQL and evaluate it on challenging environments such as kitchen and adroit tasks [Rajeswaran et al.2017], which have sparse reward and large action spaces as well as requiring hierarchical planning. Our experiment results are shown in figure 3. UADA significantly boosts the performance of vanilla CQL on such challenging environments



(a) Meta World Environments



(b) RoboSuite Environments

Figure 3: Results on challenging dexterous robotics environments using data collected by a similar strategy as S4RL. We report the percentage of goals that the agent is able to reach during evaluation.

and perform better than the augmentation baseline S4RL($\mathcal{N}$) and S4RL(*Adv*).

Conclusion

In this work, the main contribution is our proposed Uncertainty-Aware Data Augmentation technique for offline RL datasets. The proposed framework is general and can be added to a number of off-the-shelf offline RL algorithms. Empirically, we evaluate our approach on several challenging benchmark datasets from D4RL, MetaWorld and Robosuite, and we find that by using UADA we can improve the state-of-the-art performance on most benchmark offline reinforcement learning tasks as well as stabilizing the training process. Future works can combine the uncertainty-aware augmentation schemes with better self-supervised learning algorithms.

References

Agarwal, R.; Schuurmans, D.; and Norouzi, M. 2019. An optimistic perspective on offline reinforcement learning. *international conference on machine learning*.

Chen, X.; Zhou, Z.; Wang, Z.; Wang, C.; Wu, Y.; and Ross, K. W. 2019. Bail: Best-action imitation learning for batch deep reinforcement learning. *neural information processing systems*.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. *arXiv: Learning*.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.

Fujimoto, S., and Gu, S. S. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems* 34:20132–20145.

Fujimoto, S.; Meger, D.; and Precup, D. 2018. Off-policy deep reinforcement learning without exploration. *international conference on machine learning*.

Gal, Y., and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059. PMLR.

Hoel, C.-J.; Tram, T.; and Sjöberg, J. 2020. Reinforcement learning with uncertainty estimation for tactical decision-making in intersections. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 1–7. IEEE.

Janner, M.; Li, Q.; and Levine, S. 2021. Reinforcement learning as one big sequence modeling problem. *arXiv: Learning*.

Kostrikov, I.; Tompson, J.; Fergus, R.; and Nachum, O. 2021. Offline reinforcement learning with fisher divergence critic regularization. *international conference on machine learning*.

Kumar, A.; Fu, J.; Tucker, G.; and Levine, S. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *arXiv: Learning*.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative q-learning for offline reinforcement learning. *neural information processing systems*.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv: Learning*.

Levine, S.; Huang, C.; Smith, L.; Nair, A.; Pong, V. H.; Pong, V.; Nair, A.; Smith, L.; Huang, C.; and Levine, S. 2021. Offline meta-reinforcement learning with online self-supervision. *international conference on machine learning*.

Ma, G.; Wang, Z.; Yuan, Z.; Wang, X.; Yuan, B.; and Tao, D. 2022. A comprehensive survey of data augmentation in visual reinforcement learning.

Mandlekar, A.; Zhu, Y.; Garg, A.; Fei-Fei, L.; and Savarese, S. 2017. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3932–3939. IEEE.

Osband, I.; Aslanides, J.; and Cassirer, A. 2018. Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems* 31.

Peng, X. B.; Kumar, A.; Zhang, G.; and Levine, S. 2021. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv: Learning*.

Rajeswaran, A.; Kumar, V.; Gupta, A.; Vezzani, G.; Schulman, J.; Todorov, E.; and Levine, S. 2017. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*.

Siegel, N.; Springenberg, J. T.; Berkenkamp, F.; Abdolmaleki, A.; Neunert, M.; Lampe, T.; Hafner, R.; Heess, N.; and Riedmiller, M. 2020. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv: Learning*.

Sinha, S.; Mandlekar, A.; and Garg, A. 2022. S4rl: Surprisingly simple self-supervision for offline reinforcement learning in robotics.

Wu, Y.; Zhai, S.; Srivastava, N.; Susskind, J. M.; Zhang, J.; Salakhutdinov, R.; and Goh, H. 2021. Uncertainty weighted actor-critic for offline reinforcement learning. *international conference on machine learning*.

Wu, Y.; Tucker, G.; and Nachum, O. 2019. Behavior regularized offline reinforcement learning. *arXiv: Learning*.