CS285 Final Project - DIQL: Distance-Sensitive Q-Learning

Yutaka Shimizu, Hiva Mohammadzadeh, Bill Zheng[1]

Abstract

Offline reinforcement learning (Offline RL) has gained substantial traction for its capacity to utilize pre-existing datasets for agent training, circumventing the complexities associated with online interactions. Despite diligent endeavors in recent years, these initiatives often grapple with conservative estimations of action values (Q-values). This paper presents Distance-sensitive Implicit Q-Learning (DIQL), a novel offline reinforcement learning algorithm tailored to harness the intrinsic structure and attributes of datasets. DIQL extends upon Implicit Q-Learning (IQL) by integrating a mechanism that adjusts the conservatism level based on the proximity of queried data points to the dataset. We demonstrate the superior performance of our algorithm, Distance-sensitive Implicit-Q Learning (DIQL), through several Open-AI Gym environments as well as a custom Grid-World environment, showing its efficacy over prior methods. DIQL effectively balances the need for conservatism in value estimations with the ability to exploit the available data efficiently.

1. Introduction

Reinforcement Learning (RL) has emerged as a powerful paradigm in machine learning, enabling algorithms to learn optimal strategies through interaction with their environment. However, the majority of RL methods rely on active interaction with the environment, which is not always feasible or practical. This limitation has led to the development of offline RL, where algorithms learn from a fixed dataset without further interaction with the environment. Offline RL emerges as a promising solution, utilizing pre-collected datasets to train agents. These algorithms can learn from large, previously collected datasets, without interaction. This approach enables algorithms to learn from extensive, previously gathered datasets without further interaction, marking a significant evolution in RL. Despite its promise, offline RL faces unique challenges, primarily regarding the estimation of action values (Q-values) from limited, static datasets.

One of the main issues in offline RL is the overestimation of Q-values, especially for out-of-distribution (OOD) actions, leading to suboptimal policy learning. This problem is further exacerbated in model-free offline RL, where algorithms lack an explicit model of the environment and must entirely rely on the dataset. Traditional approaches, such as Conservative Q-Learning (CQL) and Implicit Q-Learning (IQL), have attempted to mitigate this by introducing various forms of pessimism in the estimation process. However, these methods often result in overly conservative estimates, particularly for OOD actions, hindering the algorithm's ability to effectively learn from the available data.

In this paper, we introduce Distance-sensitive Implicit Q-Learning (DIQL), a novel algorithm that addresses these challenges and the limitations of traditional Offline RL methods. DIQL builds on the principles of IQL but incorporates a unique mechanism that adjusts the level of conservatism based on the proximity of the queried data point to the dataset. With a distance-sensitive function, DIQL can vary its target values based on the closeness of the queried point to the dataset. This approach allows for less conservative Q-value estimates when the data point is close to the dataset, potentially overcoming the limitations of excessive pessimism seen in existing methods.

In this paper, we will first review the limitations of current offline RL methods, including model-free approaches and their drawbacks in handling OOD actions. We then introduce our novel DIQL algorithm, detailing its methodology and the rationale behind incorporating distance sensitivity into the Q-learning framework. Through this paper, we aim to provide a comprehensive understanding of DIQL, its underlying principles, and its practical implications. Finally, we test our algorithm on some of the well-known Open-AI Gym environments as well as customized Grid-World environments. Our results highlight the advantages of DIQL over traditional offline RL methods, showcasing its potential in producing more accurate and less conservative Q-value

[1] University of California, Berkeley. Correspondence to: Yutaka Shimizu, Hiva Mohammadzadeh, Bill Zheng < purewater0901@berkeley.edu, hiva@berkeley.edu, bill.cy.zheng@berkeley.edu>.

estimates. We believe that our contributions will not only address the existing challenges in offline RL but also catalyze further innovations in this rapidly evolving field.

2. Related Works

Our work builds on a variety of different algorithms designed for offline RL tasks with a concentration on improving the estimates of pessimism within the dataset.

Model-free Offline RL. To counteract the overestimation bias of modern Most other offline RL algorithm conditions the network to pessimistically estimate any and all out of distribution (OOD) values. (Kumar et al., 2020) have used an incremental loss to actively punish any actions that are OOD while (Lyu et al., 2022) used an approach where . While some learning frameworks have attempted to distinguish , some have universally implemented pessimism across any and all data distributions in reinforcement learning (Ma et al., 2021). Recently, more works by directly optimizing the soft-value function shown in CQL (Garg et al., 2023); however many of these works building on CQL have discarded any and all out od distribution actions.

Implicit Q-Learning. Implicit Q-Learning (IQL) and its derivative algorithms are known for its sample efficiency and ability to fine-tune online while maintaining adequate conservatism (Kostrikov et al., 2022). These models have based themselves in AWAC and other forms of weighted behavior cloning and other off-policy learning methods (Nair et al., 2021). In recent months, some have utilized other derivative models to implement this in a wider setting (Hansen-Estruch et al., 2023). IQL has also recently been implemented in model-based control (Chitnis et al., 2023), showing promise in other fields of RL other than model-free learning.

OOD Detection. Previous works have used exploration methods (Burda et al., 2019) to implicitly calculate whether a newly sampled datapoint is OOD or not. Newer advancements have instead focused on supervised learning methods (Yang et al., 2022). Although some OOD detection methods have been proven effective in various areas of supervised learning with recent works (Fort et al., 2021), it is uncertain whether SOTA methods are customizable (Tajwar et al., 2021), which would be important for offline RL methods.

3. Background

The RL problem is based on Markov Decision Processes (MDP) (state space $\mathcal{S}$, action space $\mathcal{A}$, initial state distribution $p_0(s)$, environment dynamics $p(s' \mid s, a)$, reward function $r(s, a)$, discount factor γ), where our objective is to derive the maximum expected reward:

$\pi^* = \arg\max_\pi \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$

Such that $s_0 \sim p_0(s), a_t \sim \pi(\cdot \mid s_t), s_{t+1} \sim p(\cdot \mid s, a)$. In off-policy methods, we would then employ a Q-function $Q(s, a)$ to document the discounted returns depending both on the state and the action.

Offline RL Methods. Compared to online RL methods, which collects data on-the-go and computes estimates of policies' returns and changes the parameters of the network, offline RL methods samples data from a data buffer. With respect to Q-learning, we employ a TD-loss objective:

$$\min_{Q_\theta^k} \mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[r(\mathbf{s},\mathbf{a})+\gamma\max_{\mathbf{a}'}Q_{\hat{\theta}}^k(\mathbf{s}',\mathbf{a}')-Q_\theta^k(\mathbf{s},\mathbf{a})\right]^2 \tag{1}$$

4. Limitations of Current Offline RL Methods

In this section, we briefly talk about the problem of applying naive online reinforcement learning methods to the static dataset. We also go through some of the previous offline methodologies to see the potential limitations. To illustrate the drawbacks of each problem more intuitively, we present some pictures that demonstrate the limitations associated with conservatism. Assume we have the following Q functions 1 at a certain state $\mathbf{s}$. The green points in Fig.1 describe the action data in the dataset.

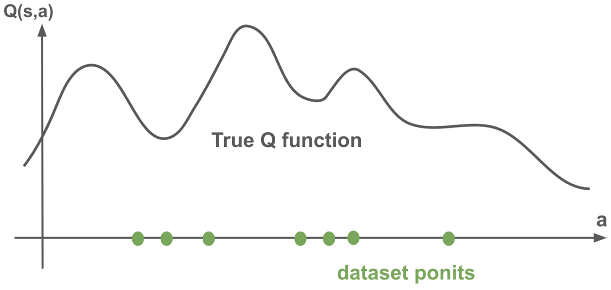

Figure 1. An example of a Q function at a certain state $\mathbf{s}$. Green points show actions in the dataset.

4.1. Naive Offline RL methods

We explore the application of Deep Q-Learning (DQL) in offline reinforcement learning scenarios. Deep neural networks have demonstrated strong performance in reinforcement learning tasks, particularly in handling intricate challenges. However, this method encounters limitations when applied to problems lacking direct interactions with the environment. The DQL updates are expressed as

$$\min_{Q_\theta^k} \mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[r(\mathbf{s},\mathbf{a})+\gamma\max_{\mathbf{a}'}Q_{\hat{\theta}}^k(\mathbf{s}',\mathbf{a}')-Q_\theta^k(\mathbf{s},\mathbf{a})\right]^2 \tag{2}$$

where $\mathbf{s}$ is state, $\mathbf{a}$ is current action, $\mathbf{s}'$ is next state and $\mathbf{a}'$ is next action. In this formulation, $\mathbf{s}$, $\mathbf{a}$, and $\mathbf{s}'$ are obtained from the dataset, but $\mathbf{a}'$ can exist beyond the dataset, known as an out-of-distribution (OOD) action.

The presence of an OOD action may lead to an erroneously large Q-value, consequently amplifying the target value for Q-learning. As a result, the estimated values tend towards an overly optimistic, and it cannot gain any useful information from the dataset. In the following sections, we explain some of the well-known methods to tackle these problems. In our example problem, the estimated Q value of DQL is described in Fig.2.

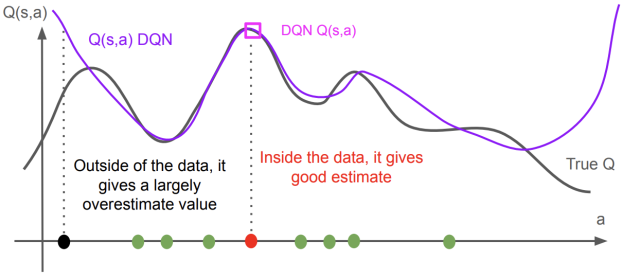

Figure 2. An example of a Q function of DQL at a certain state $\mathbf{s}$. As it shown, if the data is not in the dataset, and far from the data, the estimated value can be significantly larger than the true Q value.

4.2. Conservative Q-Learning

Conservative Q-Learning (CQL) is proposed by (Kumar et al., 2020), and it is one of the most widely used offline reinforcement learning methods. Instead of relying solely on Q-values, CQL introduces a conservative lower bound on Q-values to reduce overestimation. By doing so, it creates a more conservative estimate of the action values when it queries OOD actions. The mathematical formulation of CQL with KL-divergence can be expressed as

$$\min_{Q_\theta^k}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[r(\mathbf{s},\mathbf{a})+\gamma\max_{\mathbf{a}'}Q_{\hat{\theta}}^k(\mathbf{s}',\mathbf{a}')-Q_\theta^k(\mathbf{s},\mathbf{a})\right]^2 + \alpha\mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[\log\sum_{\mathbf{a}}\exp(Q_\theta(\mathbf{s},\mathbf{a}))-\mathbb{E}_{\mathbf{a}\sim\pi_\beta}[Q_\theta(\mathbf{s},\mathbf{a})]\right] \tag{3}$$

The initial term aligns with DQL but tempers Q-values through the inclusion of the second expectation term. The parameter α decides the conservativeness of Q values. Despite exhibiting notable efficacy in challenging RL tasks, CQL often yields excessively conservative Q-values. The generated Q values for the example problem is described in Fig.3. From this picture, we can tell that the estimated values are reliable at a data point if it is in the dataset. However, beyond the dataset boundaries, the estimated Q values are significantly lower than true Q values.

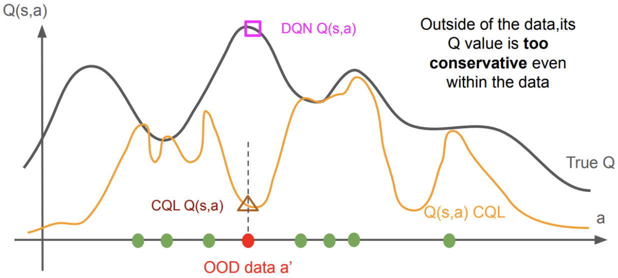

Figure 3. An example of a Q function of CQL at a certain state $\mathbf{s}$. CQL produces reliable Q values within the dataset. Yet, beyond this dataset, the estimated Q value may notably lean towards conservatism.

4.3. Implicit Q Learnig

On the contrary, Implicit Q-learning (Kostrikov et al., 2022) exclusively employs in-distribution data for updating Q-values. Unlike CQL and policy-constrained techniques, it restricts its queries solely to dataset information, effectively circumventing OOD issues. It uses expectile regression to fit the V-value and estimates Q values with it. The formulation of IQL can be expressed as:

$$\begin{aligned}&\min_{V_\phi^k}\mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathcal{D}}[L_2^\tau(Q_{\hat{\theta}}^k(\mathbf{s},\mathbf{a})-V_\phi^k(\mathbf{s}))]\\&\min_{Q_\theta^k}\mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}}[(r(\mathbf{s},\mathbf{a})+\gamma V_{\hat{\phi}}(\mathbf{s}')-Q_\theta^k(\mathbf{s},\mathbf{a}))^2]\end{aligned} \tag{4}$$

where $L_2^\tau(u)$ is an asymmetric squared loss used for expectile regression. This function is defined as

$$L_2^\tau(u)=|\tau-\mathbb{1}(u<0)|u^2 \tag{5}$$

Although it successfully avoids querying OOD actions, it still produces conservative Q-values due to its target value formulation using the dataset D

$$V(s)=\max_{a\in D}Q(s,a) \tag{6}$$

This implies that the Q-values are constrained to the maximum $Q(s,a)$ within the dataset, limiting their potential upper bounds. Consequently, this approach can exhibit excessive conservatism contingent upon the dataset characteristics. The IQL example is shown in Fig.4. As previously discussed, IQL uses maximum Q values within the dataset for the OOD action Q values. However, this trait can lead to conservative Q values and potentially degrade overall performance.

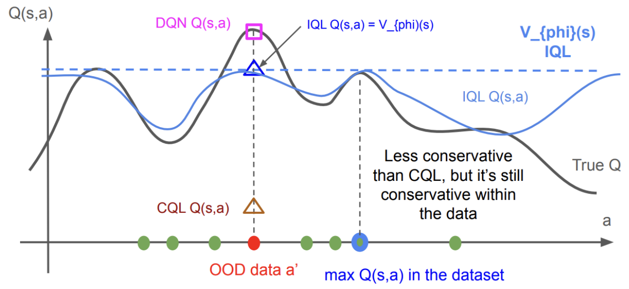

Figure 4. An example of a Q function of IQL at a certain state $\mathbf{s}$. While it displays reduced conservatism outside the dataset, its Q values may still tend to be lower than the true Q values.

5. Distance-Sensitive Offline Reinforcement Learning

In Section 4.1, we highlighted the potential challenges in major offline reinforcement learning paradigms. This section delves into our proposed method. The motivation behind our approach lies in the belief that if out-of-distribution (OOD) data is close to the dataset, the estimated Q value should be sufficiently accurate for reliable estimation. This premise finds support in prior research (Li et al., 2022), where it was observed that Deep Neural Networks (DNNs) can accurately estimate values within the convex hull and periphery of the dataset. Leveraging this insight, our work introduces a novel algorithm aimed at producing less conservative Q values where the spatial distance between any queried data point and data points sampled from the dataset to be sufficiently close.

5.1. Distance-Sensitive Implicit Q-Learning

Now, we aim to introduce a new offline RL method that can incorporate distance information from queried new data to the dataset. To achieve this goal, we propose a new approach called, Distance-sensitive Implicit Q-Learning (DIQL). This algorithm is based on IQL, and it also uses both state-value functions $V(s)$ and state-action $Q(s,a)$ functions. The formulation of DIQL can be written as:

$$\min_{V_\phi^k} \mathbb{E}_{(\mathbf{s},\mathbf{a})\sim\mathcal{D}}[L_2^\tau(Q_{\hat{\theta}}^k(\mathbf{s},\mathbf{a}) - V_\phi^k(\mathbf{s}))] \tag{7}$$

$$\min_{Q_\theta^k} \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}}[(r(\mathbf{s},\mathbf{a}) + \gamma\{V_{\hat{\phi}}(\mathbf{s}') + (1-\alpha(\mathbf{s}',\mathbf{a}_{max},\mathcal{D})) \\ A(\mathbf{s},\mathbf{a}_{max})\} - Q_\theta^k(\mathbf{s},\mathbf{a}))^2] \tag{8}$$

where $\mathbf{a}_{max}$ is defined as

$$\mathbf{a}_{max} = \underset{\mathbf{a}'}{\mathrm{argmax}} Q_{\hat{\theta}}^k(\mathbf{s}',\mathbf{a}') \tag{9}$$

and advantage $A(\mathbf{s},\mathbf{a}_{max})$ is

$$A(\mathbf{s},\mathbf{a}_{max}) = Q_{\hat{\theta}}^k(\mathbf{s}',\mathbf{a}_{max}) - V_{\hat{\phi}}(\mathbf{s}') \tag{10}$$

Note that $\alpha(\mathbf{s}',\mathbf{a}_{max},\mathcal{D})$ is defined as a function based on the distance from $(\mathbf{s}',\mathbf{a}_{max})$ to the dataset $\mathcal{D}$, and it takes value from 0 to 1. It decreases as the proximity of ($\mathbf{s}'$, $\mathbf{a}_{\mathrm{max}}$) to the dataset increases, while it amplifies with greater distance between ($\mathbf{s}'$, $\mathbf{a}_{\mathrm{max}}$) and the dataset. In section 4.2, we will further discuss the definition of such distance metric.

Similar to IQL, our proposed method comprises two updates: the value update and the Q-update. During the value update stage, $V_\phi^k(\mathbf{s})$ undergoes an update utilizing expectile regression, same as IQL. However, in the Q-update, we modified the objective function. Through the introduction of the advantage term $A(\mathbf{s},\mathbf{a}_{max})$, the Q-update mechanism can change target values based on the proximity of sampled maximum actions to the dataset. The underlying idea is that as the sampled action $\mathbf{a}_{max}$ approaches the dataset, the distance function $\alpha(\mathbf{s}',\mathbf{a}_{max},\mathcal{D})$ gets small, aligning its target value with that of DQN. Conversely, if the sampled action diverges from the dataset, $\alpha(\mathbf{s}',\mathbf{a}_{max},\mathcal{D})$ increases, resulting in a target value similar to that of IQL. The image of this new approach is described in Fig. 5.

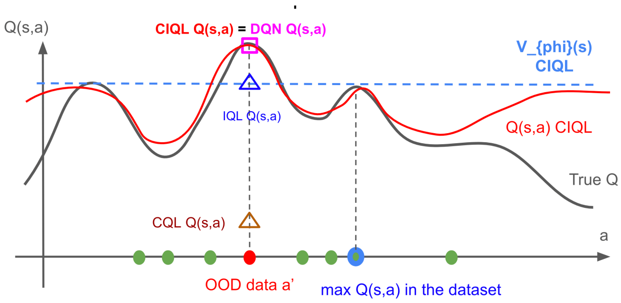

Figure 5. An example of a Q function of DIQL at a certain state $\mathbf{s}$. Contrary to IQL, it will not generate conservative values around the points in the dataset.

5.2. Distance approximation

In the previous section, we introduce the proposed algorithm DIQL. DIQL necessitates defining the function $\alpha(\mathbf{s}',\mathbf{a}_{\mathrm{max}},\mathcal{D})$ to precisely compute the distance between the queried data and the dataset. However, two significant challenges arise when determining the distance from the specific point ($\mathbf{s}'$, $\mathbf{a}_{\mathrm{max}}$) to the dataset. Firstly, defining distance becomes challenging when the state or action space is discrete. Additionally, in cases where the state is continuous and the action is discrete, defining this distance becomes even more complex. Secondly, computing the distance between the queried data point and every point in the dataset proves inefficient and computationally expensive. Performing such calculations for each queried data point and the dataset is impractical.

To address these challenges, we employ Random Network Distillation (RND) (Burda et al., 2019; Rezaeifar et al., 2021) as an approximation approach for determining the distance between the queried data point and the dataset. Within RND, two networks are employed: the prediction network and the fixed random neural network known as the target network. The prediction network is trained to predict the target network's output. The distance function $\alpha(\mathbf{s}', \mathbf{a}_{\max}, \mathcal{D})$ is determined by the prediction error of encoded features, ideally generating larger values when encountering unfamiliar input. Let $f_\theta(\mathbf{s}, \mathbf{a})$ be prediction network and $f_{\theta'}(\mathbf{s}, \mathbf{a})$ be target network. The approximated distance between queried point $(\mathbf{s}, \mathbf{a})$ and dataset $\mathcal{D}$ is compute as

$$\alpha(\mathbf{s}, \mathbf{a}, \mathcal{D}) = ||f_\theta(\mathbf{s}, \mathbf{a}) - f_{\theta'}(\mathbf{s}, \mathbf{a})||_2 \qquad (11)$$

Note that when we normalize the output of Eq.(11) when we train DIQL. Since it is impossible to normalize the data through all of the state-action pairs, we normalize α only within the sampled mini-batch dataset. The computation image is shown in Fig.6.

Before we train DIQL, we train the RND network with the dataset to learn the latent representations of the dataset. After that, we start training DIQL. The overall algorithm is shown in Alg.1.

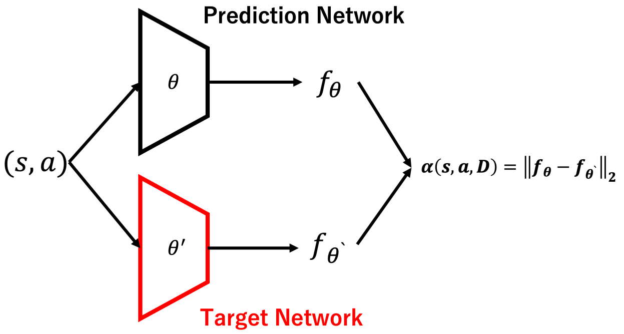

Figure 6. Approximated distance function with RND. We input $(\mathbf{s}, \mathbf{a})$ to both the prediction and the target network. We use the output difference as distance function $\alpha(\mathbf{s}, \mathbf{a}, \mathcal{D})$.

6. Experiment and Evaluations

To accurately assess the potency and generalizability of our algorithm, we employed our algorithm on a multitude of datasets and environments, spanning continuous and discrete state spaces. We employed this on both OpenAI's Gym environment as well as discrete maze environment.

Algorithm 1 DIQL

```
Input: Dataset $\mathcal{D}$
Step 1 train RND
for $i = 1$ to $iter_{rnd}$ do
    Sample Batch $(\mathbf{s}, \mathbf{a}) \sim \mathcal{D}$
    train RND with sampled batch
end for

Step2 train DIQL
for $i = 1$ to $iter_{diql}$ do
    Sample Batch $(\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s}') \sim \mathcal{D}$
    V-update Eq.(7)
    compute $a_{max}$ Eq.(9)
    compute distance of the queried point Eq.(11)
    Normalize $\alpha$ to 0 to 1 in the batch
    Q-update Eq.(8)
end for
```

6.1. Grid-World Environment

In this experiment, we compare DQL, CQL, IQL, and DIQL in a discrete map environment to see their estimated Q values. The goal here is to see the overestimation or underestimation of the previous methods, and show DIQL can estimate reasonable Q values. Our evaluation takes place in a 4×8 GridWorld, with environment specifics detailed in Fig.7. To derive the optimal value, we implement tabular Q-learning and utilize its estimated Q values as ground truth. Additionally, we generate a dataset by randomly sampling data from the environment, illustrated in Fig.8."

The 'Total Error' column in the table represents the sum of absolute Q-value errors across all state-action pairs, while 'Overestimation' indicates the average Q-value error among these pairs. Analysis of the table reveals that the proposed method exhibits the lowest error values. Although it shows slight overestimation compared to CQL, this overestimation is relatively modest.

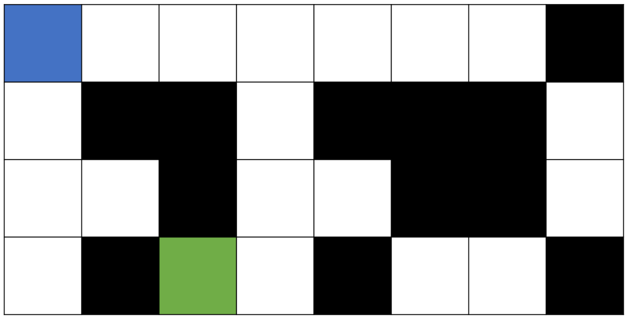

Figure 7. Illustration of 4×8 grid world. he top left corner marks the maze's starting point. When the agent reaches the green tile, it receives a reward, while black tiles represent walls.

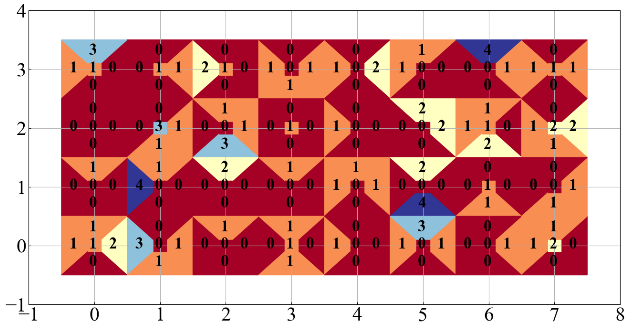

Figure 8. Illustration of dataset composition. We randomly sample the data from the environment and use it as dataset for offline Reinforcement Learning methods.

Table 1. Grid-world Q errors evaluation of each offline RL method

METHODOLOGY	TOTAL ERROR	OVERESTIMATION
DQL	1385.5	3.68
CQL	1254.1	-5.09
IQL	418.6	2.15
DIQL(OURS)	392.8	1.96

6.2. Continuous State Environment

Here, we test DIQL with OpenAI's Gym environment, and we specifically focus on continuous state environment. For this section, we also compare the proposed method with DQL, CQL and IQL. We use OpenAI Gym environment and test it with CartPole, Mountain Car, and Acrobot. For the dataset, we collect the data using DQL + RND method (Burda et al., 2019). Fig.9, Fig.10 and Fig.11 show the evaluation returns of all methods. Note that we plot the returns with average moving values.

For all of the results, CQL converges faster than the other approaches. However, the proposed method, DIQL, attains the same or higher reward compared to the other methods. Moreover, IQL is not as stable as its successor DIQL, especially for Mountain Car and Acrobot results.

7. Conclusion and Future Work

We presented Distance-sensitive Implicit Q-Learning (DIQL), a general algorithm for offline RL that leverages the distance from the queried data point to the dataset. Notably, this marks the pioneering integration of distance attributes into Q-learning-based Reinforcement Learning algorithms. The major advantage lies in its capability to produce less conservative Q values when the queried data point is close to the dataset. Additionally, the computationally inexpensive nature of the approximate distance calculation minimally impacts computational overhead. We verified that DIQL outperforms the other offline reinforcement learning methods in several tasks. There exist two potential future extensions for the proposed method. First, enhancing the precision of the distance function stands as a viable option. Finding more reasonable distance functions holds the promise of significantly augmenting the performance of DIQL. Second, broadening the application of the method to include continuous action spaces presents another potential avenue. Currently, DIQL is confined to discrete action environments due to the max operation in Q-update. However, leveraging an actor-critic method with DIQL harbors a robust prospect of extending its applicability to continuous action spaces.

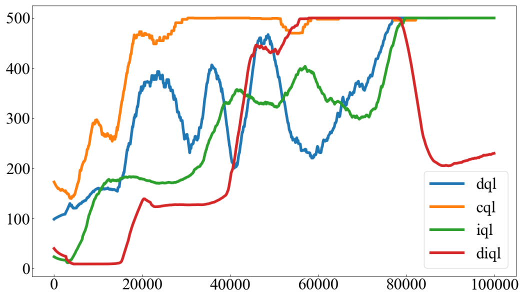

Figure 9. Evaluation return of each method in CartPole environment. The blue line is DQL, yellow is CQL, green is IQL, and red is DIQL. Here we plot the moving average returns.

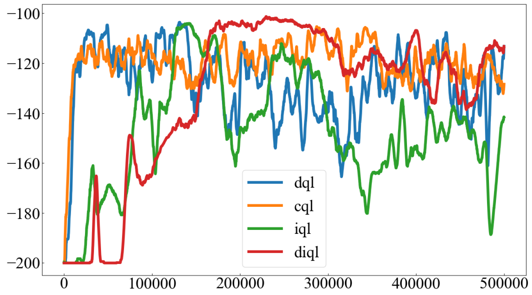

Figure 10. Evaluation return of each method in MountainCar environment. The blue line is DQL, yellow is CQL, green is IQL, and red is DIQL. Here we plot the moving average returns.

References

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?

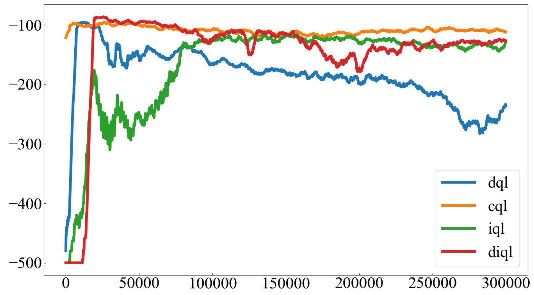

Figure 11. Evaluation return of each method in Acrobot environment. The blue line is DQL, yellow is CQL, green is IQL, and red is DIQL. Here we plot the moving average returns.

id=H1lJJnR5Ym.

Chitnis, R., Xu, Y., Hashemi, B., Lehnert, L., Dogan, U., Zhu, Z., and Delalleau, O. Iql-td-mpc: Implicit q-learning for hierarchical model predictive control, 2023.

Fort, S., Ren, J., and Lakshminarayanan, B. Exploring the limits of out-of-distribution detection. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 7068–7081. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/3941c4358616274ac2436eacf67fae05-Paper.pdf.

Garg, D., Hejna, J., Geist, M., and Ermon, S. Extreme q-learning: Maxent reinforcement learning without entropy. 2023. URL https://arxiv.org/abs/2301.02328.

Hansen-Estruch, P., Kostrikov, I., Janner, M., Kuba, J. G., and Levine, S. Idql: Implicit q-learning as an actor-critic method with diffusion policies, 2023.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, J., Zhan, X., Xu, H., Zhu, X., Liu, J., and Zhang, Y.-Q. When data geometry meets deep function: Generalizing offline reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://api.semanticscholar.org/CorpusID:256597815.

Lyu, J., Ma, X., Li, X., and Lu, Z. Mildly conservative q-learning for offline reinforcement learning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=VYYf6S67pQc.

Ma, Y. J., Jayaraman, D., and Bastani, O. Conservative offline distributional reinforcement learning. *CoRR*, abs/2107.06106, 2021. URL https://arxiv.org/abs/2107.06106.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets, 2021.

Rezaeifar, S., Dadashi, R., Vieillard, N., Hussenot, L., Bachem, O., Pietquin, O., and Geist, M. Offline reinforcement learning as anti-exploration. *ArXiv*, abs/2106.06431, 2021. URL https://api.semanticscholar.org/CorpusID:235417276.

Tajwar, F., Kumar, A., Xie, S. M., and Liang, P. No true state-of-the-art? ood detection methods are inconsistent across datasets, 2021.

Yang, J., Zhou, K., Li, Y., and Liu, Z. Generalized out-of-distribution detection: A survey, 2022.