Project Weekly Report 1

Boyang Xia 2023533073
Jiawen Dai 2023533132
Zhichen Zhong 2023533131

1 Weekly Progress Overview

This week, our team initiated the project by focusing on understanding and replicating the foundational concepts presented in the ICLR 2023 paper, "Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning" by Wang et al. Our primary accomplishment was the successful reproduction of the toy example detailed in Section 4 and part of the experiments detailed in Section 5 of the paper.

Building upon this, we extended the investigation by designing and implementing our own set of toy examples. This involved modifying the underlying data distributions and rewards of bandits from the original paper to create new multimodal scenarios. The objective of this extension was to further test and compare the performance of various methods, including the Diffusion-QL, under a broader range of complex policies and rewards.

2 Detailed Experimental Work

2.1 Reproduction of toy examples in Section 4 of the paper

In this section, we detail our work on reproducing the toy example presented in Section 4 of the paper "Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning".

2.1.1 Behavior Cloning

This initial experiment is designed to evaluate the capabilities of various policy representation methods, particularly in the context of a behavior-cloning task with multimodal data.

The experimental setup, as described in the paper, defines a simple bandit task. Actions, denoted by a, are real-valued and exist in a 2-dimensional space, such that $a \in [-1, 1]^2$. An offline dataset, $\mathcal{D} = \{(a_j)\}_{j=1}^{M}$, is constructed using $M = 10,000$ action examples. These actions are generated by sampling from an equal mixture of four distinct Gaussian distributions. The centers (means) μ for these Gaussian distributions are specified as:

$$\mu \in \{(0.8, 0.8), (0.8, -0.8), (-0.8, -0.8), (-0.8, 0.8)\}$$

The standard deviation for each Gaussian distribution, along both dimensions, is set to $\sigma_d = (0.05, 0.05)$. This configuration results in a multimodal behavior policy.

Our primary objective in reproducing this setup is to observe and verify how effectively different policy models, with a particular focus on the diffusion-based approach, can capture and represent the four density modes of this behavior policy. This serves as a foundational step for understanding the expressiveness of diffusion policies compared to other established methods when dealing with complex, multimodal action distributions commonly found in offline datasets.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

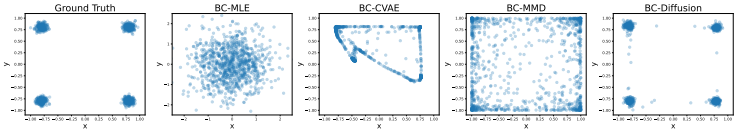

Figure 1: Equal Gaussian Mixture Distribution: BC

2.1.2 Policy Improvement

Following the behavior cloning evaluation, Section 4 of the paper extends the toy example to investigate policy improvement within an offline reinforcement learning (RL) framework. This phase utilizes the same offline dataset $\mathcal{D}$ consisting of $10,000$ actions generated from the four Gaussian modes previously described. The state s for this bandit task can be considered fixed or uniform, with the learning focused on the action policy $\pi(a|s)$ and value function $Q(s,a)$.

To enable policy learning, a reward mechanism is introduced. Each action a_j in the dataset is assigned a reward r_j. According to the paper, these rewards are sampled from a Gaussian distribution. The standard deviation of this reward distribution is fixed at 0.5. The mean of the reward for any given action is determined by its relation to a specific target "data center". As illustrated in the second row of Figure 1 in the original paper (which depicts the reward landscape), this reward structure is intentionally designed to make one of the four modes optimal. Specifically, the mode corresponding to actions around the center $(0.8, -0.8)$ is associated with higher expected rewards. This setup mimics an offline RL scenario where the true underlying reward function is unknown to the agent and must be inferred from the static dataset.

In this policy improvement phase, the performance of Diffusion Q-Learning (Diffusion-QL) is benchmarked against several prior offline RL algorithms: TD3+BC, Behavior Cloning Q-learning (BCQ), and BEAR-MMD. The paper states that all methods are trained for 1000 epochs to ensure convergence.

Our objective in reproducing this part of the experiment is to assess whether Diffusion-QL, by leveraging its expressive diffusion policy and Q-learning guidance, can effectively identify and converge to the high-reward region (the optimal mode). This contrasts with the expected performance of other methods, which, as the paper suggests, may be constrained by their policy regularization schemes or limited policy expressiveness, leading to suboptimal solutions or failure to converge to the true optimal mode. The original paper reports that Diffusion-QL successfully converges to this optimal region.

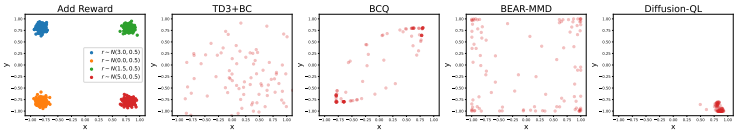

Figure 2: Equal Gaussian Mixture Distribution: QL

2.1.3 Investigating the Impact of Diffusion Steps N

A crucial hyperparameter in the Diffusion Q-Learning (Diffusion-QL) algorithm is the number of diffusion timesteps, denoted by N. This parameter determines the granularity of the diffusion process and significantly influences the model's behavior, policy expressiveness, and computational cost. Section 4 of the original paper (specifically Figure 2 and its accompanying discussion) investigates the effects of varying the value of N.

In our reproduction efforts, we also aimed to investigate the sensitivity of Diffusion-QL (or BC-Diffusion) to the parameter N. We conducted experiments where we varied N across values: $N \in \{2, 5, 10, 50\}$, utilizing the same experimental setting detailed in Sections 2.1.1 and 2.1.2. We then observed its effects on the performance of BC-Diffusion and Diffusion-QL.

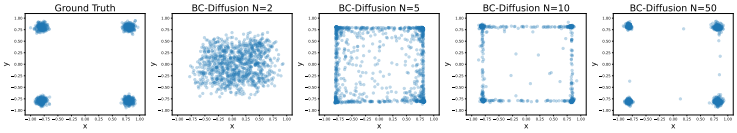

Figure 3: Modify N: BC

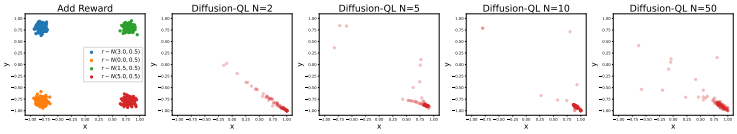

Figure 4: Modify N: QL

2.2 Extended Investigation: Other Toy Bandit Examples

Building upon our reproduction of the toy examples presented in the original paper, we further extended our investigation by designing and implementing three distinct sets of custom toy bandit examples. The primary motivation for this extension was to evaluate the robustness, adaptability, and potential limitations of Diffusion Q-Learning (and possibly other baseline methods) under scenarios that introduce increased complexity in both the underlying action distributions of the behavior policy and the associated reward landscapes.

2.2.1 Mixture Gaussian with Increased Modal Complexity

This first new toy example was designed to significantly increase the complexity of the action distribution compared to the original paper's four-mode bandit task. The goal was to test the algorithms' ability to handle a much larger number of modes with varying spatial arrangements and densities. The actions a remain in a 2D space, typically clipped to $a \in [-1, 1]^2$. For our experiments, we generated an offline dataset $\mathcal{D}_1$ consisting of $M_1 = 10,000$ action samples.

Action Distribution Design The action distribution in this example is a mixture of Gaussian modes sourced from three distinct geometric structures:

- **Inner Ring Modes**: A set of $N_{inner} = 16$ Gaussian modes are equally spaced on an inner circle of radius $r_{inner} = 0.5$. Each mode in this ring has a standard deviation of $\sigma_{ring} = 0.015$. Approximately 35% of the total action samples are drawn from these inner ring modes.
- **Outer Ring Modes**: A second set of $N_{outer} = 24$ Gaussian modes are equally spaced on an outer circle of radius $r_{outer} = 0.75$. These modes also have a standard deviation of $\sigma_{ring} = 0.015$ and are angularly offset relative to the inner ring modes. Approximately 45% of the total samples originate from this outer ring.
- **Corner Modes**: Finally, $N_{corner} = 4$ Gaussian modes are placed at the corners of the action space, specifically at coordinates $(\pm 0.9, \pm 0.9)$. These corner modes have a larger standard deviation of $\sigma_{corner} = 0.03$. The remaining approximately 20% of samples are drawn from these corner modes.

In total, this configuration results in $N_{total} = N_{inner} + N_{outer} + N_{corner} = 16 + 24 + 4 = 44$ distinct Gaussian modes. The samples from each mode are clipped to ensure they stay within the $[-1, 1]^2$ action space. The state s for this bandit task is considered fixed (e.g., all zeros).

Reward Function Design The reward function $R_1(a)$ for this complex action distribution was designed to be a continuous landscape formed by the sum of multiple 2D Gaussian peaks, rather than being tied to specific modes directly. This creates a scenario with multiple local optima of varying magnitudes and spreads. The primary Gaussian peaks contributing to the reward are centered at:

- Peak 1: Center $(0.8, -0.8)$, Amplitude ≈ 6.0, Standard Deviations $(\sigma_x, \sigma_y) = (0.15, 0.15)$. This forms the dominant high-reward region.
- Peak 2: Center $(0.0, r_{outer} = 0.75)$, Amplitude ≈ 3.5, Standard Deviations $(0.2, 0.2)$.
- Peak 3: Center $(-r_{inner} = -0.5, 0.0)$, Amplitude ≈ 2.0, Standard Deviations $(0.15, 0.15)$.
- Peak 4: Center $(-0.7, 0.7)$, Amplitude ≈ 1.0, Standard Deviations $(0.2, 0.2)$.

The final reward assigned to an action a is the sum of contributions from these peaks, plus a small amount of Gaussian noise with a standard deviation of $\sigma_{reward\_noise} = 0.1$. This reward structure is intentionally complex and potentially misaligned with many of the action modes from the behavior policy, providing a challenging test for policy improvement algorithms to identify and exploit the true high-reward regions. The ground truth data, when visualized, often involves coloring actions by their true reward value to illustrate this landscape.

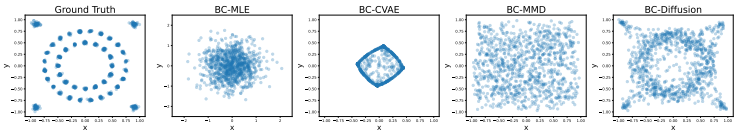

Figure 5: Complex distribution 1:Increased Modal Complexity

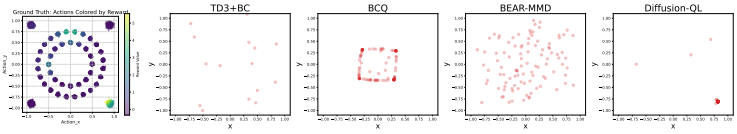

Figure 6: Complex distribution 1: Increased Modal Complexity

- Performance of BC: The policy of BC-MLE is limited to a single mode. The CVAE model even fails to exhibit mode-covering behavior, excluding the corner samples. The Tanh-Gaussian policy optimized under MMD fails to capture the true distribution. Diffusion seems to be the best among the four, but it fails to distinguish between the inner ring and outer ring.
- Performance of QL: As expected, TD3+BC, BCQ, and BEAR-MMD fail to converge to high reward regions since they cannot capture the true policy distribution. Diffusion QL, however, performs well with the guidance of Q-Learning, even though it fails to accurately capture the ground truth.

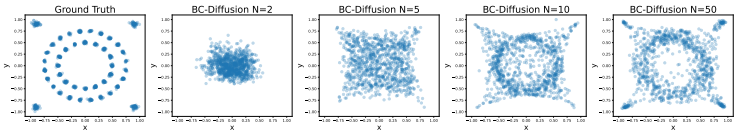

Figure 7: Complex distribution 1: Modify N BC

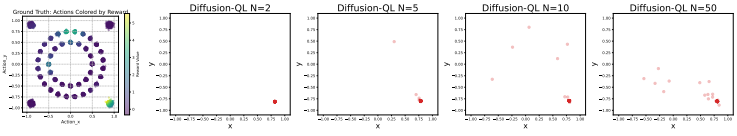

Figure 8: Complex distribution 1: Modify N QL

The influence of N:

- **N=2**: Generated points exhibit dispersed Gaussian distribution, failing to capture the four corner clusters and dual-ring structure of the true distribution.
- **N=5**: Find part of the distribution. Initial ring structures emerge, but corner clusters remain underdeveloped compared to ground truth.
- As N increases, the scattered points have decreased, and more data points are increasingly concentrated in the original distribution locations. And a complex policy distribution takes bigger N to converge.
- **N=50**: Generated samples best approximate the Ground Truth.
- The experiments validate the *positive correlation* between the number of diffusion timesteps (N) and the model's expressive capacity, where more steps enable finer-grained distribution learning through:
  - Extended reverse diffusion chain for gradual distribution refinement
  - Enhanced noise scheduling precision (β_t interpolation)
  - Improved gradient flow through deeper temporal unfolding

2.2.2 Continuous Spiral Action Distribution with Radial Sinusoidal Rewards

This second new toy example introduces a different type of complexity, featuring a continuous, spiral-like action distribution and a reward function that depends solely on the magnitude (radius) of the action, creating concentric rings of varying reward. The actions a are in a 2D space and naturally fall within the unit circle, $a \in [-1, 1]^2$. For our experiments with this setup, we generated an offline dataset $\mathcal{D}_2$ consisting of $M_2 = 10,000$ action samples.

Action Distribution Design The action distribution is generated as follows:

- A radial component R is sampled uniformly from the interval $[0, 1)$, i.e., $R \sim U(0, 1)$.
- An angular component θ_{actual} is determined as a function of R: $\theta_{\text{actual}} = k_R \cdot R$, where the coefficient k_R is set to 6π by default.
- The 2D action $a = (x, y)$ is then derived using standard polar to Cartesian conversion:

$$x = R\cos(\theta_{\text{actual}}) = R\cos(k_R R)$$
$$y = R\sin(\theta_{\text{actual}}) = R\sin(k_R R)$$

This generation process results in a continuous action distribution forming a spiral pattern, where the angle of an action is coupled with its radius. As R increases, the action completes multiple rotations (specifically, 3 full rotations as R goes from 0 to 1 with $k_R = 6\pi$). The density of points is inherently higher towards the center ($R \approx 0$). The state s for this bandit task is considered fixed (e.g., all zeros).

Reward Function Design The reward function $R_2(a)$ in this example is designed to be dependent only on the radial distance R from the origin (where $R = \sqrt{x^2 + y^2}$ for an action $a = (x, y)$). It is defined by a sinusoidal function:

$$R_2(a) = A\sin(\omega_R R + \phi) + C$$

The default parameters for this function are:

- Amplitude $A = 0.5$
- Angular frequency on radius $\omega_R = 10\pi$
- Phase $\phi = 0.0$
- Offset $C = 0.5$

This reward structure creates multiple concentric rings of high and low rewards as R varies from 0 to 1 (specifically, 5 cycles of the sine wave since $\omega_R = 10\pi$). A key characteristic is that the reward is independent of the action's angle θ_{actual}. This setup challenges the learning algorithms to discern the radial dependency of the reward, potentially averaging over angles if the policy representation is not sufficiently expressive or if the Q-function learning is not precise. No explicit noise is added

to this reward function in its generation. The ground truth data, when visualized (as in ‘axs[0]‘ of the provided script), typically shows actions colored by their reward value, clearly illustrating these concentric reward bands overlaid on the spiral action distribution.

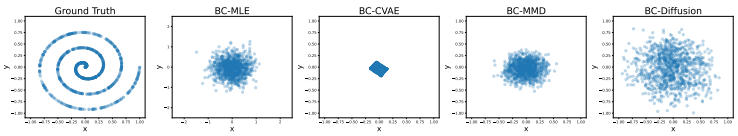

Figure 9: Complex distribution 2: Mosquito coil shape

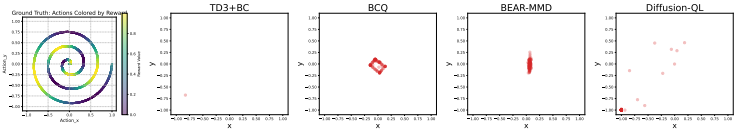

Figure 10: Complex distribution 2: Mosquito coil shape

- Performance of BC: the data points of MLE and MMD are scattered but irregular, and there is a significant difference from the true spiral distribution. The data of CAVE is more likely to be concentrated, while the diffusion method is more dispersive, which almost cover the shape of the origin distribution.
- Performance of QL: all four methods behavior poorly, pointing out their limit on complex distribution. Poor performance may also come from unsuitable hyper-parameters, which is also the direction we will investigate.

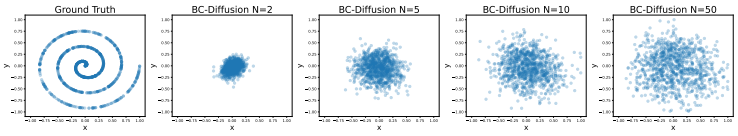

Figure 11: Complex distribution 2: Modify N BC

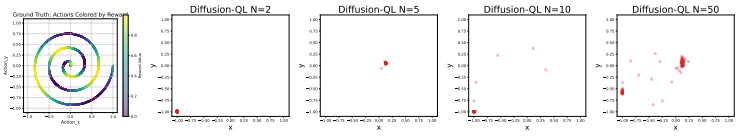

Figure 12: Complex distribution 2: Modify N QL

- For complex policy, simply increasing the size of N does not lead to a significant improvement in performance.

2.2.3 Action Distribution Forming ‘SI252’ with Targeted Rewards

This third custom toy example introduces a highly structured action distribution where data points form a sequence of characters, specifically “SI252". The reward function is designed to be highly targeted, with a specific small region within one of the characters yielding significantly higher rewards than the rest of the action space. This setup aims to test an algorithm’s ability to explore and identify a sparse, high-reward region within a complex, multi-modal behavior policy. The actions a are in a 2D space, generally clipped to $a \in [-1, 1]^2$. The offline dataset $\mathcal{D}_3$ for this experiment consists of $M_3 = 10,000$ action samples.

Action Distribution Design The action distribution is constructed by arranging a sequence of five characters, 'S', 'I', '2', '5', '2', horizontally across the 2D action space. This process results in a complex, highly structured multi-modal distribution where the modes collectively form the visual pattern "SI252".

Reward Function Design The reward function $R_3(a)$ for this "SI252" action distribution is specifically designed to be targeted, creating a "needle in a haystack" problem:

- **High-Reward Region**: A small, specific set of modes within the central 'I' character (specifically, the two middle points of its vertical stem) are designated as high-reward modes. Samples generated from these specific modes are assigned rewards drawn from a Gaussian distribution with a mean of $\mu_{\text{high}} = 5.0$.
- **Low-Reward Regions**: All other modes that constitute the characters 'S', the remaining parts of 'I', '2', and '5' are designated as low-reward modes. Samples from these modes are assigned rewards drawn from a Gaussian distribution whose mean is randomly chosen from the set $\{1.0, 2.0, 3.0\}$.
- **Reward Standard Deviation**: For all rewards, whether from high or low-mean modes, the standard deviation for sampling the actual reward value is fixed at $\sigma_{\text{reward}} = 0.5$.

This reward structure ensures that most of the action space yields relatively low and somewhat variable rewards, while only a very precise sub-region (middle of the 'I') offers a significantly higher reward. This challenges algorithms to effectively explore the detailed structure of the behavior policy and identify the sparse high-reward signals without getting stuck in regions that are frequently visited by the behavior policy but offer suboptimal rewards. The ground truth visualization typically colors actions by their reward, highlighting this sparse reward characteristic.

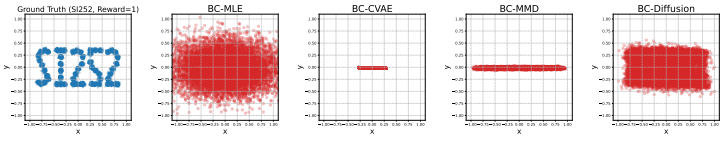

Figure 13: Complex distribution 3: clusters forming SI252

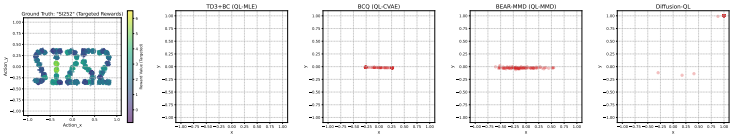

Figure 14: Complex distribution 3: clusters forming SI252

- Performance of BC: The data points are scattered and irregular.
- Performance of QL: The action distribution is almost invisible, indicating that the model fails to effectively explore or generate meaningful actions.

2.3 Reproduction of experiments in Section 5 of the paper

This section reproduced the Diffusion QL algorithm in a classic RL environment in order to verify its effectiveness. Choosing walker2d-medium-expert-v2, in which the agent controls a 2D bipedal walker with multiple joints is demanded to move straight forward without excessive energy use and falling, the algorithm stopped after 1450 epochs with the loss curves showcased below. The declining BC and Actor loss curve with decreasing slopes indicates that the algorithm can effectively clone a successful behavior, the QL loss, finally fluctuated around 4e-5, indicates that the Q-network could produce stable predictions about the state-action rewards, while the single peak critic loss reflects the critic network's ability to better approximate the true Q-values with self-correction, demonstrating its capability in complex non-bandit problem settings.

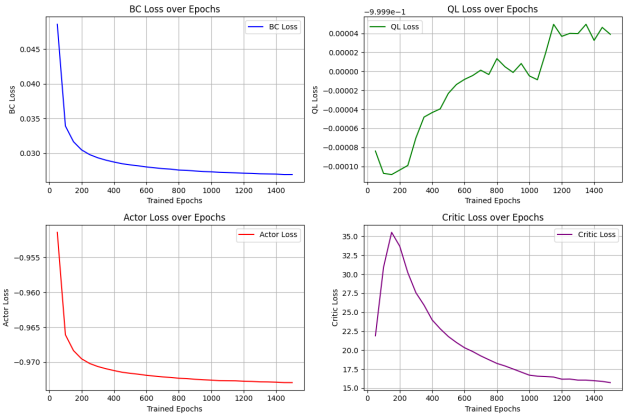

Figure 15: Enter Caption