Enhancing Diffusion Policies with Distribution-Matching Generator in Offline Reinforcement Learning

Xuemin Hu[1,2*], **Shen Li**[1,3*], **Yingfen Xu**[1], **Bo Tang**[4], **Long Chen**[5†]

[1]Hubei University, Wuhan, Hubei, China
[2]Key Laboratory of Intelligent Sensing System and Security (Hubei University), Ministry of Education, Wuhan, Hubei, China
[3]Tongji University, Shanghai, China
[4]Worcester Polytechnic Institute, Worcester, MA, USA
[5]Institute of Automation, Chinese Academy of Sciences, Beijing, China
long.chen@ia.ac.cn

Abstract

Offline reinforcement learning (RL) can learn policies from pre-collected offline datasets without interacting with the environment, but it suffers from the issue of out-of-distribution (OOD). Recent methods use the generative adversarial paradigm to learn policies, but easily fail to handle the conflict of fooling the discriminator and maximizing expected returns. In this paper, we propose a novel offline RL method named Distribution-Matching Generator-based Diffusion Policies (DMGDP). A distribution matching-based policy learning method is first developed, where the diffusion serves as the policy generator, to handle the conflict of fooling the discriminator and maximizing expected returns. Furthermore, a policy confidence mechanism based on discriminator regularization is designed to prevent the agent from taking OOD actions, with the aim of robust generative adversarial learning. We conducted extensive experiments on the D4RL benchmarks, and the results demonstrate that DMGDP outperforms state-of-the-art methods.

Introduction

Offline reinforcement learning attracts significant attention from researchers due to its ability to learn from pre-collected datasets without interacting with the environment. This approach can effectively decrease the risk and cost of training an agent in the fields of autonomous driving (Hu et al. 2023, 2024), medical treatment (Liu et al. 2020), and robotics (Kober, Bagnell, and Peters 2013), etc. However, unlike imitation learning (Hu et al. 2021) and traditional online RL methods, offline RL methods train the agent using offline datasets collected by behavior policies, and the datasets cannot often cover the entire state-action space. The out-of-distribution (OOD) data increases the cumulative error of value function estimation, which is the OOD issue, leading to poor performance of the learned policy (Fujimoto, Meger, and Precup 2019; Kumar et al. 2019).

A naive approach to solve the OOD issue is constraining the deviation between the learned policy and the behavior policy (Fujimoto and Gu 2021; Wang, Hunt, and Zhou 2023; Zhang et al. 2024). However, due to the difficulty in modeling behavior policies, these methods often generate suboptimal policies. One of the main reasons is that existing methods fail to represent the distribution of multimodal actions, especially in environments where multiple actions could be selected to maximize the expected return for the same state (Wang, Hunt, and Zhou 2023). This limitation weakens the expressiveness of the learned policy and the ability to align with the true behavior distribution, leading to conservative or suboptimal policies (Chi et al. 2023; Ada et al. 2024).

To address this issue, recent studies utilize deep generative models such as generative adversarial networks (GANs) (Goodfellow et al. 2014; Grover, Dhar, and Ermon 2018) and

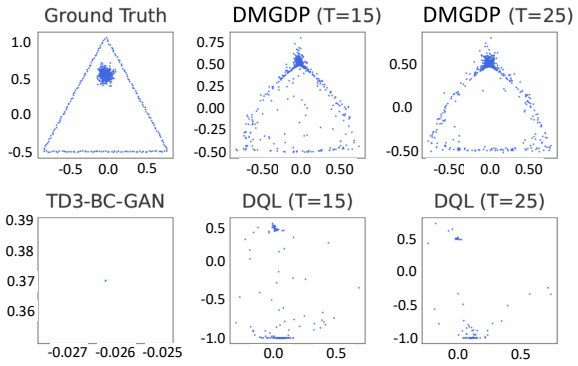

Figure 1: Offline RL experiment on a simple task with sparse rewards, where the triangle contains more data points with lower rewards, while the central region has fewer points but offers higher rewards. The ground truth shows the actions taken by the behavior policy. Other plots show the actions collected by DMGDP and DQL. The subplot titled "TD3-BC-GAN" is obtained by replacing the diffusion model in DMGDP with a simple MLP. Except DMGDP, other methods fail to represent the policy distribution, which denotes that policy expressivity is crucial in offline RL and the constraint provided by GAN framework is also important for diffusion policy. Details can be found in Appendix A.6.

*These authors contributed equally.
†Corresponding author.

diffusion models (Ho, Jain, and Abbeel 2020). Some methods provide a solution that combines the GAN and actor-critic (AC) framework to constrain the learned policy by training a discriminator to identify OOD data (Vuong et al. 2022; Wu et al. 2024; Liu and Hofert 2024). The discriminator learns through a supervised objective and can easily identify samples in the state-action space, but the multi-layer perceptron (MLP)-based generator in existing methods struggles to generate actions that both fool the discriminator and maximize expected returns, leading to suboptimal policies and unstable training processes. As shown in the subplot titled "TD3-BC-GAN" in Figure 1, a simple MLP-based generator cannot model the action in the designed task with sparse rewards, which means it cannot generate expressive policies to support the GAN-AC framework. Some recent researches (Wang, Hunt, and Zhou 2023; Chen et al. 2023a) have proved that introducing the diffusion into offline RL methods can greatly improve the expressiveness of the learned policy, but their performance is limited by the behavior policy and offline datasets, and the learning process is unstable for OOD data since they only rely on the diffusion objective similar to behavior clone (BC) to adjust the Q-function estimation (Levine et al. 2020; Ada et al. 2024). As shown in Figure 1, the results of Diffusion Q-learning (DQL) (Wang, Hunt, and Zhou 2023) in the designed sparse reward task demonstrate that DQL cannot effectively model the initial behavior policy and learn the action distribution with sparse rewards.

In this paper, we propose a novel method named Distribution-Matching Generator-based Diffusion Policies (DMGDP) for offline reinforcement learning. We focus on the problems of maximizing the expected returns while fooling the discriminator in the GAN-AC framework, so as to handle the OOD issue. The core idea is to leverage the high expressiveness of diffusion policies and the regularization capabilities of generative adversarial framework to enable stable and efficient policy learning in offline RL. This paper offers the following three contributions: 1) A generator-based diffusion policy learning method, which is derived from the theoretical formalization of distribution matching, is proposed to effectively maximize expected returns while fooling the discriminator. 2) We develop a policy confidence mechanism based on the discriminator regularization, which quantifies the probability of distribution shift, to regularize the Q-function and constrain the diffusion policy. In this process, we also design a sequential stacking input technique to improve training stability. 3) We conducted comprehensive experiments on the D4RL, and the results demonstrate that the proposed method outperforms the state-of-the-art methods on the benchmark.

Related Work

Offline reinforcement learning The extrapolation error caused by OOD data is an essential and challenging problem in offline RL. Most of existing offline RL methods handle this issue by incorporating an additional policy regularization term that includes explicitly divergence constraints (Fujimoto, Meger, and Precup 2019; Fujimoto and Gu 2021; Xu et al. 2021) and implicit behavior weightings (Ashvin et al. 2020; Xu et al. 2022). Some methods enforce pessimism over OOD data by applying regularization in the form of value function (Kumar et al. 2020; Xu et al. 2023; Lyu et al. 2022). Other methods utilize uncertainty estimation (An et al. 2021) or distance function (Li et al. 2022) to constrain the influence of different factors during policy learning. Unlike existing policy regularization approaches that mainly regularize policies by approximating the deviation degree, our method uses more powerful generative models to represent the policy and regularizes policies via both the diffusion loss term and the discriminator output in the GAN.

Generative adversarial networks in offline RL Some recent methods utilize GANs for policy learning in the actor-critic framework. Vuong et al. (Vuong et al. 2022) propose an offline RL method named DASCO with dual generators to remove the tension between maximizing the expected return and matching the data distribution by generating the mixed data distribution. Wu et al. (Wu et al. 2024) use CGAN and uncertainty estimation to construct a generative distribution that effectively alleviates the conflict between reward maximization and strict data distribution constraints. Liu et al. (Liu and Hofert 2024) leverages GANs to align the action space between the learned and behavior policies. A significant limitation of these methods is that they ignore the generator's expressiveness for complex policy tasks. Unlike these methods, we propose to combine maximum likelihood estimation (MLE) with GANs and formulate the diffusion as the generator in GANs.

Diffusion models in offline RL Diffusion models have gained considerable attention in offline RL due to their impressive generative capacities and have been used to model trajectories (Janner et al. 2022; Ni et al. 2023; Ajay et al. 2022), expand datasets (Chen et al. 2023b; Lu et al. 2024), and generate policies (Wang, Hunt, and Zhou 2023; Ada et al. 2024; Lu et al. 2023; Venkatraman et al. 2024), etc. Wang propose a method named Diffusion Q learning (DQL) (Wang, Hunt, and Zhou 2023), where the diffusion is used as an actor to learn policies with high expressiveness, and then the policy is regularized to strengthen the distribution constraint. Ada (Ada et al. 2024) reconstructs the state for diffusion policies to alleviate the distribution shift. Lu (Lu et al. 2023) implements diffusion policies by incorporating an energy function to guide the policy learning process. The work most similar to ours is DQL (Wang, Hunt, and Zhou 2023). Unlike previous work, we introduce discriminator-based regularization to constrain diffusion policy and reduce overestimation of the value function in OOD actions.

Preliminaries

Offline reinforcement learning Reinforcement learning is often described as a Markov decision process (MDP) with the tuple $M = \{S, A, P, R, \gamma\}$, where $S \in \mathbb{R}^N$, $A \in \mathbb{R}^M$, $P\colon S \times A \to S$, $R : S \times A \to \mathbb{R}$, and $\gamma \in (0, 1]$ represent the state space, action space, transition dynamics, reward function, and discount factor, respectively. The behavior of an agent in RL is determined by the policy π, and the goal in RL is to learn an optimal policy $\pi(a|s)$ to maximize the expected discounted return $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$, where s_t and a_t are the state and action at the time step t, respectively. Offline

RL algorithms learn a policy from an offline dataset $\mathcal{D} \triangleq \{(s, a, r, s')\}$, which is collected by the unknown behavior policy π_β (Fujimoto, Meger, and Precup 2019). s, a, r, and s' represent a state, the action performed on the state, the obtained reward from this action, and the next state after performing this action, respectively.

Generative adversarial networks Generative adversarial networks (Goodfellow et al. 2014) model data distributions through an adversarial game between a generator G and a discriminator D. The training objective $V(G, D)$ is denoted as a maximization and minimization (minimax) game between G and D, as shown by Eq. 1.

$$V(G,D) = \min_G \max_D \mathbb{E}_{x\sim P_{data}}[\log(D(x))] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))], \quad (1)$$

where D is trained to maximize the probability of correctly classifying real data and fake ones from G, and P_{data} is the true data distribution. The minimax problem in GANs can be converted into an optimization problem for the generator G: $\min_G JS(P_X(\cdot)||P_G(\cdot))$, where $JS(\cdot)$ denotes the Jensen-Shannon divergence (Menéndez et al. 1997). The essence of the formula is that G is trained to generate data that match the distribution of real data.

Diffusion models Diffusion models (Ho, Jain, and Abbeel 2020) are latent variable generative models that consist of a forward noising process and a backward denoising process. Diffusion models fix the approximate posterior $q(x_{1:T}|x_0)$, known as the forward process or diffusion process, to a Markov chain that gradually adds Gaussian noise to the data in T steps with a variance schedule $\beta_i \in \{\beta_1, ..., \beta_T\}$.

The joint distribution $p_\theta(x_{0:T})$ is commonly referred to as the *reverse process*, which is defined as a Markov chain with learned Gaussian transitions starting at $p(x_T) = \mathcal{N}(x_T; \mathbf{0}, \mathbf{I})$, as shown by Eq. 2.

$$\begin{aligned} p_\theta(x_{0:T}) &:= p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \\ p_\theta(x_{t-1}|x_t) &:= \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \end{aligned} \quad (2)$$

where μ_θ and Σ_θ represent the predicted mean and variance, respectively. The training objective of the *reverse process* is to maximize the evidence lower bound defined as $\mathbb{E}_q[\ln \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}]$. Based on the Denoising Diffusion Probabilistic Model (DDPM) (Ho, Jain, and Abbeel 2020), we approximate the Evidence Lower Bound (ELBO) by using a simplified surrogate loss function $L_d(\theta)$, as shown by Eq. 3.

$$L_d(\theta) = \mathbb{E}_{i\sim[1,T],\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),x_0\sim q}\left[||\epsilon - \epsilon_\theta(x_i, i)||^2\right], \quad (3)$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ denotes the Gaussian noise added in the forward process, and ϵ_θ denotes the parametric model.

Methodology

The discriminator can provide the probability of distribution shift to address the OOD issue (Vuong et al. 2022), but the method of introducing GAN into RL faces the challenge of how to maximize expected returns while fooling the discriminator. To address this challenge, the DMGDP is proposed in this paper. We proposed the generative adversarial learning method based on distribution matching of the learned policy and the behavior policy, where the generator is constructed by the diffusion. Then we explain the discriminator-based policy confidence mechanism that clarifies the role of the discriminator in the distribution matching-based diffusion policy method. In this process, we also design a sequential stacking input technique to improve training stability of the learning process. Finally, we describe the method as a complete offline RL algorithm.

Distribution Matching-based Diffusion Policy Generation

In this section, we introduce the generative adversarial method combined with Maximum Likelihood Estimation (MLE) to enhance the distribution matching between the learned policy and the behavior policy, where we transfer distribution matching into the objective of the Q-function and prove its convergence by constructing a distribution matching operator on the offline RL framework. In addition, we approximate the MLE term using an MSE-like diffusion loss to improve the expressive capability of the policy.

Distribution matching: GANs provide a simple and attractive solution to address the distribution shift problem in offline RL, with the discriminator providing a probability that directly quantifies the extent of distributional shift (Vuong et al. 2022). Besides, GANs can be used to improve policy performance through adversarial learning. However, a challenge is that the learned policy cannot fit the data distribution well since vanilla GANs often fail to model the behavior policy (Zhang et al. 2024). To address this issue, one needs to perform exact likelihood evaluation of the learned policy compared to the behavior policy, which helps to enhance the distribution-matching capability of the learned policy, thus enabling the learned policy to effectively fit the distribution of the behavior policy. Maximum likelihood estimation, which tries to minimize the Kullback-Leibler (KL) divergence between the distributions of the learned policy P_θ and the behavior policy P_{data} (Grover, Dhar, and Ermon 2018), provides an effective method to realize this scheme, as shown by Eq. 4.

$$\min_\theta KL\left(P_{\text{data}}, P_\theta\right) = \mathbb{E}_{\tau\sim P_{\text{data}}}\left[\log\frac{p_{\text{data}}(\tau)}{p_\theta(\tau)}\right], \quad (4)$$

where τ is a trajectory which consists of (s, a) pairs. $p_{data}(\tau)$ and $p_\theta(\tau)$ are the probability densities of P_{data} and P_θ, respectively. Since p_{data} is independent of θ, the above optimization issue can be equivalently formulated by Eq. 5.

$$\max_\theta \mathbb{E}_{\tau\sim P_{\text{data}}}\left[\log p_\theta(\tau)\right]. \quad (5)$$

Therefore, we introduce Eq. 5 into GAN and extend it to the state-action space for policy generation, to improve the learned policy's ability for matching the distribution of the behavior policy, as shown by Eq. 6.

$$V(G_\theta, D_\phi) = \min_\theta \max_\phi \mathcal{L}_G(G_\theta, D_\phi) \\ -\lambda \mathbb{E}_{\tau \sim P_{data}}[\log(p_\theta(\tau))], \tag{6}$$

where G_θ and D_ϕ denote the networks of the generator and the discriminator, respectively. The term $\mathcal{L}_G(G_\theta, D_\phi)$ is the min-max optimization objective of GANs. The second term $\mathbb{E}_{\tau \sim P_{data}}[\log(p_\theta(\tau))]$ is the MLE between the generated and real data, which serves as the secondary objective for the generator to optimize the distribution match between the generated and behavior policies.

The objective shown by Eq. 6 is conducive to exact likelihood evaluation and supports adversarial and maximum likelihood training, but it brings an additional objective to the generator, increasing training instability. To handle this issue, we directly embed MLE into Value Iteration (VI) to drive the learned policy towards distribution matching, considering that the AC and the GAN share the same generator. In order to achieve this purpose, we first define a distribution matching operator as Definition 1, which is used to constrain policies that maximize the Q-function, and then derive its performance bounds as Proposition 1.

Definition 1 *Given a behavior policy π_β, a learned policy π, and a threshold δ, we define $A_s^\delta = \{ a \sim \pi_\theta(\cdot \mid s) \mid \log \pi_\theta(a \mid s) > \delta, \ (s,a) \in \mathcal{D}\}$. The distribution matching operator is defined as: $\mathcal{T}_\delta Q(s,a) = \mathbb{E}_{s'}\left[r(s,a) + \gamma \max_{a' \in A_{s'}^\delta} Q(s',a')\right]$, where there is an optimal fixed point $Q_\delta^*(s,a)$.*

We establish bounds on the suboptimality of a suboptimal Q_δ^* relative to the optimal Q-function Q^* .

Proposition 1 *The upper bound of the suboptimality of Q_δ^* is defined as $\|Q^* - Q_\delta^*\|_\infty \leq \frac{1}{1-\gamma}\alpha(\delta)$, where $\alpha(\delta) = \max_{s,a} |\mathcal{T}_\delta Q^*(s,a) - \mathcal{T}Q^*(s,a)|$.*

To sum up, to improve the ability to generate distribution-matching data for the generator, we incorporate the MLE term into the objective of the policy improvement. We rewrite the MLE term by introducing the state-action pairs and use π_θ to replace the generator p_θ in Eq. 6, as shown by Eq. 7.

$$\pi = \underset{\pi}{\operatorname{argmax}}[\mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\beta(\cdot|s)}[Q(s, \pi_\theta(s)) + \lambda \log(\pi_\theta(a|s))]], \tag{7}$$

where $\mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\beta(\cdot|s)}[\log(\pi_\theta(a|s))]$ denotes the MLE term, and λ is a hyperparameter.

Diffusion policy-based MLE: MLE tends to cover all the modes of distribution and exhibits poor robustness to model misspecification in some high-dimensional scenarios. In contrast, diffusion models not only better fit the offline data distribution to fool the discriminator, but also improve the generalization ability of the policy. Therefore, we approximate MLE through a diffusion process. Based on the aforementioned analysis, we can formally describe the approximation process of MLE in policy distribution matching. To this end, we present Proposition 2.

Proposition 2 *The MLE between the distributions of the learned policy and behavior policy can be approximated by MSE-like loss based on ELBO, as shown by Eq. 8.*

$$\mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\beta(\cdot|s)}[\log(\pi_\theta(a|s))] \approx -L_d(\pi_\beta, \pi_\theta). \tag{8}$$

where $L_d(\pi_\beta, \pi_\theta)$ is the diffusion loss.

The policy π_θ can be represented by a conditional diffusion, as shown by Eq. 9.

$$\pi_\theta(a \mid s) = p_\theta(a^{0:N} \mid s) \\ = \mathcal{N}(a^N; \mathbf{0}, \mathbf{I}) \prod_{i=1}^{N} p_\theta(a^{i-1} \mid a^i, s), \tag{9}$$

where a^0 represents the end sample of the reverse chain and is used in the evaluation process of RL. $p_\theta(a^{0:N} \mid s)$ denotes the joint distribution of all noisy samples.

In the sampling process, we first sample $a^N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and then sample from the reverse diffusion chain parameterized by θ, as shown by Eq. 10.

$$a^{i-1} \mid a^i = \frac{a^i}{\sqrt{\alpha_i}} - \frac{\beta_i}{\sqrt{\alpha_i(1-\bar{\alpha}_i)}}\epsilon_\theta(a^i, s, i) + \sqrt{\beta_i}\epsilon, \\ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \ \text{ for } i = N, \ldots, 1. \tag{10}$$

where we adopt the same noise schedule β_i as in (Wang, Hunt, and Zhou 2023).

The diffusion policy can be converted into parametrized policy with the optimization objective, as shown by Eq. 11.

$$\pi = \arg\max_\pi \mathbb{E}_{s\sim\mathcal{D}}[Q(s, \pi_\theta(s))] - \lambda L_d(\pi_\beta, \pi_\theta) \tag{11}$$

where λ is a hyperparameter. In this case, we convert the MLE into a solvable parameterized network that satisfies the distribution matching requirement, thus generating sufficient data to fool the discriminator, which is also an alternative interpretation for the behavior cloning term of diffusion policies in Diffusion Q-learning (Wang, Hunt, and Zhou 2023).

Discriminator-based Policy Confidence Mechanism

The distribution matching method proposed above can simultaneously fool the discriminator while maximizing expected rewards in offline RL. However, the issues of the stability of policy learning and the difficulty of learning high-reward behaviors in complex environments with sparse rewards are not taken into account. To this end, we first design a sequential stacking input technique that enables the discriminator to provide smooth and stable gradient feedback to the generator. A policy confidence is then proposed to balance the diffusion loss and Q-values and guide the exploration of optimal actions in complex environments. In addition, considering the adversarial learning relation of the generator and the discriminator, we use the discriminator to measure the distributional shift of the generated data and incorporate the policy confidence as a regularization term to constrain policy exploration.

Sequential stacking input: GANs often suffer from instability and vanishing gradients. We found that incorporating continuous trajectory information, named Sequential Stacking (Zhang et al. 2020; Zhu et al. 2022), as input allows the discriminator to comprehensively assess whether the generated samples originate from the real distribution in offline RL, leading to smooth gradient feedback for stable training.

Proposition 3 *Let $\mathcal{D} = \{(s_i, a_i)\}$ be an offline dataset generated by a single policy. Assume* local smoothness condition: $\exists\, \delta, \varepsilon > 0,\ \|s - s'\| \leq \delta \implies \|a - a'\| \leq \varepsilon$. *The true density is indicated by $p_{\mathrm{d}}(s, a)$. Let*

$$p_g(s, a) = p_{\mathrm{d}}(s)\, \delta\big(a - G(s)\big), \tag{12}$$

where the deterministic generator outputs $\hat{a} = G(s)$. $p_g(s, a)$ and $p_{\mathrm{d}}(s)$ denote generated state–action joint density and empirical state-only marginal density, respectively. $\hat{a}$ represents the generated action. Define

- *$D_1(s, a)$: marginal discriminator on a single pair (s, a);*
- *$D_2(s, a, s', a')$: stacked discriminator on a neighboring pair $(s, a), (s', a')$ with $\|s - s'\| \leq \delta$.*

If G is not ε-Lipschitz, under 1*-Lipschitz for both the two discriminators $D_1(s, a)$ and $D_2(s, a, s', a')$, the generator gradients $\nabla_\theta \mathcal{L}_g$ satisfy:*

$$\left\|\nabla_\theta \mathcal{L}_g^{(2)}\right\| \ \geq\ \left\|\nabla_\theta \mathcal{L}_g^{(1)}\right\| + \Delta, \qquad \Delta > 0. \tag{13}$$

Based on Eq. 13, feeding stacked neighboring state–action pairs into the discriminator can deliver a large gradient to the generator, mitigating gradient vanishing and improving training stability of GAN. In the follow text, we use $D_{neb}(s, a)$ and $D_{neb}(s, \pi_\theta(s))$ to replace $D_2(s, a, s', a')$ and $D_2(s, \hat{a}, s', \hat{a'})$, respectively.

Diffusion policy confidence: To balance the attention on exploration and decision-making, and ensure the stability of policy learning, we propose a method that dynamically balances the diffusion loss and Q-values by constructing the confidence of the diffusion policy using the ratio of the discriminator scores, as shown by Eq. 14.

$$\begin{aligned}\pi = \arg\max_\pi \mathbb{E}_{s,a\sim\mathcal{D}(a|s)}\left[Q(s, \pi_\theta(s))\right] \\ -\lambda \frac{D_{neb}(s, \pi_\theta(s))}{D_{neb}(s, a)} L_d(\pi_\beta, \pi_\theta)\end{aligned} \tag{14}$$

where $D_{neb}(s, \pi_\theta(s))$ and $D_{neb}(s, a)$ denote the discriminator's score for generated and real neighboring pair, respectively. This confidence indicates the relative accuracy of the current diffusion policy, which evolves as the policy learning process. In the early stage of policy learning, the distribution-matching constraint decreases to encourage the agent to explore more data. And in the later stage, it gradually increases to enhance the distribution-matching ratio as the policy network continues to learn, thereby achieving effective policy regularization. Moreover, the confidence term can be regarded as a form of importance weighting, thereby effectively mitigating policy bias and improving the accuracy of value estimation.

Discriminator regularization: Solely relying on the aforementioned behavior-cloning-like loss to adjust the Q-function would make the learned policy overly dependent on the behavior policy, similar to TD3+BC (Fujimoto and Gu 2021), which significantly impairs the policy performance in complex environments with sparse rewards. Considering that the discriminator scoring $D(s, a)$ for the actions can indicate the degree of match between the generated actions and the real actions, the discriminator scoring can be used to assess the quality of generated policies. Therefore, we use the discriminator scoring $D(s, a)$ as a regularization of the Q-function to limit the overestimation of poor actions, handling the issue of Q-value overestimation for OOD data in diffusion policies, as shown by Eq. 15.

$$\begin{aligned}\pi = \arg\max_\pi \mathbb{E}_{s,a\sim\mathcal{D}}[Q(s, \pi_\theta(s))] \\ + S_{D_{neb}}(s, \pi_\theta(s)) - \lambda \frac{D_{neb}(s, \pi_\theta(s))}{D_{neb}(s, a)} L_d(\pi_\beta, \pi_\theta),\end{aligned} \tag{15}$$

where $S_{D_{neb}}(s, \pi_\theta(s)) = -log(-D_{neb}(s, \pi_\theta(s)) + c)$ represents the score of the learned policy, $D_{neb}(s, \pi_\theta(s))$ is the discriminator scoring for the generated action $\pi_\theta(s)$, and the generator loss in the GAN is based on this score.

Practical Implementation

We incorporate the whole diffusion policy into the GAN framework to train the policies for offline RL. The updating process are shown by Eqs. 16 and 17.

$$\begin{aligned}Q^{k+1} \leftarrow \arg\min_Q\ \mathbb{E}_{(s,a,s')\sim\mathcal{D},\ a'\sim\pi^k_{\theta'}}\Big[\big\|\big(r(s, a) \\ +\gamma \min_{i=1,2} Q_{\phi'_i}(s', a') - Q_{\phi_i}(s, a)\big)\big\|^2\Big]\end{aligned} \tag{16}$$

$$\begin{aligned}\pi^{k+1} \leftarrow \arg\max_\pi \mathbb{E}_{s,a\sim\mathcal{D}}[\alpha \cdot Q^{k+1}(s, \pi^k_\theta(s))] \\ + S_{D_{neb}}(s, \pi^k_\theta(s)) - \lambda \frac{D^k_{neb}(s, \pi^k_\theta(s))}{D^k_{neb}(s, a)} L_d(\pi_\beta, \pi^k_\theta),\end{aligned} \tag{17}$$

where k denotes the k^{th} step of policy iteration, and $\alpha = \frac{1}{\mathbb{E}_{(s,a)\sim\mathcal{D}}[|Q_\phi(s,a)|]}$ is a normalization term based on the mini-batch $\{s, a\}$ according to (Fujimoto and Gu 2021).

After introducing the optimization objectives of the offline RL network, we define the update rules for the discriminator. We choose the same loss function as WGAN-GP (Gulrajani et al. 2017) to update the discriminator, as shown by Eq. 18.

$$\begin{aligned}\mathcal{L}_D = \mathbb{E}_{s\sim\mathcal{D}}\left[f_w(s, \pi_\theta(s))\right] - \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[f_w(s, a)\right] \\ + \mu \cdot \mathbb{E}_{(s,a)\sim\mathcal{D},\epsilon\sim\mathcal{U}(0,1)}\left[\left(\left\|\nabla_{\hat{a}} f_w(s, \hat{a})\right\|_2 - 1\right)^2\right]\end{aligned} \tag{18}$$

where $\mathcal{L}_D$ is the loss function of WGAN-GP. f_w denotes the discriminator, and $\hat{a} = \epsilon a + (1 - \epsilon)\pi_\theta(s)$.

Algorithm 1 provides the overall procedure of our algorithm. At each training step, we sample a batch of training data from the offline dataset and then update the parameters of the value function, policy, and discriminator, respectively.

Experiments

Benchmarks and Baselines

We evaluate our method on the D4RL (Fu et al. 2020) benchmark with two different domains: Gym-MuJoCo and Antmaze. Gym-MuJoCo is a classical domain for evaluating locomotion tasks, where the data collected by the behavior

Dataset	IQL	CQL	IDQL-A	SFBC	Diffuser	DQL	QIPO	D-DICE	DMGDP(Ours)
Halfcheetah-medium	47.4	44.0	51.0	45.9	44.2	51.1	48.2	60.0	**71.6 ± 0.7**
Hopper-medium	66.3	58.5	65.4	57.1	58.5	90.5	89.5	100.2	**101.3 ± 2.1**
Walker2d-medium	72.5	72.5	82.5	77.9	79.7	87.0	85.0	**89.3**	88.7 ± 1.8
Halfcheetah-medium-expert	86.7	90.7	95.9	92.6	79.2	96.8	94.1	97.3	**99.4 ± 2.1**
Hopper-medium-expert	101.5	105.4	108.6	108.6	107.2	111.1	112.1	**112.2**	111.1 ± 0.4
Walker2d-medium-expert	110.6	109.6	112.7	109.8	108.4	110.1	110.1	**114.1**	110.3 ± 1.7
Halfcheetah-medium-replay	44.2	45.5	45.9	37.1	42.4	47.8	45.3	49.2	**64.5 ± 0.8**
Hopper-medium-replay	95.2	95.0	92.1	86.2	96.8	101.3	101.2	**102.3**	102.1 ± 0.6
Walker2d-medium-replay	76.1	77.2	85.1	65.1	61.2	95.5	90.1	90.8	**99.5 ± 1.4**
Mujoco-mean	77.8	77.6	82.1	75.6	75.3	87.9	86.2	90.6	**94.28**
Antmaze-umaze	85.5	84.8	94.0	92.0	-	93.4	97.5	98.1	**99.4 ± 0.8**
Antmaze-umaze-diverse	66.7	43.4	80.2	85.3	-	66.2	73.9	82.0	**88.2 ± 2.9**
Antmaze-medium-play	72.2	65.2	84.5	81.3	-	76.6	82.8	**91.3**	85.3 ± 5.1
Antmaze-medium-diverse	71.0	54.0	84.8	82.0	-	78.6	86.0	85.7	**89.3 ± 3.2**
Antmaze-large-play	39.6	38.4	63.5	59.3	-	46.4	**73.3**	68.6	62.7 ± 4.2
Antmaze-large-diverse	47.5	31.6	67.9	45.5	-	57.3	40.5	**72.0**	69.8 ± 2.2
Antmaze-mean	63.7	52.9	79.1	74.2	-	69.8	77.3	**82.95**	82.45
All-tasks mean	72.2	67.7	80.9	75.0	-	80.6	82.64	87.54	**89.54**

Table 1: Experimental results of DMGDP and other SOTA methods on D4RL. The mean and standard deviation of the normalized scores are averaged over 5 random seeds. We report the performance of baselines using their best results from the original papers.

Algorithm 1: DMGDP algorithm

Initialize the policy network π_θ, critic network $Q_{\phi 1}$, $Q_{\phi 2}$ and target network $\pi_{\theta'}$, $Q'_{\phi 1}$, $Q'_{\phi 2}$, discriminator D_ω, offline replay buffer $\mathcal{D}$ and policy delay step d.
for step $i = 0$ to T **do**
  Sample minibatch of transitions (s, a, r, s') from $\mathcal{D}$
  Sample $a' \sim \pi_\theta(\cdot|s')$ according to Eq. 10
  // **Discriminate updating**
  $\omega^{k+1} \leftarrow$ Update D_ω according to Eq. 18
  // **Q-value function updating**
  $\phi^{k+1} \leftarrow$ Update Q-function Q_ϕ using the Bellman update according to Eq. 16
  if i mod d **then**
    // **Policy updating**
    Sample $a_t^0 \sim \pi_\theta(a_t|s_t)$ according to Eq. 10
    $\theta^{k+1} \leftarrow$ Update policy π_θ according to Eq. 17
  end if
  // **Target networks updating**
  $\theta' = \rho\theta' + (1-\rho)\theta$, $\phi'_i = \rho\phi'_i + (1-\rho)\phi_i$ for $i = 1, 2$
end for

policy only cover a small part of the state-action space, so it is appropriate to evaluate the methods of handling the OOD issue (Fujimoto, Meger, and Precup 2019; Kumar et al. 2020). In the Gym-MuJoCo domain, we focus on three locomotion tasks: Walker2d-v2, Hopper-v2, and HalfCheetah-v2. All the three tasks contain three different datasets including medium, medium-expert, and medium-replay. Antmaze is a domain of evaluating control and navigation tasks, where the data consists of undirected trajectories, and it is appropriate to evaluate the data reorganization capacity for RL methods. In the Antmaze domain, different tasks represent the mazes with different sizes and different complexity levels.

To show the performance of DMGDP, we compare it with five SOTA diffusion-based offline RL methods including DQL (Wang, Hunt, and Zhou 2023), IDQL (Hansen-Estruch et al. 2023), SFBC (Chen et al. 2023a), Diffuser (Janner et al. 2022), QIPO (Zhang et al. 2025), and D-DICE (Mao et al. 2024). Moreover, we also select IQL (Kostrikov, Nair, and Levine 2021) and CQL (Kumar et al. 2020) as the comparative methods because of their competitive performance in offline RL. Details are provided in Appendix A.4.

Comparative Results with SOTA Methods

Table 1 shows that DMGDP achieves competitive results compared with other offline RL methods. DMGDP achieves the best average results for all tasks and outperforms the baseline method Diffusion-QL by 7% and 18% in the Mujoco and Antmaze domains, respectively, owing to highly expressive policies and effective policy optimization. In the medium-replay dataset that contains a lot of bad data from the early training stages, where offline RL algorithms are arduous to learn optimal policies, DMGDP achieves comparative performance among these methods.

In the Antmaze domain with sparse rewards and a large number of undirected trajectories, DMGDP can also achieve competitive results, as shown in Table 1. In some small scenario tasks like umaze, DMGDP achieves an average success rate of 99.4, because the scoring for the sampled actions by the discriminator can help the Q-function make a reasonable estimation for the current action, which effectively prevents the agent from selecting bad data and ensures a stable learning process. In the medium-play and large-play tasks, our

Tasks	Gaussian policy	Diffusion policy
Halfcheetah-medium	70.3 ± 1.4	$\mathbf{71.6 \pm 0.7}$
Hopper-medium	60.2 ± 2.3	$\mathbf{101.3 \pm 2.1}$
Walker2d-medium	86.1 ± 1.7	$\mathbf{88.7 \pm 1.8}$
Halfcheetah-medium-expert	96.1 ± 2.9	$\mathbf{99.4 \pm 2.1}$
Hopper-medium-expert	104.7 ± 2.3	$\mathbf{111.1 \pm 0.4}$
Walker2d-medium-expert	108.7 ± 2.5	$\mathbf{110.3 \pm 1.7}$
Halfcheetah-medium-replay	57.5 ± 1.0	$\mathbf{64.5 \pm 0.8}$
Hopper-medium-replay	94.8 ± 1.4	$\mathbf{102.1 \pm 0.6}$
Walker2d-medium-replay	96.1 ± 2.8	$\mathbf{99.5 \pm 1.4}$
average score	86.06	**94.28**

Table 2: Comparative results of the Gaussian and diffusion policy. The results are obtained from 5 random seeds.

Tasks	Without	With
Halfcheetah-medium	68.3 ± 1.9	$\mathbf{71.6 \pm 0.7}$
Hopper-medium	97.0 ± 3.2	$\mathbf{101.3 \pm 2.1}$
Walker2d-medium	$\mathbf{89.2 \pm 1.9}$	88.7 ± 1.8
Halfcheetah-medium-expert	$\mathbf{100.8 \pm 2.6}$	99.4 ± 2.1
Hopper-medium-expert	102.7 ± 0.8	$\mathbf{111.1 \pm 0.4}$
Walker2d-medium-expert	109.8 ± 2.9	$\mathbf{110.3 \pm 1.7}$
Halfcheetah-medium-replay	57.6 ± 0.9	$\mathbf{64.5 \pm 0.8}$
Hopper-medium-replay	99.2 ± 1.1	$\mathbf{102.1 \pm 0.6}$
Walker2d-medium-replay	91.5 ± 2.6	$\mathbf{99.5 \pm 1.4}$
Antmaze-medium-play	79 ± 6.4	$\mathbf{85.3 \pm 5.1}$
Antmaze-medium-diverse	82 ± 4.6	$\mathbf{89.3 \pm 3.2}$
average score	88.83	**93.01**

Table 3: Experiment training without and with the policy confidence term. Results are averaged over 5 random seeds.

method performs a little worse than D-DICE because the double regularization in DMGDP reduces the sensitivity to a single behavior policy, which leads to performance degradation in some special tasks.

Furthermore, DMGDP performs well on other robotic domains, such as the multi-task domain Adroit and spare-reward domain Maze2d. More details are shown in Appendix A.5.

Ablation Studies

We conduct ablation studies on three components that affect the performance of the proposed DMGDP method, including the policy type, hyperparameter λ, and the regularization term based on the discriminator output.

Policy type: To demonstrate the feasibility of the proposed distribution matching method and the experimental viability of approximating MLE using diffusion loss, we compare the different policy types in our distribution-matching RL framework, including Gaussian policy and diffusion policy. The Gaussian policy is learned through a variational auto-encoder (VAE), and the mean and variance provided by the VAE are used to approximate the MLE. The result in Table 2 shows that the Gaussian policy can play a certain role,

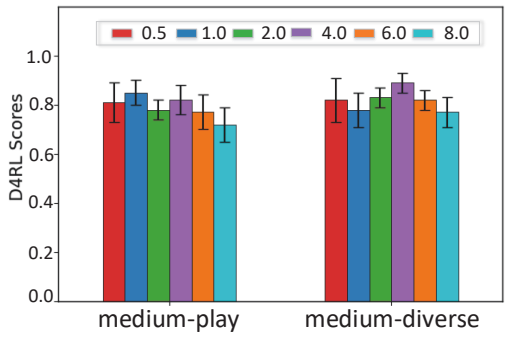

Figure 2: Ablation studies with different λ on Antmaze-medium environments over 5 random seeds.

but becomes overly constrained by the behavior policy due to its limited policy expressiveness, leading to suboptimal performance with an average score of 86.06. In contrast, the proposed diffusion policy achieves an average score of 94.28, which is better than the Gaussian policy.

Hyperparameter λ: In our method, λ controls the strength of policy regularization, a high λ implies a strict constraint and vice versa. We evaluated our method with the values of 0.5, 1, 2, 4, 6, and 8, which cover the adjustment range of λ, on AntMaze-medium environments with sparse rewards. Figure 2 shows that the value of λ does not obviously affect the stability of the proposed method.

Policy confidence: The policy confidence based on the discriminator, which is used to score the effectiveness of the sampling results, is important to obtain good performance in our method. To show the effectiveness of the policy confidence based on the discriminator, we conduct an additional experiment that evaluates the performances of the proposed DMGDP with and without the policy confidence on the selected tasks. As shown in Table 3, the average score of the method with policy confidence exceeds the method without it by 4%, which means that the proposed policy confidence term provides a reasonable and conservative weakening scheme for the diffusion loss to avoid overestimation for the Q-value. Besides, the improvement is more applicable for Antmaze tasks because the policy confidence term enables a more rational evaluation for actions, especially in Antmaze tasks with sparse rewards.

Conclusions

In this paper, we propose the DMGDP, an offline RL method based on diffusion policies with distribution-matching generator, to address the challenge of fooling the discriminator while maximizing the expected returns. In DMGDP, the generator serves as the actor and the discriminator provides the probability that generated actions are in behavior policies. Besides, we develop a discriminator-based policy optimization method to appropriately constrain policy exploration. Experimental results on the D4RL benchmark demonstrate that DMGDP outperforms SOTA methods. Future work will focus on how to improve the sampling efficiency and reduce the time consuming of diffusion policies.

Acknowledgements

We gratefully acknowledge the support from the National Natural Science Foundation of China (62273135), the Natural Science Foundation of Hubei Province (2025AFA083), and the Original Exploration Seed Project of Hubei University (202416403000001).

References

Ada, S. E.; Oztop, E.; Ugur, E.; and Wu. 2024. Diffusion policies for out-of-distribution generalization in offline reinforcement learning. *IEEE Robotics and Automation Letters*, 9(4): 3116–3123.

Ajay, A.; Du, Y.; Gupta, A.; Tenenbaum, J.; Jaakkola, T.; and Agrawal, P. 2022. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*.

An, G.; Moon, S.; Kim, J.-H.; and Song, H. O. 2021. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34: 7436–7447.

Ashvin, N.; Murtaza, D.; Abhishek, G.; and Sergey, L. 2020. Accelerating online reinforcement learning with offline datasets. *CoRR, vol. abs/2006.09359*.

Chen, H.; Lu, C.; Ying, C.; Su, H.; and Zhu, J. 2023a. Offline reinforcement learning via high-fidelity generative behavior modeling. In *The Eleventh International Conference on Learning Representations*.

Chen, Z.; Kiami, S.; Gupta, A.; and Kumar, V. 2023b. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 02783649241273668.

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. In *Advances in neural information processing systems*, volume 34, 20132–20145.

Fujimoto, S.; Meger, D.; and Precup, D. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, 2052–2062.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, volume 27.

Grover, A.; Dhar, M.; and Ermon, S. 2018. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Hansen-Estruch, P.; Kostrikov, I.; Janner, M.; Kuba, J. G.; and Levine, S. 2023. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*, volume 33, 6840–6851.

Hu, X.; Li, S.; Huang, T.; Tang, B.; Huai, R.; and Chen, L. 2024. How Simulation Helps Autonomous Driving: A Survey of Sim2real, Digital Twins, and Parallel Intelligence. *IEEE Transactions on Intelligent Vehicles*.

Hu, X.; Liu, Y.; Tang, B.; Yan, J.; and Chen, L. 2023. Learning dynamic graph for overtaking strategy in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*.

Hu, X.; Tang, B.; Chen, L.; Song, S.; and Tong, X. 2021. Learning a deep cascaded neural network for multiple motion commands prediction in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*.

Janner, M.; Du, Y.; Tenenbaum, J. B.; and Levine, S. 2022. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 9902–9915.

Kober, J.; Bagnell, J. A.; and Peters, J. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*.

Kostrikov, I.; Nair, A.; and Levine, S. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*.

Kumar, A.; Fu, J.; Soh, M.; Tucker, G.; and Levine, S. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 1179–1191.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Li, J.; Zhan, X.; Xu, H.; Zhu, X.; Liu, J.; and Zhang, Y.-Q. 2022. When data geometry meets deep function: Generalizing offline reinforcement learning. *arXiv preprint arXiv:2205.11027*.

Liu, S.; See, K. C.; Ngiam, K. Y.; Celi, L. A.; Sun, X.; and Feng, M. 2020. Reinforcement learning for clinical decision support in critical care: comprehensive review. *Journal of medical Internet research*.

Liu, Y.; and Hofert, M. 2024. Implicit and Explicit Policy Constraints for Offline Reinforcement Learning. In *Causal Learning and Reasoning*, 499–513. PMLR.

Lu, C.; Ball, P.; Teh, Y. W.; and Parker-Holder, J. 2024. Synthetic experience replay. In *Advances in Neural Information Processing Systems*, volume 36.

Lu, C.; Chen, H.; Chen, J.; Su, H.; Li, C.; and Zhu, J. 2023. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning*, 22825–22855. PMLR.

Lyu, J.; Ma, X.; Li, X.; and Lu, Z. 2022. Mildly conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 1711–1724.

Mao, L.; Xu, H.; Zhan, X.; Zhang, W.; and Zhang, A. 2024. Diffusion-dice: In-sample diffusion guidance for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37: 98806–98834.

Menéndez, M.; Pardo, J.; Pardo, L.; and Pardo, M. 1997. The jensen-shannon divergence. *Journal of the Franklin Institute*.

Ni, F.; Hao, J.; Mu, Y.; Yuan, Y.; Zheng, Y.; Wang, B.; and Liang, Z. 2023. Metadiffuser: Diffusion model as conditional planner for offline meta-rl. In *International Conference on Machine Learning*, 26087–26105. PMLR.

Venkatraman, S.; Jain, M.; Scimeca, L.; Kim, M.; Sendera, M.; Hasan, M.; Rowe, L.; Mittal, S.; Lemos, P.; Bengio, E.; et al. 2024. Amortizing intractable inference in diffusion models for vision, language, and control. *Advances in neural information processing systems*, 37: 76080–76114.

Vuong, Q.; Kumar, A.; Levine, S.; and Chebotar, Y. 2022. Dasco: Dual-generator adversarial support constrained offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 38937–38949.

Wang, Z.; Hunt, J. J.; and Zhou, M. 2023. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*.

Wu, L.; Liu, Q.; Zhang, L.; and Huang, Z. 2024. Offline Reinforcement Learning with Generative Adversarial Networks and Uncertainty Estimation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5255–5259. IEEE.

Xu, H.; Jiang, L.; Jianxiong, L.; and Zhan, X. 2022. A policy-guided imitation approach for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 4085–4098.

Xu, H.; Jiang, L.; Li, J.; Yang, Z.; Wang, Z.; Chan, V. W. K.; and Zhan, X. 2023. Offline rl with no ood actions: In-sample learning via implicit value regularization. *arXiv preprint arXiv:2303.15810*.

Xu, H.; Zhan, X.; Li, J.; and Yin, H. 2021. Offline reinforcement learning with soft behavior regularization. *arXiv preprint arXiv:2110.07395*.

Zhang, J.; Zhang, C.; Wang, W.; and Jing, B. 2024. Constrained Policy Optimization with Explicit Behavior Density For Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 36.

Zhang, J.; Zhang, W.; Song, R.; Ma, L.; and Li, Y. 2020. Grasp for stacking via deep reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2543–2549. IEEE.

Zhang, S.; Zhang, W.; Gu, Q.; and Li. 2025. Energy-weighted flow matching for offline reinforcement learning. *arXiv preprint arXiv:2503.04975*.

Zhu, J.; Xia, Y.; Wu, L.; Deng, J.; Zhou, W.; Qin, T.; Liu, T.-Y.; and Li, H. 2022. Masked contrastive representation learning for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3421–3433.