Entropy-regularized Diffusion Policy with Q-Ensembles for Offline Reinforcement Learning

Ruoqi Zhang[*] **Ziwei Luo**[*] **Jens Sjölund** **Thomas B. Schön** **Per Mattsson**
Department of Information Technology, Uppsala University
`{ruoqi.zhang,ziwei.luo,jens.sjolund,thomas.schon,per.mattsson}@it.uu.se`

Abstract

Diffusion policy has shown a strong ability to express complex action distributions in offline reinforcement learning (RL). However, it suffers from overestimating Q-value functions on out-of-distribution (OOD) data points due to the offline dataset limitation. To address it, this paper proposes a novel entropy-regularized diffusion policy and takes into account the confidence of the Q-value prediction with Q-ensembles. At the core of our diffusion policy is a mean-reverting stochastic differential equation (SDE) that transfers the action distribution into a standard Gaussian form and then samples actions conditioned on the environment state with a corresponding reverse-time process. We show that the entropy of such a policy is tractable and that can be used to increase the exploration of OOD samples in offline RL training. Moreover, we propose using the lower confidence bound of Q-ensembles for pessimistic Q-value function estimation. The proposed approach demonstrates state-of-the-art performance across a range of tasks in the D4RL benchmarks, significantly improving upon existing diffusion-based policies. The code is available at https://github.com/ruoqizzz/entropy-offlineRL.

1 Introduction

Offline reinforcement learning (RL), also known as batch RL [29] focuses on learning optimal policies from a previously collected dataset without further active interactions with the environment [31]. Although offline RL offers a promising avenue for deploying RL in real-world settings where online exploration is infeasible, a key challenge lies in deriving effective policies from fixed datasets, which usually are diversified and sub-optimal. The direct application of standard policy improvement approaches is hindered by the distribution shift problem [10]. Previous works mainly address this issue by either regularizing the learned policy close to the behavior policy [10, 9] or by making conservative updates for Q-networks [28, 26].

Diffusion models have rapidly become a prominent class of highly expressive policies in offline RL [44, 49]. While this expressiveness is beneficial when modeling complex behaviors, it also means that the model has a higher capacity to overfit the noise or specific idiosyncrasies in the training data. To address this, existing work introduce Q-learning guidance and regard the diffusion loss as a special regularizer adding to the policy improvement process [44, 17, 22]. Such a framework has achieved impressive results on offline RL tasks. However, its performance is limited by pre-collected datasets (or behavior policies) and the learning suffers severe overestimation of Q-value functions on unseen state-action samples [31].

One promising approach is to increase exploration for out-of-distribution (OOD) actions, with the hope that the RL agent can be more robust to diverse Q-values and estimation errors [50]. Previous online RL algorithms achieve this by maximizing the entropy of pre-defined tractable policies such
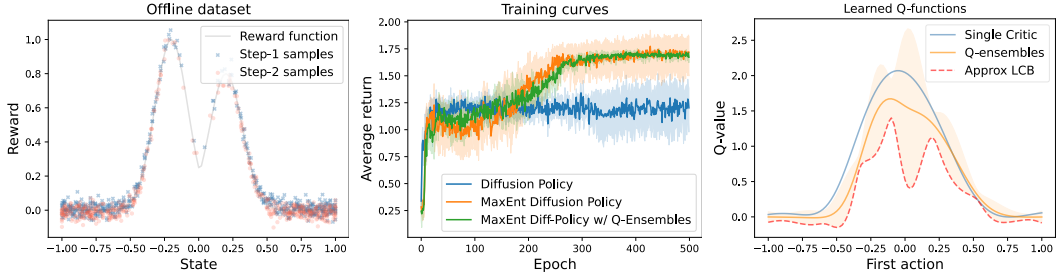
---

[*]Corresponding authors

Figure 1: A toy RL task in which the agent sequentially takes two steps (starting from 0) to seek a state with the highest reward. **Left**: The reward function is a mixture of Gaussian, and the offline data distribution is unbalanced with most samples located in low-reward states. **Center**: Training different policies on this task with 5 random seeds for 500 epochs. We find that a diffusion policy with entropy regularization and Q-ensembles yields the best results with low training variance. **Right**: Learned Q-value curve for the first step actions in state 0. The approximation of the lower confidence bound (LCB) of Q-ensembles is also plotted.

as Gaussians [35, 13, 15]. Unfortunately, directly computing the log probability of a diffusion policy is almost impossible since its generative process is a stochastic denoising sequence. Moreover, it is worth noting that entropy is seldom used in offline settings because it may lead to a distributional shift issue which may cause overestimation of Q-values on unseen actions in the offline dataset.

Another line of work addresses the overestimation problem by enforcing the Q-values to be more pessimistic [28, 20]. Inspired by this, uncertainty-driven RL algorithms employ an ensemble of Q-networks to provide different Q-value predictions for the same state-action pairs [1, 3]. The variation in these predictions serves as a measure of uncertainty. For state-action pairs exhibiting high predictive variance (e.g., OOD data points), these methods preferentially adopt pessimistic Q-value estimations as policy guidance.

In this paper, we present an entropy-regularized diffusion policy with Q-ensembles for offline RL. At the core of our method is a mean-reverting stochastic differential equation (SDE) [32] which allows us to sample actions from standard Gaussian conditioned on the environment state. We show that such an SDE provides a tractable entropy regularization that can be added in training to increase the exploration of OOD data points. In addition, we approximate the lower confidence bound (LCB) of Q-ensembles to alleviate potential distributional shifts, thereby learning a pessimistic policy to handle high uncertainty scenarios from offline datasets. As illustrated in Figure 1, both entropy regularization and Q-ensembles can improve RL performance on unbalanced offline datasets. The LCB approach further reduces the variance between different trials and provides a better estimation of unseen state-action pairs.

Our model achieves highly competitive performance across a range of offline D4RL benchmark tasks [8] and, in particular, significantly outperforms other diffusion-based approaches in the Antmaze environment. The superior performance demonstrates the effectiveness of the entropy-regularization and Q-ensembles. Overall, the proposed method encourages policy diversity and cautious decision-making, enhancing exploration while grounding the policy in the confidence of its value estimates derived from the offline dataset.

2   Background

This section reviews the core concepts of offline RL and then introduces the mean-reverting SDEs and shows how we sample actions from its reverse-time process. Note that there are two types of timesteps for RL and SDE. To clarify that, we use $i \in \{0, \ldots, N\}$ to denote the RL trajectories' step and $t \in \{0, \ldots, T\}$ to index diffusion discrete times.

Offline RL. We consider learning a Markov decision process (MDP) defined as $M = \{\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0\}$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively. The state transition probability is denoted $P(\mathbf{s}_{i+1} \mid \mathbf{s}_i, \mathbf{a}_i)$ and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ represents a reward function,

$\gamma \in (0, 1]$ is the discount factor, and d_0 is the initial state distribution. The goal of RL is to maximize the cumulative discounted reward $\sum_{i=0}^{\infty} \gamma^i \mathbb{E}_{\mathbf{a}_i \sim \pi(\mathbf{s}_i)} [r(\mathbf{s}_i, \mathbf{a}_i)]$ with a learned policy π. In contrast to online RL which requires continuous interactions with the environment, offline RL directly learns the policy from the static dataset $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}_{i+1})\}_{i=1}^{N_\mathcal{D}}$. In the offline setting, two primary challenges are frequently encountered: over-conservatism and a limited capacity to effectively utilize diversified datasets [31]. To address the issue of limited capacity, diffusion models have recently been employed to learn complex behavior policies from datasets [44].

Mean-Reverting SDE. Assume that we have a random variable $\boldsymbol{a}^0$ sampled from an unknown distribution $p_0(\boldsymbol{a})$. The mean-reverting SDE [32] is a diffusion process $\{\boldsymbol{a}^t\}_{t\in[0,T]}$ that gradually injects noise to $\boldsymbol{a}^0$:

$$\mathrm{d}\boldsymbol{a} = -\theta_t \boldsymbol{a}\, \mathrm{d}t + \sigma_t\, \mathrm{d}\boldsymbol{w}, \quad \boldsymbol{a}^0 \sim p_0(\boldsymbol{a}), \tag{1}$$

where $\boldsymbol{w}$ is the standard Wiener process, θ_t and σ_t are predefined positive parameters that characterize the speed of mean reversion and the stochastic volatility, respectively. Compared to IR-SDE [32], we set the mean to 0 to let the process drift to pure noise to fit the RL environment. The mean can however be tuned to high-reward actions in the offline dataset or prior knowledge. By setting $\sigma_t^2 = 2\theta_t$ for all diffusion steps, the solution to the forward SDE ($\tau < t$) is given by

$$p(\boldsymbol{a}^t \mid \boldsymbol{a}^\tau) = \mathcal{N}(\boldsymbol{a}^t \mid \boldsymbol{a}^\tau e^{-\bar{\theta}_{\tau:t}}, (1 - e^{-2\bar{\theta}_{\tau:t}})\boldsymbol{I}), \tag{2}$$

where $\bar{\theta}_{\tau:t} := \int_\tau^t \theta_z\, \mathrm{d}z$ are known coefficients [32]. In the limit $t \to \infty$, the marginal distribution $p_t(\boldsymbol{a}) = p(\boldsymbol{a}^t \mid \boldsymbol{a}^0)$ converges to a standard Gaussian $\mathcal{N}(0, \boldsymbol{I})$. This gives the forward process its informative name, i.e. "*mean-reverting*". Then, Anderson [2] states that we can generate new samples from Gaussian noises by reversing the SDE (1) as

$$\mathrm{d}\boldsymbol{a} = \left[-\theta_t\,\boldsymbol{a} - \sigma_t^2\,\nabla_{\boldsymbol{a}} \log p_t(\boldsymbol{a}) \right] \mathrm{d}t + \sigma_t\, \mathrm{d}\bar{\boldsymbol{w}}, \tag{3}$$

where $\boldsymbol{a}^T \sim \mathcal{N}(0, \boldsymbol{I})$ and $\bar{\boldsymbol{w}}$ is the reverse-time Wiener process. This reverse-time SDE provides a strong ability to fit complex distributions, such as the policy distribution represented in the dataset $\mathcal{D}$. Moreover, the ground truth score $\nabla_{\boldsymbol{a}} \log p_t(\boldsymbol{a})$ is acquirable in training. We can thus combine it with the reparameterization trick

$$\boldsymbol{a}^t = \boldsymbol{a}^0 e^{-\bar{\theta}_t} + \sqrt{1 - e^{-2\bar{\theta}_t}} \cdot \boldsymbol{\epsilon}_t \tag{4}$$

to train a time-dependent neural network $\boldsymbol{\epsilon}_\phi$ using noise matching on randomly sampled timesteps:

$$L_{\mathrm{diff}}(\phi) := \mathbb{E}_{t\in[0,T]}\left[\left\| \boldsymbol{\epsilon}_\phi(\boldsymbol{a}^t, t) - \boldsymbol{\epsilon}_t \right\| \right], \tag{5}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \boldsymbol{I})$ is a Gaussian noise and $\{\boldsymbol{a}^t\}_{t=0}^T$ denotes the discretization of the diffusion process. See Appendix A.1 for more details about the solution, reverse process, and loss function.

Sample Actions with SDE. Most existing RL algorithms employ unimodal Gaussian policies with learned mean and variance. However, this approach encounters a challenge when applied to offline datasets, which are typically collected by a mixture of policies and therefore hard to represent by a simple Gaussian model. Thus we prefer to represent the policy with an expressive model such as the reverse-time SDE. More specifically, the forward SDE provides theoretical guidance to train the neural network, and the reverse-time SDE (3) generates actions from Gaussian noise conditioned on the current environment state as a typical score-based generative process [43].

3 Method

We present the three core components of our method: 1) an efficient sampling strategy based on the mean-reverting SDE; 2) an entropy regularization term that enhances action space exploration; and 3) a pessimistic evaluation with Q-ensembles that avoids overestimation of unseen actions.

3.1 Optimal Sampling with Mean-Reverting SDE

We have shown how to sample actions with reverse-time SDEs in Section 2. However, generating data from the standard mean-reverting SDE [32] requires many diffusion steps and is sensitive to the noise scheduler [36]. To improve sample efficiency, we propose generating actions from the posterior distribution $p(\boldsymbol{a}^{t-1} \mid \boldsymbol{a}^t)$ conditioned on $\boldsymbol{a}^0$. This approach ensures fast convergence of the generative process while preserving its stochasticity.

Proposition 3.1. *Given an initial variable $\boldsymbol{a}^0$, for any diffusion state $\boldsymbol{a}^t$ at time $t \in [1, T]$, the posterior of the mean-reverting SDE (1) conditioned on $\boldsymbol{a}^0$ is*

$$p(\boldsymbol{a}^{t-1} \mid \boldsymbol{a}^t, \boldsymbol{a}^0) = \mathcal{N}(\boldsymbol{a}^{t-1} \mid \tilde{\mu}_t(\boldsymbol{a}^t, \boldsymbol{a}^0), \; \tilde{\beta}_t \mathbf{I}), \tag{6}$$

which is a Gaussian with mean and variance given by:

$$\tilde{\mu}_t(\boldsymbol{a}^t, \boldsymbol{a}^0) := \frac{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}}{1 - \mathrm{e}^{-2\bar{\theta}_t}} \mathrm{e}^{-\theta'_t} \boldsymbol{a}^t + \frac{1 - \mathrm{e}^{-2\theta'_t}}{1 - \mathrm{e}^{-2\bar{\theta}_t}} \mathrm{e}^{-\bar{\theta}_{t-1}} \boldsymbol{a}^0 \quad and \quad \tilde{\beta}_t := \frac{(1 - \mathrm{e}^{-2\bar{\theta}_{t-1}})(1 - \mathrm{e}^{-2\theta'_t})}{1 - \mathrm{e}^{-2\bar{\theta}_t}}, \tag{7}$$

where $\theta'_i := \int_{i-1}^{i} \theta_t dt$ and $\bar{\theta}_t$ is to substitute $\bar{\theta}_{0:t}$ for clear notation.

The proof is provided in Appendix A.2. Moreover, thanks to the reparameterization trick [25], we can approximate the variable $\boldsymbol{a}_0$ by reformulating Eq. (4) to

$$\hat{\boldsymbol{a}}^0 = \mathrm{e}^{\bar{\theta}_t} \left(\boldsymbol{a}^t - \sqrt{1 - \mathrm{e}^{-2\bar{\theta}_t}} \boldsymbol{\epsilon}_\phi(\boldsymbol{a}^t, t) \right), \tag{8}$$

where $\boldsymbol{\epsilon}_\phi$ is the learned noise prediction network. Then we combine Eq. (8) with Eq. (6) to iteratively construct the sampling process. In addition, it can be proved that the distribution mean $\tilde{\mu}_t(\boldsymbol{a}^t, \boldsymbol{a}^0)$ is the optimal reverse path from $\boldsymbol{a}^t$ to $\boldsymbol{a}^{t-1}$ (see Appendix A.3). Here, we illustrate a simple example of data reconstruction with different diffusion steps in Figure 2. Additional results are shown are provided in Appendix B.1. The proposed optimal sampling process from Proposition 3.1 requires only 5 steps, versus over 30 for the standard process. It clearly shows that the proposed optimal sampling is more efficient than the standard reverse-time SDE process.
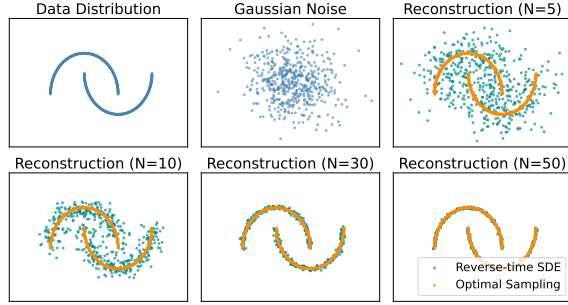

Figure 2: Comparison of the reverse-time SDE and optimal sampling process in data reconstruction.

Notation Note Recall that we have two distinct types of timesteps for RL and SDE denoted by i and t, respectively. To clarify the notation, in the following sections, we use $\boldsymbol{a}_i^t$ to represent the intermediate variable of an action taken at RL trajectory step i with SDE timestep t, as $\boldsymbol{a}_i^t = \boldsymbol{a}^t$ at state $\boldsymbol{s}_i$. Therefore, the action to take for state $\boldsymbol{s}_i$ is the final sampled action $\boldsymbol{a}_i$ denoted by $\boldsymbol{a}_i^0$. Hence, the policy is given by

$$\pi_\phi(\boldsymbol{a}_i^0 \mid \boldsymbol{s}_i) = p_\phi(\boldsymbol{a}^0) \tag{9}$$

While we cannot sample directly from this distribution we can efficiently sample the SDE's reverse joint distribution as

$$p_\phi(\boldsymbol{a}^{0:T}) = p(\boldsymbol{a}^T) \prod_{i=1}^{T} p_\phi(\boldsymbol{a}^{t-1} \mid \boldsymbol{a}^t), \tag{10}$$

where $p(\boldsymbol{a}^T) = \mathcal{N}(0, \boldsymbol{I})$ is Gaussian noise and the generative process is conditioned on the environment state $\boldsymbol{s}_i$. So to take an action from $\pi_\phi(\boldsymbol{a}_i^0 \mid \boldsymbol{s}_i)$, we sample from the joint distribution using Eq. (6) and Eq. (8) and finally pick out $\boldsymbol{a}_0$ as our sampled action. The visualization of is out method is provided in Appendix A.5.

3.2 Diffusion Policy with Entropy Regularization

The simplest strategy of learning a diffusion policy is to inject Q-value function guidance to the noise matching loss (5), in the hope that the reverse-time SDE (3) would learn to sample actions with higher values. This can be easily achieved by minimizing the following objective:

$$J_\pi(\phi) = L_{\text{diff}}(\phi) - \mathbb{E}_{\boldsymbol{s}_i \sim \mathcal{D}, \boldsymbol{a}_i^0 \sim \pi_\phi} \left[Q_\psi(\boldsymbol{s}_i, \boldsymbol{a}_i^0) \right], \tag{11}$$

where Q_ψ is the state-action value function approximated by a neural network, see Section 3.3.

4

This combination regards diffusion loss as a behavior-cloning term that learns the overall action distribution from offline datasets. However, the training is limited to existing data samples and the Q-learning term is sensitive to unseen actions. To address it, we propose to add an additional entropy term $\mathcal{H} = \mathbb{E}_{\boldsymbol{s}_i \sim \mathcal{D}} \left[- \log \pi_\phi(\cdot \mid \boldsymbol{s}_i) \right]$ to increase the exploration of the action space during training and rewrite the policy loss (11) to

$$J_\pi(\phi) = L_{\text{diff}}(\phi) - \lambda \, \mathbb{E}_{\boldsymbol{s}_i \sim \mathcal{D}, \boldsymbol{a}_i^0 \sim \pi_\phi} \left[Q_\psi(\boldsymbol{s}_i, \boldsymbol{a}_i^0) - \alpha \log \pi_\phi(\boldsymbol{a}_i^0 \mid \boldsymbol{s}_i) \right]. \tag{12}$$

where α is a hyperparameter that determines the relative importance of the entropy term versus Q-values, and $\lambda = \eta \, / \, \mathbb{E}_{(s,a) \sim \mathcal{D}}[|Q_\psi(s,a)|]$ to normalize the scale of the Q-values and balance loss terms. Iteratively generating the action $\boldsymbol{a}_i^0$ though a reverse diffusion process is computationally costly but, with an estimated noise $\boldsymbol{\epsilon}_\phi$ from diffusion term (5), we can thus directly use it to approximate $\boldsymbol{a}_i^0$ based on Eq. (8) for more efficient training.

Entropy Approximation. It is worth noting that the log probability of the policy $\log(\pi_\phi(\boldsymbol{a}_i^0 \mid \boldsymbol{s}_i))$ is in general intractable in the diffusion process. However, we found that the log probability of the joint distribution in Eq. (10) is tractable when conditioned on the sampled action $\boldsymbol{a}_i^0$. Proposition 3.1 further shows that the conditional posterior from $\boldsymbol{a}_i^1$ to $\boldsymbol{a}_i^0$ is Gaussian, meaning that

$$- \log \pi_\phi(\boldsymbol{a}_i^0 \mid \boldsymbol{s}_i) = - \log \pi_\phi(\boldsymbol{a}_i^1 \mid \boldsymbol{s}_i) + \mathcal{C}, \tag{13}$$

where $\mathcal{C}$ is a constant and $\boldsymbol{a}_i^1$ can be approximated using Eq. (2) similar to $\boldsymbol{a}_i^0$. The proof is provided in Appendix A.4. Then we can focus on the conditional reverse marginal distribution $p_\phi(\boldsymbol{a}_i^1 \mid \boldsymbol{a}_i^T, \boldsymbol{s}_i)$ that determines the exploration of actions and is acquirable via Bayes' rule:

$$p_\phi(\boldsymbol{a}_i^1 \mid \boldsymbol{a}_i^T, s_i) = \frac{p_\phi(\boldsymbol{a}_i^T \mid \boldsymbol{a}_i^1, \boldsymbol{s}_i) \, p_\phi(\boldsymbol{a}_i^1 \mid \boldsymbol{a}_i^0, \boldsymbol{s}_i)}{p_\phi(\boldsymbol{a}_i^T \mid \boldsymbol{a}_i^0, \boldsymbol{s}_i)}. \tag{14}$$

Since all terms in Eq. (14) can be computed with Eq. (2), we can rewrite the policy objective as

$$J_\pi(\phi) = L_{\text{diff}}(\phi) - \lambda \, \mathbb{E}_{\boldsymbol{s}_i \sim \mathcal{D}, (\hat{\boldsymbol{a}}_i^0, \hat{\boldsymbol{a}}_i^1) \sim \pi_\phi} \left[Q_\psi(\boldsymbol{s}_i, \hat{\boldsymbol{a}}_i^0) - \alpha \log(p(\hat{\boldsymbol{a}}_i^1 \mid \boldsymbol{a}_i^T, \boldsymbol{s}_i))) \right], \tag{15}$$

where $\hat{a}_i^0$ and $\hat{a}_i^1$ are approximate values calculated based on samples from the diffusion term. Note that the temperature α usually plays an important role in the maximum entropy RL framework and we thus provide a detailed analysis in Section 4.3.

3.3 Pessimistic Evaluation via Q-ensembles

Entropy regularization encourages diffusion policies to explore the action space, reducing the risk of overfitting pre-collected data. However, in offline RL, since the agent cannot collect new data during training, this exploration can lead to inaccuracies in value estimation for unseen state-action pairs [3, 11]. Instead of staying close to the behavior policy and being overly conservative, considering the uncertainty in the value function is an alternative approach.

In this work, we consider a pessimistic variant of a value-based method to manage the uncertainty and risks, i.e., the lower confidence bounds (LCB) with Q-ensembles. More specifically, we use an ensemble of Q-functions with independent targets to obtain an accurate LCB of Q-values. Each Q-function is updated based on its own Bellman target without sharing targets among ensemble members [11], as follows:

$$J_Q(\psi^i) = \mathbb{E}_{\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_{i+1} \sim \mathcal{D}} \left[Q_{\psi^m}(\boldsymbol{s}_i, \boldsymbol{a}_i) - y^m(r_i, \boldsymbol{s}_{i+1}, \pi_\phi) \right]$$
$$y^m = r_i + \gamma \mathbb{E}_{\boldsymbol{a}_{i+1} \sim \pi_\phi} [Q_{\bar{\psi}^m}(\boldsymbol{s}_{i+1}, \boldsymbol{a}_{i+1})] \tag{16}$$

where $\psi^m, \bar{\psi}^m$ are the parameters of the Q network and Q-target network for the mth Q-function. Then, the pessimistic LCB values are derived by subtracting the standard deviation from the mean of the Q-value ensemble,

$$Q_\psi^{\text{LCB}} = \mathbb{E}_{\text{ens}} \left[Q_{\psi^m}(\boldsymbol{s}, \boldsymbol{a}) \right] - \beta \left[\sqrt{\mathbb{V}_{\text{ens}} [Q_{\psi^m}(\boldsymbol{s}, \boldsymbol{a})]} \right] \tag{17}$$

where $\beta \geq 0$ is a hyperparameter determining the amount of pessimism, $\mathbb{V}[Q_{\psi^m}]$ is the variance of the ensembles, and $m \in \{1, \ldots, M\}$ where M the number of ensembles. Then, Q_ψ^{LCB} is used in the policy improvement step to balance entropy regularization and ensure robust performance. Finally, we use Q_ψ^{LCB} as the Q_ψ to (15). We summarize our method in Algorithm 1.

4 Experiment

In this section, we evaluate our methods on standard D4RL offline benchmark tasks [8] and provide a detailed analysis of entropy regularization, Q-ensembles, and training stability.

4.1 Setup

Datasets We evaluate our approach on four D4RL benchmark domains: Gym, AntMaze, Adroit, and Kitchen. In Gym, we examine three robots (halfcheetah, hopper, walker2d) across sub-optimal (medium), near-optimal (medium-expert), and diverse (medium-replay) datasets. The AntMaze domain challenges a quadrupedal ant robot to navigate mazes of varying complexities. The Adroit domain focuses on high-dimensional robotic hand manipulation, using datasets from human demonstrations and robot-imitated human actions. Lastly, the Kitchen domain explores different tasks within a simulated kitchen. These domains collectively provide a comprehensive framework for assessing RL algorithms across diverse scenarios.

Algorithm 1 Diffusion Policy with Q-Ensembles

Initialize parameters for $\pi_\phi, \pi_{\bar\phi}, \{Q_{\psi^m}, Q_{\bar\psi^m}\}_{m=1}^M$.
for each iteration **do**
  Sample mini-batch $\{(\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_{i+1})\}$ from $\mathcal{D}$.
  # Ensemble-Q learning
  Generate $\boldsymbol{a}_{i+1}^0 \sim \pi_{\bar\phi}(\boldsymbol{a}_{i+1} \mid \boldsymbol{s}_i)$ with (6) and (8).
  Update Q-networks $\{Q_{\psi^m}\}_{m=1}^M$ by (16).
  # Diffusion policy learning
  Sample $\{\boldsymbol{a}_i^t\}_{t\in[0,T]}$ from $\boldsymbol{a}_i$ with (4).
  Predict noise and approximate $\boldsymbol{a}_i^0, \boldsymbol{a}_i^1$ with (8).
  Update policy π_ϕ by (15) using Q_ψ^{LCB} from (17).
  # Update target networks
  $\bar\phi \leftarrow \eta\phi + (1-\eta)\bar\phi,$
  $\bar\psi^m \leftarrow \eta\psi^m + (1-\eta)\bar\psi^m, m \in \{1, \ldots, M\}$
end for

Implementation Details Following Diffusion-QL [44], we keep the network structure the same for all tasks with three MLP layers (hidden size 256, Mish activation [34]), and train models for 2000 epochs for Gym and 1000 epochs for others. Each epoch consists of 1000 training steps with a batch size of 256. We use Adam [24] to optimize both SDE and the Q-ensembles. Each model is evaluated by 10 trajectories for Gym tasks and 100 trajectories for others. In addition, our model is trained on an A100 GPU with 40GB memory for about 8 hours per task, and results are averaged over five random seeds.

Hyperparameters We keep key hyperparameters consistent: Q-ensemble size 64, LCB coefficient $\beta = 4.0$. The entropy temperature $\alpha = 0.01$ for Gym and AntMaze tasks and automated for Adroit and Kitchen tasks. The SDE sampling step is set to $T = 5$ for Gym and Antmaze tasks, $T = 10$ for Adroid and Kitchen tasks. For 'medium' and 'large' datasets of AntMaze, we use max Q-backup following Wang et al. [44] and Kumar et al. [28]. We also introduce the maximum likelihood loss for SDE training as proposed by Luo et al. [32]. More details are in Appendix B.2.

4.2 Comparison with other Methods

We compare our method with extensive baselines for each domain to provide a thorough evaluation and to understand the contributions of different components in our approach. The most fundamental among these are the behavior cloning (BC) method, BCQ [10] and BEAR [27] which restrict the policy to dataset behavior, highlighting the need for policy regularization and exploration. We also assess against Diffusion-QL [44] which integrates a diffusion model for policy regularization guided by Q-values. This comparison isolates the benefits of our enhanced sampling process and Q-ensemble integration. Our comparison includes CQL [28] and IQL [26], known for conservative Q-value updates. Additionally, we consider EDP [21], a variant of IQL with an efficient diffusion policy, and IDQL [17], which combines IQL as a critic with behavior cloning diffusion policy reweighted by learned Q-values. These comparisons evaluate the effectiveness of integrating diffusion policies with conservative value estimation. Finally, we include MSG [11], which combines independent Q-ensembles with CQL, and DT [4], treating offline RL as a sequence-to-sequence translation problem. These baselines help assess the robustness and generalizability of our method across different approaches. The performance comparison between baselines and ours is reported in Table 1 (Gym, Adroit and Kitchen) and Table 2 (AntMaze). Detailed results are discussed below.

Table 1: Average normalized scores on D4RL benchmark tasks. Results of BC, CQL, IQL, and IQL+EDP are taken directly from Kang et al. [21], and all other results are taken from their original papers. Our results are reported by averaging 5 random seeds.

Gym Tasks	BC	DT	CQL	IQL	IDQL-A	IQL+EDP	Diff-QL	Ours
Halfcheetah-medium-v2	42.6	42.6	44.0	47.4	51.0	48.1	51.1	**54.9**
Hopper-medium-v2	52.9	67.6	58.5	66.3	65.4	63.1	90.5	**94.2**
Walker2d-medium-v2	75.3	74.0	72.5	78.3	82.5	85.4	87.0	**92.5**
Halfcheetah-medium-replay-v2	36.6	36.6	45.5	44.2	45.9	43.8	47.8	**57.0**
Hopper-medium-replay-v2	18.1	82.7	95.0	94.7	92.1	99.1	101.3	**102.7**
Walker2d-medium-replay-v2	26.0	66.6	77.2	73.9	85.1	84.0	**95.5**	94.2
Halfcheetah-medium-expert-v2	55.2	86.8	91.6	86.7	95.9	86.7	**96.8**	90.3
Hopper-medium-expert-v2	52.5	107.6	105.4	91.5	108.6	99.6	111.1	**111.9**
Walker2d-medium-expert-v2	107.5	108.1	108.8	109.6	**112.7**	109.0	110.1	111.2
Average	51.9	74.7	77.6	77.0	82.1	79.9	88.0	**89.9**

Adroit Tasks	BC	BCQ	BEAR	CQL	IQL	IQL+EDP	Diff-QL	Ours
Pen-human-v1	63.9	68.9	-1.0	37.5	71.5	72.7	**72.8**	70.0
Pen-cloned-v1	37.0	44.0	26.5	39.2	37.3	**70.0**	57.3	68.4
Average	50.5	56.5	12.8	38.4	54.4	**71.4**	65.1	69.2

Kitchen Tasks	BC	BCQ	BEAR	CQL	IQL	IQL+EDP	Diff-QL	Ours
kitchen-complete-v0	65.0	8.1	0.0	43.8	62.5	75.5	84	**92.7**
kitchen-partial-v0	38.0	18.9	13.1	49.8	46.3	46.3	60.5	**66.3**
kitchen-mixed-v0	51.5	8.1	47.2	51	51	56.5	62.6	**68.0**
Average	51.5	11.7	20.1	48.2	53.3	59.4	69.0	**75.7**

Table 2: Average normalized scores on D4RL **AntMaze** tasks. Results of BC, DT, CQL, IQL, and IQL+EDP are taken directly from Kang et al. [21], and all other results are taken from their original papers. Our results are reported by averaging 5 random seeds.

AntMaze Tasks	BC	DT	CQL	IQL	MSG	IDQL-A	IQL+EDP	Diff-QL	Ours
Antmaze-umaze-v0	54.6	59.2	74	87.5	97.8	94.0	87.5	93.4	**100**
Antmaze-umaze-diverse-v0	45.6	53.0	84.0	62.2	**81.8**	80.2	62.2	66.2	79.8
Antmaze-medium-play-v0	0.0	0.0	61.2	71.2	89.6	84.5	71.2	76.6	**91.4**
Antmaze-medium-diverse-v0	0.0	0.0	53.7	70.0	88.6	84.8	70.0	78.6	**91.6**
Antmaze-large-play-v0	0.0	0.0	15.8	39.6	72.6	63.5	39.6	46.4	**81.2**
Antmaze-large-diverse-v0	0.0	0.0	14.9	47.5	71.4	67.9	47.6	56.6	**76.4**
Average	16.7	18.7	50.6	63.0	83.6	79.1	63.0	69.6	**86.7**

Gym tasks Most approaches perform well on Gym 'medium-expert' and 'medium-replay' tasks with high-quality data but drop severely on 'medium' tasks with suboptimal trajectories. Diffusion-QL [44] achieves a better performance through a highly expressive diffusion policy. Our method further improves performance across all three 'medium' tasks. The results illustrate the efficacy of combining diffusion policy with entropy regularization and Q-ensembles in preventing overfitting to suboptimal behaviors. By maintaining policy stochasticity, our algorithm encourages the exploration of action spaces, potentially discovering better strategies than those in the dataset.

Adroit and Kitchen Most offline approaches cannot achieve expert performance on these tasks due to the narrowness of human demonstrations in Adroit and the indirect, multitask data in Kitchen [44]. Our method outperforms all other approaches in the Kitchen tasks which suggests its ability to "stitching" the dataset and generalization. In addition, we fix the entropy coefficient α to be the same as other tasks for a robust setting. Even so, our method still achieves a competitive performanc in Adroit tasks. This fixed α leads the agent to continuously explore the action space throughout the entire training process, even when encountering unseen states. While exploration is generally advantageous, it can be detrimental in environments with limited data variability. Additionally, unlike in antmaze tasks, random actions are more likely to negatively impact performance in tasks where precise control is essential like Adroit. Moreover, it's worth noting that slightly tuning α leads to a SOTA performance, as illustrated in Table 3.

AntMaze AntMaze tasks are more challenging, requiring point-to-point navigation with sparse rewards from sub-optimal trajectories [8]. As shown in Table 2, traditional behavior cloning methods (BC and DT) get 0 rewards on AntMaze medium and large environments. Our method shows excellent performance on all the tasks in AntMaze even with large complex maze settings and outperforms other methods by a margin. The result is not surprising because the entropy regularization incentivizes the policy to explore various sub-optimal trajectories within the dataset and stitch them to find a path toward the goal. In tasks with sparse rewards, this can be crucial because it prevents premature convergence to suboptimal deterministic policies. Additionally, employing the LCB of Q-ensembles effectively reduces the risk of taking low-value actions, enabling the development of robust policies.

In general, employing consistent hyperparameters for each domain, along with fixed entropy temperature α, LCB coefficient β, and ensemble size M across all tasks, our method not only achieves substantial overall performance but also outperforms prior works in the challenging AntMaze tasks. By the comparison with MSG[11] (Q-ensemble alone) and Diffusion-QL (Diffusion alone), our method further improves results demonstrating its effectiveness in handling complex environments with sparse rewards by effectively combining suboptimal trajectories to find better solutions via action space exploration.

4.3 Analysis and Discussion

We first study the core components of our method: entropy regularization and Q-ensemble. Then we show that adding both significantly improves the training robustness of diffusion-based policies.

Entropy Regularization The core idea of applying entropy regularization in offline RL is to increase the exploration of new actions such that the estimation of Q-functions is more accurate, especially for datasets with unbalanced action distribution such as the toy example in Figure 1. Here we report the results of training the diffusion policy with different entropy temperatures in Table 3. It is observed that our method with positive entropy coefficients performs better than that without the entropy term. In addition, we can extend our model with an automatic entropy adjustment

Table 3: Ablation study on entropy temperatures.

Entropy temperature α	0	0.01	0.05	0.1	auto
Antmaze-medium-play-v0	85.7	**91.4**	91	88.3	92.0
Antmaze-medium-diverse-v0	89.0	91.6	90.7	**93.5**	90.8
Antmaze-large-play-v0	77.7	81.2	78.3	**82**	82.0
Antmaze-large-diverse-v0	73.7	76.4	71.3	**78.3**	76.0
HalfCheetah-Medium-v2	53.7	54.9	54.0	**55.3**	54.2
Hopper-Medium-v2	94.8	94.2	93.3	**97.7**	94.0
Walker2D-Medium-v2	89.6	**92.5**	91.6	89.9	91.9
Pen-human-v1	60.9	67.2	63.6	69.8	**78.5**
Pen-cloned-v1	57.9	66.3	61.8	56.5	**79.8**
Kitchen-complete-v0	80.6	82.3	77.6	54.4	**84.4**
Kitchen-Mixed-v0	57.0	60.2	50.8	56.5	**60.4**

similar to the work in [16]. This approach is marked as "*auto*" in Table 3. The results show that auto-tuning the entropy temperature further improves the performance in the Adroit and Kitchen domains. Please refer to Appendix B.3 for more details.

Q-Ensembles We evaluate our method under different numbers of Q networks $M \in \{2, 4, 64\}$ in the AntMaze environment to explore the effectiveness of Q-ensembles. The results with average performance within 5 different seeds are provided in Table 4. The key observations are 1) As the M increases, the model gets better performance and the training process becomes more stable; 2) The standard deviation in the results decreases as M increases, suggesting larger ensembles not only perform better on

Table 4: Ablation experiments of our entropy-based diffusion policy with different ensemble sizes on selected AntMaze tasks.

Ensemble Size	2	4	64
Antmaze-medium-play-v0	84.0	87.2	**91.4**
Antmaze-medium-diverse-v0	71.8	87.2	**91.6**
Antmaze-large-play-v0	54.2	52.4	**81.2**
Antmaze-large-diverse-v0	43.2	69.0	**76.4**
Average	63.3	74.0	**85.2**

average but also provide more reliable and consistent results. 3) While increasing M from 2 to 4 shows a substantial improvement, the performance gains decrease with an even larger size. It is worth noting that other offline RL approaches like Diffusion-QL [44] also adopt two Q networks for training robustness. See Appendix B for more detailed results.

LCB coefficients β We evaluate our method with β values of 1, 2, and 4 on AntMaze-medium environments Figure 3 demonstrates that adjusting the LCB coefficient improves performance, partic-
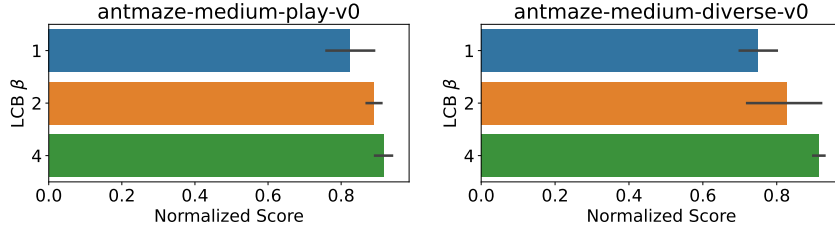
Figure 3: Ablation experiments of our method with different values of LCB coefficient $\beta = 1, 2, 4$ on AntMaze-Medium environments over 5 different random seeds.

Table 5: Computational time comparison with different settings on Antmaze-medium-play-v0. Training time is for 1 epoch (1000 training steps) and eval time is for 1000 RL steps.

Policy	Diffusion Step T	# Critics M	Training Time	Eval Time
Gaussian	1	2	5m 35s	1s 450ms
Gaussian	1	64	7m 20s	1s 450ms
Diffusion	5	2	9m 30s	4s 800ms
Diffusion	5	64	11m	4s 800ms
Diffusion	10	2	12m 23s	8s
Diffusion	10	64	13m 55s	8s

ularly for higher values, which helps in managing the exploration-exploitation trade-off effectively. In addition, the numerical results are provided in Appendix Table 8.

Training Stability and Computational Time   Empirically we observe that the training of diffusion policies is always unstable, particularly for sparse-reward environments such as AntMaze medium and large tasks. Our method alleviates this problem by incorporating the entropy regularization and Q-ensembles as stated in the introduction. Here, we further show the comparison of training Diffusion-QL and our method on four AntMaze tasks as illustrated in Figure 4, maintaining the same number of diffusion steps $T = 5$ for both. It is observed that the performance of Diffusion-QL even drops down as the training step increases, while our method is substantially more stable and achieves higher results throughout all the training processes. We also included a detailed comparison of training and evaluation times for Gaussian and diffusion policies with Q-ensembles in Table 5. Increasing M from 2 to 64 almost does not influence the evaluation time. The diffusion step T has more impact on both training and evaluation time which is a common problem in diffusion models.

5   Related Work

Generative Diffusion Models and Mean-reverting SDEs   Recent advancements have integrated diffusion models [18, 42, 41, 40] and SDEs [43, 32, 45, 39] for real-istic generative modeling. The development of Denoising Diffusion Probabilistic Models [18] showcases the ability of diffusion models to generate high-fidelity images through iterative reverse diffusion processes guided by deep neural networks, achieving state-of-the-art performance in generative tasks. In [33, 45, 39], mean-reverting SDEs are applied to speech processing and image restoration tasks. These SDEs, similar to (1) but with different parameters, ensure our policy adapts across various distributions without bias. The general applicability of our method is demonstrated in 4 D4RL benchmark domains. The comparison between our SDE and [43] are provided in Appendix A.6.
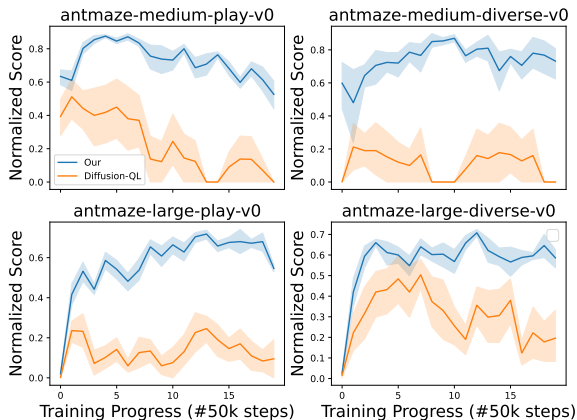

Figure 4: Learning curves of the Diffusion-QL and our method on selected Antmaze tasks over 5 random seeds.

9

Diffusion Models in Offline RL Diffusion models in offline RL have gained growing attention for their potent modeling capabilities. In Janner et al. [19], diffusion models are introduced as trajectory planners trained with offline datasets for guided sampling, significantly mitigating compounding errors in model-based planning [46]. Diffusion models are also used as data synthesizers [5, 48], generating augmented training data to enhance offline RL robustness. Additionally, diffusion models approximate behavior policies [44, 21, 17], integrating Q-learning for policy improvement, though this can lead to overly conservative policies.

Entropy Regularization In online RL, maximum entropy strategies encourage exploration by maximizing rewards while maintaining high entropy [14, 15]. This approach develops diverse skills [7] and adapts to unseen goals [38]. However, its application in offline RL is challenging due to the multi-modal nature of datasets from various policies and expert demonstrations.

Uncertainty Measurement Balancing exploration and exploitation is crucial when data is limited. Online RL methods like bootstrapped DQN [37] and Thompson sampling [30] estimate uncertainty for exploration guidance. In offline RL, handling uncertainty is critical due to the lack of environment interaction. Model-based methods like MOPO [47] and MORel [23] measure and penalize uncertain model dynamics. Similarly, model-free methods like EDAC [1] and MSG [11] use Q-network ensembles to obtain pessimistic value estimations for policy guidance.

6 Conclusion

In this work, we present an entropy-regularized diffusion policy for offline RL, introducing mean-reverting SDEs as the base framework to provide tractable entropy. Our theoretical contributions include deriving an approximated entropy for a diffusion model, enabling its integration as an entropy regularization component within the policy loss function. We also propose an optimal sampling process, ensuring the fast convergence of action generation from diffusion policy. Additionally, we enhance our method by incorporating Q-ensembles to handle the data uncertainty. Our experimental results show that combining entropy regularization with the LCB approach leads to a more robust policy, achieving state-of-the-art performance across offline RL benchmarks, particularly in AntMaze tasks with sparse rewards and suboptimal trajectories.

Future Work While the proposed method performs well on most D4RL tasks, the diffusion policy requires longer time when executed on compute- and power-constrained devices. Our future work will investigate real-time policy distillation under time and compute constraints to address this challenge.

Acknowledgements

References

[1] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

[2] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[3] Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhihong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. *arXiv preprint arXiv:2202.11566*, 2022.

[4] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

[5] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[7] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[8] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[9] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

[10] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[11] Kamyar Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters. *Advances in Neural Information Processing Systems*, 35:18267–18281, 2022.

[12] Daniel T Gillespie. Exact numerical simulation of the ornstein-uhlenbeck process and its integral. *Physical review E*, 54(2):2084, 1996.

[13] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR, 2017.

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[15] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[16] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[17] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[19] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[20] Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.

[21] Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *arXiv preprint arXiv:2305.20081*, 2023.

[22] Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *arXiv preprint arXiv:2305.20081*, 2023.

[23] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[26] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

[27] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

[28] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

[29] Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pages 45–73. Springer, 2012.

[30] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[31] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[32] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Image restoration with mean-reverting stochastic differential equations. *International Conference on Machine Learning*, 2023.

[33] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Refusion: Enabling large-size realistic image restoration with latent-space diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1680–1691, 2023.

[34] Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.

[35] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[36] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[37] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. *Advances in neural information processing systems*, 29, 2016.

[38] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[39] Julius Richter, Simon Welker, Jean-Marie Lemercier, Bunlong Lay, and Timo Gerkmann. Speech enhancement and dereverberation with diffusion-based generative models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[41] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[43] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[44] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

[45] Simon Welker, Julius Richter, and Timo Gerkmann. Speech enhancement with score-based generative models in the complex STFT domain. In *Proc. Interspeech 2022*, pages 2928–2932, 2022. doi: 10.21437/Interspeech.2022-10653.

[46] Chenjun Xiao, Yifan Wu, Chen Ma, Dale Schuurmans, and Martin Müller. Learning to combat compounding-error in model-based reinforcement learning. *arXiv preprint arXiv:1912.11206*, 2019.

[47] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

[48] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.

[49] Zhengbang Zhu, Hanye Zhao, Haoran He, Yichao Zhong, Shenyu Zhang, Yong Yu, and Weinan Zhang. Diffusion models for reinforcement learning: A survey. *arXiv preprint arXiv:2311.01223*, 2023.

[50] Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

Appendix

A Proof

A.1 Solution to the Forward SDE

Given the forward Stochastic Differential Equation (SDE) represented by

$$\mathrm{d}\boldsymbol{a} = -\theta_t \boldsymbol{a}\,\mathrm{d}t + \sigma_t\,\mathrm{d}\boldsymbol{w}, \quad \boldsymbol{a}^0 \sim p_0(\boldsymbol{a}), \tag{18}$$

where θ_t and σ_t are time-dependent positive functions, and $\boldsymbol{w}$ denotes a standard Wiener process. We consider the special case where $\sigma_t^2 = 2\theta_t$ for all t. The solution for the transition probability from time τ to t ($\tau < t$) is given by

$$p(\boldsymbol{a}^t|\boldsymbol{a}^\tau) = \mathcal{N}\left(\boldsymbol{a}^t|\boldsymbol{a}^\tau \mathrm{e}^{-\bar{\theta}_{\tau:t}}, (1 - \mathrm{e}^{-2\bar{\theta}_{\tau:t}})\boldsymbol{I}\right). \tag{19}$$

Proof. The proof is in general similar to that in IR-SDE [32]. To solve Equation (18), we introduce the transformation

$$\psi(\boldsymbol{a}, t) = \boldsymbol{a}\mathrm{e}^{\bar{\theta}_t}, \tag{20}$$

and apply Itô's formula to obtain

$$\mathrm{d}\psi(\boldsymbol{a}, t) = \sigma_t \mathrm{e}^{\bar{\theta}_t}\,\mathrm{d}\boldsymbol{w}(t). \tag{21}$$

Integrating from τ to t, we get

$$\psi(\boldsymbol{a}^t, t) - \psi(\boldsymbol{a}^\tau, \tau) = \int_\tau^t \sigma_z \mathrm{e}^{\bar{\theta}_z}\,\mathrm{d}\boldsymbol{w}(z), \tag{22}$$

we can analytically compute the two integrals as θ_t and $\boldsymbol{\sigma}_t$ are scalars and then obtain

$$\boldsymbol{a}(t)\mathrm{e}^{\bar{\theta}_t} - \boldsymbol{a}^\tau \mathrm{e}^{\bar{\theta}_\tau} = \int_\tau^t \sigma_z \mathrm{e}^{\bar{\theta}_z}\,\mathrm{d}\boldsymbol{w}(z). \tag{23}$$

Rearranging terms and dividing by $\mathrm{e}^{\bar{\theta}_t}$, we obtain

$$\boldsymbol{a}(t) = \boldsymbol{a}(\tau)\mathrm{e}^{-\bar{\theta}_{\tau:t}} + \int_\tau^t \sigma_z \mathrm{e}^{-\bar{\theta}_{z:t}}\,\mathrm{d}\boldsymbol{w}(z). \tag{24}$$

The integral term is actually a Gaussian random variable with mean zero and variance

$$\int_\tau^t \sigma_z^2 \mathrm{e}^{-2\bar{\theta}_{z:t}}\,\mathrm{d}z = \lambda^2(1 - \mathrm{e}^{-2\bar{\theta}_{\tau:t}}), \tag{25}$$

under the condition $\sigma_t^2 = 2\theta_t$. Thus, the transition probability is

$$p(\boldsymbol{a}^t|\boldsymbol{a}^\tau) = \mathcal{N}(\boldsymbol{a}^t|\boldsymbol{a}^\tau \mathrm{e}^{-\bar{\theta}_{\tau:t}}, (1 - \mathrm{e}^{-2\bar{\theta}_{\tau:t}})\boldsymbol{I}). \tag{26}$$

This completes the proof. □

Loss function From (19), the marginal distribution of $p(\boldsymbol{a}(t))$ can be written as

$$\begin{aligned} p(\boldsymbol{a}(t)) &= p(\boldsymbol{a}(t) \mid \boldsymbol{a}(0)) \\ &= \mathcal{N}(\boldsymbol{a}(t) \mid \boldsymbol{a}(0)\mathrm{e}^{-\bar{\theta}_t}, (1 - \mathrm{e}^{-2\bar{\theta}_t})\boldsymbol{I}). \end{aligned} \tag{27}$$

where we substitute $\bar{\theta}_{0:t}$ with $\bar{\theta}_t$ for clear notation. During training, the initial diffusion state $\boldsymbol{a}^0$ is given and thus we can obtain the ground truth score $\nabla_{\boldsymbol{a}} \log p_t(\boldsymbol{a})$ based on the marginal distribution:

$$\nabla_{\boldsymbol{a}} \log p_t(\boldsymbol{a} \mid \boldsymbol{a}_0) = -\frac{\boldsymbol{a}_t - \boldsymbol{a}^0 \mathrm{e}^{-\bar{\theta}_t}}{1 - \mathrm{e}^{-2\bar{\theta}_t}}, \tag{28}$$

which can be approximated using a neural network and optimized with score-matching loss. Moreover, the marginal distribution (27) gives the reparameterization of the state:

$$\boldsymbol{a}^t = \boldsymbol{a}^0 \mathrm{e}^{-\bar{\theta}_t} + \sqrt{1 - \mathrm{e}^{-2\bar{\theta}_t}} \cdot \boldsymbol{\epsilon}_t, \tag{29}$$

14

where ϵ_t is a standard Gaussian noise $\epsilon_t \sim \mathcal{N}(0, \boldsymbol{I})$. By substituting (29) into (28), the score function can be re-written in terms of the noise as

$$\nabla_{\boldsymbol{a}} \log p_t(\boldsymbol{a} \mid \boldsymbol{a}_0) = -\frac{\epsilon_t}{\sqrt{1 - \mathrm{e}^{-2\bar{\theta}_t}}}. \tag{30}$$

Then we follow the practical settings in diffusion models [18, 6] to estimate the noise with a time-dependent neural network ϵ_ϕ and optimize it with a simplified noise matching loss:

$$L(\phi) := \mathbb{E}_{t \in [0,T]} \left[\left\| \epsilon_\phi(\boldsymbol{a}^0 \mathrm{e}^{-\bar{\theta}_t} + \sqrt{1 - \mathrm{e}^{-2\bar{\theta}_t}} \cdot \epsilon_t, t) - \epsilon_t) \right\| \right], \tag{31}$$

where t is a randomly sampled timestep and $\{\boldsymbol{a}_t\}_{t=0}^T$ denotes the discretization of the diffusion process. And this loss (31) is the same as (5) in the main paper.

A.2 Sampling from the Posterior

Proposition 3.1. *Given an initial variable $\boldsymbol{a}^0$, for any diffusion state $\boldsymbol{a}^t$ at time $t \in [1, T]$, the posterior of the mean-reverting SDE (1) conditioned on $\boldsymbol{a}^0$ is*

$$p(\boldsymbol{a}^{t-1} \mid \boldsymbol{a}^t, \boldsymbol{a}^0) = \mathcal{N}(\boldsymbol{a}^{t-1} \mid \tilde{\mu}_t(\boldsymbol{a}^t, \boldsymbol{a}^0), \, \tilde{\beta}_t \mathbf{I}), \tag{32}$$

which is a Gaussian with mean and variance given by:

$$\tilde{\mu}_t(\boldsymbol{a}^t, \boldsymbol{a}^0) := \frac{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}}{1 - \mathrm{e}^{-2\bar{\theta}_t}} \mathrm{e}^{-\theta_t} \boldsymbol{a}^t + \frac{1 - \mathrm{e}^{-2\theta_t}}{1 - \mathrm{e}^{-2\bar{\theta}_t}} \mathrm{e}^{-\bar{\theta}_{t-1}} \boldsymbol{a}^0$$

$$\text{and} \quad \tilde{\beta}_t := \frac{(1 - \mathrm{e}^{-2\bar{\theta}_{t-1}})(1 - \mathrm{e}^{-2\theta'_t})}{1 - \mathrm{e}^{-2\bar{\theta}_t}}, \tag{33}$$

where $\theta'_i := \int_{i-1}^i \theta_t dt$ and $\bar{\theta}_t$ is to substitute $\bar{\theta}_{0:t}$ for clear notation.

Proof. The posterior of SDE can be derived from Bayes' rule,

$$p(\boldsymbol{a}^{t-1} \mid \boldsymbol{a}^t, \boldsymbol{a}^0) = \frac{p(\boldsymbol{a}^t \mid \boldsymbol{a}^{t-1}, \boldsymbol{a}^0) p(\boldsymbol{a}^{t-1} \mid \boldsymbol{a}^0)}{p(\boldsymbol{a}^t \mid \boldsymbol{a}^0)}. \tag{34}$$

Recall that the transition distribution $p(\boldsymbol{a}^t \mid \boldsymbol{a}^{t-1})$ and $p(\boldsymbol{a}^t \mid \boldsymbol{a}_0)$ can be known with the solution to the forward SDE. Since all the distributions are Gaussian, the posterior will also be a Gaussian.

$p(\boldsymbol{a}^{t-1} \mid \boldsymbol{a}^t, \boldsymbol{a}^0)$

$$\propto \exp\left(-\frac{1}{2}\left(\frac{(\boldsymbol{a}^t - \boldsymbol{a}^{t-1}\mathrm{e}^{-\theta'_t})^2}{1 - \mathrm{e}^{-2\theta'_t}} + \frac{(\boldsymbol{a}^{t-1} - \boldsymbol{a}^0 \mathrm{e}^{-\bar{\theta}_{t-1}})^2}{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}} - \frac{(\boldsymbol{a}^t - \boldsymbol{a}^0 \mathrm{e}^{-\bar{\theta}_t})^2}{1 - \mathrm{e}^{-2\bar{\theta}_t}}\right)\right)$$

$$= \exp\left(-\frac{1}{2}\left((\frac{\mathrm{e}^{-2\theta'_t}}{1 - \mathrm{e}^{-2\theta'_t}} + \frac{1}{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}})(\boldsymbol{a}^{t-1})^2 - (\frac{2\mathrm{e}^{-\theta'_t}}{1 - \mathrm{e}^{-2\theta'_t}}\boldsymbol{a}^t + \frac{2\mathrm{e}^{-\bar{\theta}_{t-1}}}{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}}\boldsymbol{a}^0)\boldsymbol{a}^{t-1} + C(\boldsymbol{a}^t, \boldsymbol{a}^0)\right)\right) \tag{35}$$

where $C(\boldsymbol{a}^t, \boldsymbol{a}^0)$ is some function not involving $(a^{t-1})^2$. With the standard Gaussian density function, the mean and the variance can be computed:

$$\tilde{\mu}_t(\boldsymbol{a}^t, \boldsymbol{a}^0) := \frac{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}}{1 - \mathrm{e}^{-2\bar{\theta}_t}} \mathrm{e}^{-\theta'_t} \boldsymbol{a}^t + \frac{1 - \mathrm{e}^{-2\theta'_t}}{1 - \mathrm{e}^{-2\bar{\theta}_t}} \mathrm{e}^{-\bar{\theta}_{t-1}} \boldsymbol{a}^0$$

$$\text{and} \quad \tilde{\beta}_t := \frac{(1 - \mathrm{e}^{-2\bar{\theta}_{t-1}})(1 - \mathrm{e}^{-2\theta'_t})}{1 - \mathrm{e}^{-2\bar{\theta}_t}}. \tag{36}$$

Thus we complete the proof. $\qquad\square$

A.3 Optimal Reverse Path

In addition, we can prove the distribution mean $\tilde{\mu}_t(\boldsymbol{a}^t, \boldsymbol{a}^0)$ is the optimal reverse path from $\boldsymbol{a}^t$ to $\boldsymbol{a}^{t-1}$.

Proof. As stated in Proposition 3.1, the posterior is a Gaussian distribution and can be derived by Bayes' rule. Thus it is natural to find the optimal reverse path by minimizing the negative log-likelihood according to

$$\boldsymbol{a}_*^{t-1} = \arg\min_{\boldsymbol{a}^{t-1}} \left[-\log p(\boldsymbol{a}^{t-1} \mid \boldsymbol{a}^t, \boldsymbol{a}^0) \right]. \tag{37}$$

From (34), we have

$$-\log p(\boldsymbol{a}^{t-1} \mid \boldsymbol{a}^t, \boldsymbol{a}^0) \propto -\log p(\boldsymbol{a}^i \mid \boldsymbol{a}^{t-1}, \boldsymbol{a}^0) - \log p(\boldsymbol{a}^{t-1} \mid \boldsymbol{a}^0) \tag{38}$$

Then we can directly solve (37) by computing the gradient of the negative log-likelihood and setting it to 0:

$$\nabla_{\boldsymbol{a}_*^{t-1}} \left\{ -\log p(\boldsymbol{a}_*^{t-1} \mid \boldsymbol{a}^t, \boldsymbol{a}^0) \right\} \propto -\nabla_{\boldsymbol{a}_*^{t-1}} \log p(\boldsymbol{a}^i \mid \boldsymbol{a}_*^{t-1}, \boldsymbol{a}^0) - \nabla_{\boldsymbol{a}_*^{t-1}} \log p(\boldsymbol{a}_*^{t-1} \mid \boldsymbol{a}^0)$$

$$= -\frac{\mathrm{e}^{-\theta_t'}(\boldsymbol{a}^t - \boldsymbol{a}_*^{t-1}\mathrm{e}^{-\theta_t'})}{1 - \mathrm{e}^{-2\theta_t'}} + \frac{\boldsymbol{a}_*^{t-1} - \boldsymbol{a}^0\mathrm{e}^{-\bar{\theta}_{t-1}}}{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}}$$

$$= \frac{\boldsymbol{a}_*^{t-1}\mathrm{e}^{-2\theta_t'}}{1 - \mathrm{e}^{-2\theta_t'}} + \frac{\boldsymbol{a}_*^{t-1}}{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}} - \frac{\boldsymbol{a}^i\mathrm{e}^{-\theta_t'}}{1 - \mathrm{e}^{-2\theta_t'}} - \frac{\boldsymbol{a}_0\mathrm{e}^{-\bar{\theta}_{t-1}}}{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}}$$

$$= \frac{\boldsymbol{a}_*^{t-1}(1 - \mathrm{e}^{-2\bar{\theta}_i})}{(1 - \mathrm{e}^{-2\theta_t'})(1 - \mathrm{e}^{-2\bar{\theta}_{t-1}})} - \frac{\boldsymbol{a}^i\mathrm{e}^{-\theta_t'}}{1 - \mathrm{e}^{-2\theta_t'}} - \frac{\boldsymbol{a}_0\mathrm{e}^{-\bar{\theta}_{t-1}}}{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}} = 0. \tag{39}$$

Since (39) is linear, we get

$$\boldsymbol{a}_*^{t-1} = \frac{1 - \mathrm{e}^{-2\bar{\theta}_{t-1}}}{1 - \mathrm{e}^{-2\bar{\theta}_t}} \mathrm{e}^{-\theta_t'} \boldsymbol{a}^t + \frac{1 - \mathrm{e}^{-2\theta_t'}}{1 - \mathrm{e}^{-2\bar{\theta}_t}} \mathrm{e}^{-\bar{\theta}_{t-1}} \boldsymbol{a}^0. \tag{40}$$

$\square$

This completes the proof. Note that the second-order derivative is a positive constant, and thus $\boldsymbol{a}_*^{t-1}$ is the optimal point. And we find that this optimal reverse path is the same as our posterior distribution mean as shown in Proposition 3.1.

A.4 Entropy Approximation

Proposition 3.1 further shows that the conditional posterior from $\boldsymbol{a}_i^1$ to $\boldsymbol{a}_i^0$ is Gaussian, meaning that

$$-\log \pi_\phi(\boldsymbol{a}_i^0 \mid \boldsymbol{s}_i) = -\log \pi_\phi(\boldsymbol{a}_i^1 \mid \boldsymbol{s}_i) + \mathcal{C}, \tag{41}$$

where $\mathcal{C}$ is a constant and $\boldsymbol{a}_i^1$ can be approximated using

$$p(\boldsymbol{a}^t \mid \boldsymbol{a}^\tau) = \mathcal{N}(\boldsymbol{a}^t \mid \boldsymbol{a}^\tau \mathrm{e}^{-\bar{\theta}_{\tau:t}}, (1 - \mathrm{e}^{-2\bar{\theta}_{\tau:t}})\boldsymbol{I}), \tag{42}$$

Proof. Let's consider **sequentially sampled action states** a_i^0, a_i^1 from our optimal sampling strategy. Then we have

$$\pi(a_i^0 | s_i) = \pi(a_i^0 | a_i^1, \hat{a}_i^0) \cdot \pi(a_i^1 | s_i), \tag{43}$$

where $\hat{a}_i^0$ is the approximated action's initial state from Eq. (8). Proposition 3.1 shows that the conditional posterior from a_i^1 to a_i^0 is a Gaussian with **certain mean and variance**, meaning that the term $\pi(a_i^0 | a_i^1, \hat{a}_i^0)$ is a computable constant and thus we can write

$$-\log \pi(a_i^0 | s_i) = -\log \pi(a_i^0 | a_i^1, \hat{a}_i^0) - \log \pi(a_i^1 | s_i) = -\log \pi(a_i^1 | s_i) + \mathcal{C}. \tag{44}$$

$\square$

A.5 Visualization

For a more intuitive explanation of our approach, Figure 5 outlines the forward and reverse processes of the mean-reverting SDE used for action prediction.
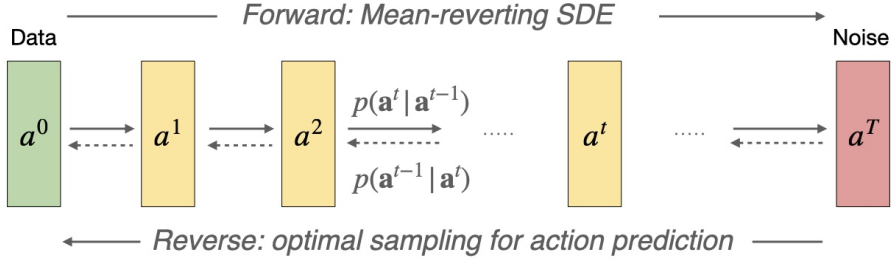

Figure 5: Visualization of the workings of the mean-reverting SDE for action prediction. The SDE models the degradation process from the action from the dataset to a noise. By guiding the policy with corresponding reverse-time SDE and the LCB of Q, a new action is generated conditioned on the RL state.

A.6 Comparison to VP SDE

Our mean-reverting SDE is derived from the well-known Ornstein-Uhlenbeck (OU) process [12] which has the following form:

$$\mathrm{d}x = \theta(\mu - x)\mathrm{d}t + \sigma\mathrm{d}w.$$

As $t \to \infty$, its marginal distribution $p_t(x)$ converges to a stationary Gaussian with mean value μ, which explains the name: "mean-reverting". We assume that there is no prior knowledge of the actions and thus set $\mu = 0$ to generate actions from standard Gaussian noise. Then, with $\mu = 0$, the mean-reverting SDE has the same form as VP SDE. However, in [43], no solution of the continuous-time SDE was given. The authors start from perturbing data with multiple noise scales and generalize this idea with an infinite number of noise scales which makes the perturbed data distributions evolve according to an SDE. They keep using the solution of DDPM while we use Itô's formula to solve the continuous-time SDE. Compared to the original VP SDE, our mean-reverting SDE is analytically tractable, see (2) and thus its score $\nabla_x \log p_t(x)$ is easier to learn. More importantly, the solution of the mean-reverting SDE can be used for entropy approximation.

B Additional Experiments Details

B.1 More experiments on proposed optimal sampling with different sample step

We added the figures of data generation with fewer steps ($T = 1$ and $T = 2$) for the toy task in Section 3.1. The results show that the optimal sampling strategy significantly outperforms the reverse-time SDE in all steps, further demonstrating the efficiency and effectiveness of our method.

B.2 Hyperparameters

As stated in Section 4.1, we keep our key hyperparameters, entropy weight $\alpha = 0.01$, ensemble size $M = 64$, LCB coefficient $\beta = 4$ and diffusion steps $T = 5$ for all tasks in different domains. As for others related to our algorithm, we consider the policy learning rate, Q-learning weight η, and whether to use max Q backup. For implementation details, we consider the gradient clip norm, diffusion loss type, and whether to clip action at every diffusion step. We keep the hyperparameter same for tasks in the same domain except for the AntMaze domain. We use max Q backup [28] for complete tasks. The hyperparameter settings are shown in Table 6.

B.3 Automating Entropy Adjustment

It is possible to consider automating entropy adjustment similar to [15] but α depends on the state since the offline dataset is pre-collected and may be imbalanced across different states shown in
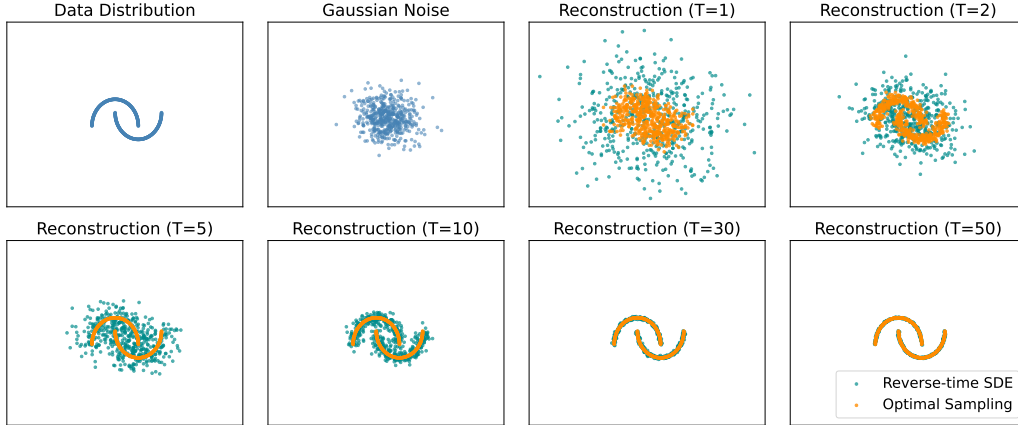
Figure 6: The proposed optimal sampling with different sample steps.

Table 6: Hyperparameter settings of all selected tasks. '*' means all the AntMaze tasks use max Q-backup trick [28] except the 'antmaze-umaze-v0' task as the same as that in other papers. The 'likelihood' loss is proposed in IR-SDE [32] which forces the model to learn optimal reverse paths from a^t to a^{t-1}.

Tasks domain	learning rate	η	max Q-backup	gradient norm	loss type	action clip	T
Gym	3e-4	1.0	False	4.0	Likelihood	False	5
AntMaze	3e-4	2.0	True*	4.0	Noise	True	5
Adroit	3e-5	0.1	False	8.0	Noise	True	10
Kitchen	3e-4	0.005	False	10.0	Likelihood	False	10

Figure 7. One way to compute the gradients for state-depend α is

$$J(\alpha) = \mathbb{E}_{s_i \sim \mathcal{D}, a_i \sim \pi_\phi}[-\alpha(s_i) \log \pi_\phi(a_i|s_i) - \alpha(s_i)\bar{\mathcal{H}}], \tag{45}$$

where $\alpha(s_i)$ is implemented as a neural network with a single hidden layer consisting of 32 units and $\bar{\mathcal{H}}$ is a desired minimum expected entropy which represents the desired level of exploration which can be set as a function of action space dimension without the need for extensive hyperparameter tuning across different tasks.

B.4 More Analysis for Q-Ensembles

Here, we provide more detailed experiments for analyzing the effect of ensemble sizes M as we discussed in Section 4.3. More specifically, the results of different ensemble sizes are reported in Table 7 and Figure 8, in which we also provide the variance that further shows the robustness of our method.

Table 7: Ablation study of ensemble size M on selected AntMaze tasks.

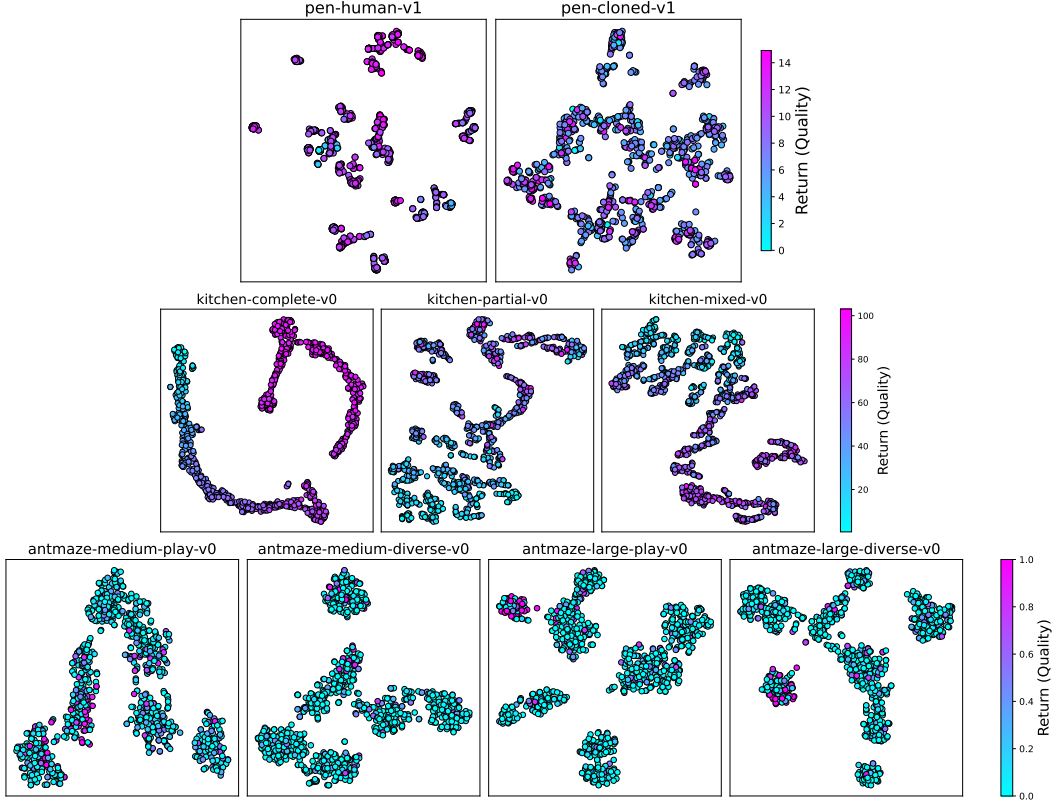| Ensemble Size	$M = 1$	$M = 2$	$M = 4$	$M = 16$	$M = 64$
antmaze-medium-play-v0	50.2 ± 26.4	84.0 ± 5.0	87.2 ± 1.1	83.6 ± 7.7	$\mathbf{91.4} \pm 1.5$
antmaze-medium-diverse-v0	67.2 ± 7.6	71.8 ± 14.1	87.2 ± 3.8	88.0 ± 2.2	$\mathbf{91.6} \pm 2.3$
antmaze-large-play-v0	48.2 ± 10.8	54.2 ± 10.0	52.4 ± 13.0	71.8 ± 5.8	$\mathbf{81.2} \pm 3.0$
antmaze-large-diverse-v0	58.8 ± 11.4	43.2 ± 16.1	69.0 ± 8.3	76.4 ± 8.47	$\mathbf{76.4} \pm 2.1$
Average	56.1	63.3	74.0	80	**85.2**

Figure 7: A t-SNE visualization of randomly selected 1000 states from Antmaze, Adroit and Kitchen domain. The color coding represents the return of the trajectory associated with each state.
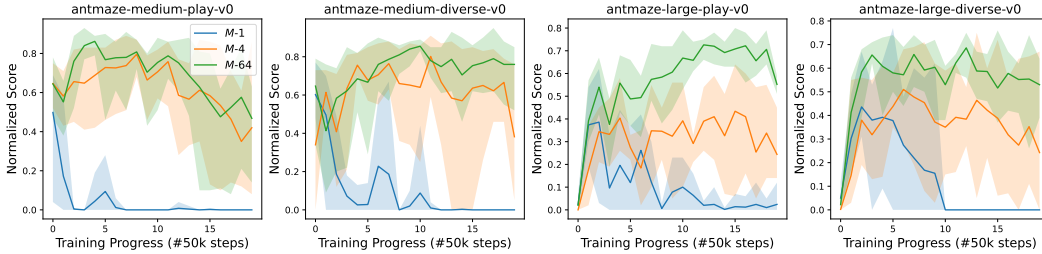

Figure 8: Abaltion study of Q-ensemble size M on selected AntMaze tasks. We consider $M \in \{1, 4, 64\}$ and we found size 64 is the best overall tasks.

B.5 Ablation study on LCB coefficients β

To explore the impact of different LCB coefficients β. We add an experiment of our method with β values of 1, 2, and 4 on AntMaze-medium environments. Figure 3 demonstrates that adjusting the LCB coefficient improves performance, particularly for higher values, which helps in managing the exploration-exploitation trade-off effectively. In addition, the numerical results are provided in Table 8.

B.6 Ablation study on Diffusion step T

We evaluated the impact of varying the number of diffusion steps on a range of tasks, including AntMaze, Gym, and Kitchen in Table 9. Our findings indicate that while increasing the number of steps generally improves performance, five steps provide the best balance across different tasks and

Table 8: Ablation study of LCB coefficients β.

LCB Coefficient β	1	2	4
Antmaze-medium-play-v0	82.4 ± 4.9	88.6 ± 1.5	$\mathbf{91.6}\pm2.3$
Antmaze-medium-diverse-v0	74.6 ± 3.7	84.0 ± 7.8	$\mathbf{91.4}\pm1.5$
Average	78.5	86.3	**91.5**

between performance and computational time in Gym and Antmaze tasks. For more complex tasks as in Kitchen and Pen, we choose $T = 10$.

Table 9: Ablation study of diffusion step T.

Diffusion Step T	3	5	10
Halfcheetah-medium-replay-v2	43.4	**57.0**	49.5
Hopper-medium-replay-v2	39.4	**102.7**	101.7
Walker2d-medium-replay-v2	51.2	94.2	**98.1**
Antmaze-medium-play-v0	**96.6**	91.6	90.2
Antmaze-medium-diverse-v0	**95.8**	91.4	83.8
Antmaze-large-play-v0	67.6	**81.2**	63.2
Antmaze-large-diverse-v0	**81.0**	76.4	70.0
pen-human-v1	65.4	67.2	**70.0**
pen-cloned-v1	67.3	66.3	**68.4**
Kitchen-complete-v0	7.5	82.3	**92.7**
Kitchen-partial-v0	10.9	60.3	**66.3**
Kitchen-mixed-v0	4.8	60.2	**68.0**
Average	52.6	**77.6**	76.8

B.7 Ablation study on Max Q-back

We conducted experiments with and without max Q-backup on AntMaze tasks in Table 10. The inclusion of max Q-backup significantly enhances performance, particularly in more complex environments (e.g., Antmaze-large).

Table 10: Ablation study of "Max Q trick".

Max Q-backup	True	False
Antmaze-medium-play-v0	$\mathbf{91.6}\pm2.3$	89.2 ± 2.9
Antmaze-medium-diverse-v0	$\mathbf{91.4}\pm1.5$	87.6 ± 1.8
Antmaze-large-play-v0	$\mathbf{81.2} \pm3.0$	22.3 ± 7.1
Antmaze-large-diverse-v0	$\mathbf{76.4} \pm2.1$	26.5 ± 6.1

B.8 Offline vs Online Model Selection

We use the online experience to evaluation our model during training. Table 11 presents a comparison of our method with Diffusion-QL, including both online and offline results. Additionally, we include our method's performance based on offline selection using the BC Loss criterion, selecting the step where the difference between consecutive steps was less than 4e-3.

Table 11: Performance comparison with online model selection and offline model selection.

AntMaze Tasks	Diffusion-QL (Offline)	DIffusion-QL (Online)	Ours (Offline)	Ours (Online)
antmaze-umaze-v0	93.4	96.0	99.0	**100.0**
antmaze-umaze-diverse-v0	66.2	**84.0**	67.5	79.8
antmaze-medium-play-v0	77.6	79.8	84.0	**91.4**
antmaze-medium-diverse-v0	78.6	82.0	85.4	**91.6**
antmaze-large-play-v0	46.6	49.0	72.6	**81.2**
antmaze-large-diverse-v0	56.6	61.7	65.9	**76.4**
Average	69.6	75.4	79.2	**86.7**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly describe the introduction of mean-reverting SDEs for entropy-regularized diffusion policies, the integration of Q-ensembles for robust policy improvement, and the empirical validation through performance on D4RL benchmarks, which are thoroughly discussed and validated in the subsequent sections of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: See Appendix A for more details.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The code is provided in supplementary materials and additional experiments Details can be found in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code with a README file is provided.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experiment setting and main hyperparameters are shown in Section 4 and additional experiments Details can be found in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our results are reported averaging by 5 random seeds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The computer resources are shown in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The dataset used in the paper is public without privacy-related data.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: We use the dataset D4RL for training offline RL agents and cite the original paper in Section 4.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.