Energy-Guided Diffusion Sampling for Offline-to-Online Reinforcement Learning

Xu-Hui Liu [*1] **Tian-Shuo Liu** [*1 2] **Shengyi Jiang** [3] **Ruifeng Chen** [1 2] **Zhilong Zhang** [1 2] **Xinwei Chen** [2] **Yang Yu** [1 2]

Abstract

Combining offline and online reinforcement learning (RL) techniques is indeed crucial for achieving efficient and safe learning where data acquisition is expensive. Existing methods replay offline data directly in the online phase, resulting in a significant challenge of data distribution shift and subsequently causing inefficiency in online fine-tuning. To address this issue, we introduce an innovative approach, **E**nergy-guided **DI**ffusion **S**ampling (EDIS), which utilizes a diffusion model to extract prior knowledge from the offline dataset and employs energy functions to distill this knowledge for enhanced data generation in the online phase. The theoretical analysis demonstrates that EDIS exhibits reduced suboptimality compared to solely utilizing online data or directly reusing offline data. EDIS is a plug-in approach and can be combined with existing methods in offline-to-online RL setting. By implementing EDIS to off-the-shelf methods Cal-QL and IQL, we observe a notable 20% average improvement in empirical performance on MuJoCo, AntMaze, and Adroit environments. Code is available at https://github.com/liuxhym/EDIS.

1. Introduction

Reinforcement learning (RL) (Sutton & Barto, 2018) has demonstrated remarkable efficacy in diverse decision-making tasks, such as sequential recommendation systems (Wang et al., 2018; Zhao et al., 2018), robotic locomotion skill learning (Peng et al., 2020; Haarnoja et al., 2018) and operations research (Hubbs et al., 2020; Wang et al., 2023c; 2024; Ling et al., 2024). Notably, RL methods, including online (Mnih et al., 2015; Liu et al., 2021; Chen et al., 2024) and offline (Fujimoto et al., 2019; Kumar et al., 2020; Jin et al., 2022; Jia et al., 2024) ones, have been employed to achieve superior performance. However, online RL methods often demand extensive data collection through interacting with the environment, a process that can be time-consuming or risky. In contrast, offline RL methods utilize pre-existing datasets to train a new policy, avoiding the need for resource-intensive online interactions but often suffering from suboptimality due to limited data.

To mitigate the challenges incurred due to data limitations in online and offline RL, researchers have proposed the offline-to-online setting, aiming to overcome the cost of online interaction and the suboptimality inherent in offline learning. In this setting, the agent undergoes two learning phases: first, it learns from an offline dataset, and then it fine-tunes through a limited number of online interactions (Lee et al., 2021; Kostrikov et al., 2022; Nakamoto et al., 2023). After the offline phase, value functions or policies are derived and used to initialize the online phase. Therefore, the algorithm can utilize prior knowledge in the offline dataset to help reduce the cost of the online phase. However, a significant drawback in this paradigm arises from the incomplete utilization of the offline dataset, as the information extracted is limited to the pre-trained policy.

While previous works have effectively addressed the performance drop because of the objective mismatch of the two-stage learning in offline-to-online settings (Nakamoto et al., 2023; Lee et al., 2021; Nair et al., 2020), the incomplete utilization of the prior knowledge in the offline dataset remains an unexplored challenge. Some studies directly replayed samples in the online phase for data augmentation (Lee et al., 2021), leading to performance improvements. Nevertheless, this approach neglects the distribution shift issue, as the data distribution in the offline dataset may differ from that induced by the current policy. Distribution shift has shown adverse effects even in off-policy reinforcement learning (Sinha et al., 2022; Liu et al., 2021),

[*]Equal contribution [1]National Key Laboratory for Novel Software Technology, Nanjing University, China & School of Artificial Intelligence, Nanjing University, China [2]Polixir Technologies [3]Department of Computer Science, The University of Hong Kong, Hong Kong, China. Correspondence to: Yang Yu <yuy@nju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

which incorporates a replay buffer, let alone when utilizing an offline dataset. Consequently, a fundamental question arises:

Is it feasible to generate samples without distribution shift based on an offline dataset?

Model-based methods (Wang et al., 2022; Rafailov et al., 2023; Chen et al., 2023b) generate such samples by first learning a model and using it for rollout. However, this style of generation inevitably suffers from the issue of compounding errors: the error of the learned model itself will accumulate during the multi-step rollout. Therefore, we aim to generate samples with the desired distribution directly. Motivated by the recent achievements of the diffusion model in image generation (Ho et al., 2020; Karras et al., 2022), we want to explore its potential application in RL sample generation. However, utilizing a diffusion model trained on an offline dataset introduces a challenge—it can only generate samples adhering to the dataset distribution, thus still being susceptible to distribution shift issues.

The desired distribution for RL has three crucial characteristics: i) the state distribution should align with that in the online training phase, ii) actions should be consistent with the current policy, and iii) the next states should conform to the transition function. To achieve this, we formulate three distinct energy functions to guide the diffusion sampling process, ensuring alignment with the aforementioned features. The developed algorithm is named as **E**nergy-Guided **DI**ffusion **S**ampling (EDIS). By presenting the new algorithm, we showcase the feasibility of generating additional, useful samples while mitigating distribution shift, which will effectively help leverage the offline dataset.

The theoretical analysis shows that EDIS exhibits reduced suboptimality compared to solely utilizing online data or directly replaying offline data. Additionally, it circumvents the compounding error issue commonly associated with model-based methods. When incorporating EDIS into existing methods, Cal-QL (Nakamoto et al., 2023) and IQL (Kostrikov et al., 2022), we observe 20% performance improvement on average across MuJoCo, AntMaze, and Adroit environments. Furthermore, our experiments demonstrate that purely model-based methods fail to achieve performance improvements due to the compounding errors in the model during rollouts, leading to an inaccurate state distribution. The ablation study further validates the effectiveness of the energy functions to guide the diffusion models.

2. Preliminaries

2.1. Markov Decision Process and RL

Let $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$ be an MDP, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function ($\Delta(\cdot)$ is the probability simplex), $r : \mathcal{S} \times \mathcal{A} \to [0, R_{\max}]$ is the reward function, $\gamma \in [0, 1)$ is the discount faction and ρ_0 is the initial distribution over states. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ describes a distribution over actions for each state. The goal of RL is to learn the best policy π^* that maximizes cumulative discounted reward, i.e., $\sum_t \mathbb{E}_{a_t \sim \pi^*} \gamma^t r(s_t, a_t)$. The value function and Q function of policy π are $V^\pi(s) = \sum_t \mathbb{E}_{a_t \sim \pi(s_t)}[\gamma^t r(s_t, a_t) | s_0 = s]$, $Q^\pi(s, a) = \sum_t \mathbb{E}_{a_t \sim \pi(s_t)}[\gamma^t r(s_t, a_t) | s_0 = s, a_0 = a]$. V^* and Q^* be the shorthand for V^{π^*} and Q^{π^*} respectively. To facilitate later analysis, we introduce the *discounted stationary state distribution* $d^\pi(s) = \sum_t \gamma^t \Pr(s_t = s; \pi)$ and the *discounted stationary state-action distribution* $d^\pi(s, a) = \sum_t \gamma^t \Pr(s_t = s, a_t = a; \pi)$. There are, in general, two learning paradigms of RL: online RL, where the agent can learn from interacting with the environment; and offline RL, where the agent can only learn from a given dataset $\mathcal{D} = \{(s, a, r, s')\}$, possibly collected by another policy.

2.2. Generative Modeling by Diffusion Model

Diffusion models (Ho et al., 2020; Karras et al., 2022) are a class of generative models inspired by non-equilibrium thermodynamics. Given a dataset $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$ with N samples of D-dimensional random variable $\boldsymbol{x}$ from an unknown data distribution $p_0(\boldsymbol{x}_0)$. Diffusion models gradually add Gaussian noise at time 0 to T according to noise levels $\sigma_{\max} = \sigma_T > \sigma_{T-1} > \cdots > \sigma_0 = 0$ so that at each noise level, $\boldsymbol{x}_t^{(i)} \sim p(\boldsymbol{x}_t; \sigma_t)$. The distribution of the endpoint $\boldsymbol{x}_T^{(i)} \sim \mathcal{N}(0, \sigma_{\max}^2 \boldsymbol{I})$ is indistinguishable from pure Gaussian noise, while the starting point $\boldsymbol{x}_0^{(i)}$ aligns with the distribution of the original dataset. Then the reverse process starts from a Gaussian distribution and iteratively denoises samples using the trained model, ultimately recovering the target distribution. The process can be interpreted as *SDE*s:

$$\begin{aligned} \mathrm{d}\boldsymbol{x}_\pm = &-\dot{\sigma}(t)\sigma(t)\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; \sigma(t))\mathrm{d}t \\ &\pm \beta(t)\sigma^2(t)\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; \sigma(t))dt + \sqrt{2\beta(t)}\sigma(t)\mathrm{d}\omega_t, \end{aligned} \tag{1}$$

where ω_t is a standard Wiener process, $d\boldsymbol{x}_+$ and $d\boldsymbol{x}_-$ are separate SDEs for moving forward and backward in time. The only unknown term is the *score function* $\nabla_{\boldsymbol{x}} \log p_t(\cdot)$. Karras et al. (2022) considers training a denoiser $D_\theta(\boldsymbol{x}_t; \sigma)$ on an L2 denoising minimization objective:

$$L_{\text{VLB}}(\theta) := \mathbb{E}_{\boldsymbol{x} \sim p_0(\boldsymbol{x}_0), \epsilon \sim \mathcal{N}(0, \sigma^2 I)} \left\| D_\theta(\boldsymbol{x} + \epsilon; \sigma) - \boldsymbol{x} \right\|_2^2, \tag{2}$$

where σ is the standard deviation of Gaussian noise. After training a diffusion model for estimating the score function with $\nabla_{\boldsymbol{x}} \log p_0(\boldsymbol{x}; \sigma) = (D_\theta(\boldsymbol{x}; \sigma) - \boldsymbol{x})/\sigma^2$, we can fastly generate samples by solving the backward *SDE*.

3. EDIS: Energy-Guided Diffusion Sampling

To extract prior knowledge from the offline dataset and generate samples to conform to the online data distribution, we introduce our innovative approach, named **E**nergy-guided **DI**ffusion **S**ampling (EDIS). At the heart of our method is to accurately generate a desired online data distribution, denoted as $q_\pi(s, a, s')$, from pre-gathered data. The distribution does not include reward r because we assume that the reward function $r(s, a)$ is accessible, either directly or through learning from the dataset. To achieve this, we have integrated a diffusion model into our framework, capitalizing on its exceptional capability for modeling complex distributions.

3.1. Distribution Adjustment via Energy Guidance

One challenge in this process is the inherent limitation of directly training a diffusion model on an offline dataset. Such a model typically yields an offline data distribution $p_{\mathcal{D}}(s, a, s')$, which does not align perfectly with online data and causes distribution shift issues. To address this, our method needs to guide the diffusion sampling process towards the online distribution. This is achieved by decomposing the online data distribution into the following form:

$$q_\pi(s, a, s') \propto p_\theta(s, a, s') e^{-\mathcal{E}(s,a,s')}, \tag{3}$$

where $p_\theta(s, a, s')$ is the distribution generated by the denoiser network, parameterized by θ. $\mathcal{E}(s, a, s')$ is the energy function, which serves as the guidance to bridge the gap between generated distribution and online data distribution. The following theorem shows such an energy function exists.

Theorem 3.1. *Let $p_\theta(s)$ be the marginal distribution of $p_\theta(s, a, s')$, $p_\theta(a|s)$ and $p_\theta(s'|s, a)$ be the conditional distribution of $p_\theta(s, a, s')$ given s and (s, a). Eq. (3) is valid if the energy function $\mathcal{E}(s, a, s')$ is structured as follows:*

$$\mathcal{E}(s, a, s') = \mathcal{E}_1(s) + \mathcal{E}_2(a) + \mathcal{E}_3(s'), \tag{4}$$

such that $e^{\mathcal{E}_1(s)} \propto \frac{p_\theta(s)}{d^\pi(s)}$, $e^{\mathcal{E}_2(a)} \propto \frac{p_\theta(a|s)}{\pi(a|s)}$, $e^{\mathcal{E}_3(s')} \propto \frac{p_\theta(s'|s,a)}{T(s'|s,a)}$.

This theorem indicates that the energy function can be decomposed into three distinct parts. Each part is responsible for aligning the generated distribution with different aspects of the online data: the online state distribution, the current policy action distribution, and the environmental dynamics.

3.2. Learning Energy Guidance by Contrastive Energy Prediction

First, we concentrate on the first component, $\mathcal{E}_1(s)$. We assume that the energy is estimated using a neural network denoted as $\mathcal{E}_{\phi_1}(s)$. Let K and K_{neg} be two positive numbers. Given $s_1, s_2, \ldots, s_K$, K i.i.d. samples drawn from the distribution $p_\theta(s)$, and $s_i^1, s_i^2, \ldots, s_i^{K_{\text{neg}}}$, K_{Neg} negative samples for s_i. We employ the Information Noise Contrastive Estimation (InfoNCE) loss (van den Oord et al., 2018):

$$\mathcal{L}(\phi_1) = -\sum_{i=1}^{K} \log \frac{e^{-\mathcal{E}_{\phi_1}(s_i)}}{e^{-\mathcal{E}_{\phi_1}(s_i)} + \sum_{j=1}^{K_{\text{neg}}} e^{-\mathcal{E}_{\phi_1}(s_i^j)}}, \tag{5}$$

Then, we devise positive and negative samples to achieve the target energy function established by Thm. 3.1. Suppose the distribution of positive samples is $\mu(s)$, the distribution of negative samples is $\nu(s)$, the final optimized results is $e^{\mathcal{E}_{\phi_1}(s)} \propto \frac{\nu(s)}{\mu(s)}$ (van den Oord et al., 2018). Compared to the function indicated by Thm. 3.1, the result can be achieved by selecting $\mu(s) = d^\pi(s)$, $\nu(s) = p_\theta(s)$. Following the approach of Sinha et al. (2022); Liu et al. (2021), we construct a *positive buffer*, containing only a small set of trajectories from very recent policies. The data distribution in this buffer can be viewed as an approximation of the on-policy distribution $d^\pi(s)$. While $p_\theta(s)$ is the distribution of the data generated during the denoising steps. Therefore, the positive samples is sampled from the *positive buffer* and the negative samples is sampled from the denoiser.

For the remaining terms $\mathcal{E}_2(a)$ and $\mathcal{E}_3(s')$, we also parameterize them as $\mathcal{E}_{\phi_2}(a)$ and $\mathcal{E}_{\phi_3}(s')$, and employ infoNCE loss to estimate them. Similarly, the loss computation requires the generation of positive and negative samples. Differently from $\mathcal{E}_{\phi_1}(s)$, here the samples from the offline buffer can also used for positive-sample generation. For $\mathcal{E}_{\phi_2}(a)$, the positive samples are obtained by first sampling s from the offline and online distribution and outputting the action of the current policy on s. For $\mathcal{E}_{\phi_3}(s')$, positive samples are obtained by sampling (s, a, s') from the from the offline and online distribution and outputting s' directly. Negative samples for both $\mathcal{E}_{\phi_2}(a)$ and $\mathcal{E}_{\phi_3}(s')$ are generated using the denoiser. For each positive samples of $\mathcal{E}_{\phi_2}(a)$, the negative samples are generated conditioned on the given states. While for each positive samples of $\mathcal{E}_{\phi_3}(s')$, they are generated conditioned on the sampled states-action pairs. This method of conditional sample generation is inspired by techniques used in decision-making scenarios (Ajay et al., 2023; Janner et al., 2022). The loss functions of parameterized $\mathcal{E}_{\phi_2}(a)$ and $\mathcal{E}_{\phi_3}(s')$ are:

$$\mathcal{L}(\phi_2) = -\sum_{i=1}^{K} \log \frac{e^{-\mathcal{E}_{\phi_2}(a_i)}}{e^{-\mathcal{E}_{\phi_2}(a_i)} + \sum_{j=1}^{K_{\text{neg}}} e^{-\mathcal{E}_{\phi_2}(a_i^j)}}, \tag{6}$$

$$\mathcal{L}(\phi_3) = -\sum_{i=1}^{K} \log \frac{e^{-\mathcal{E}_{\phi_3}(s_i')}}{e^{-\mathcal{E}_{\phi_3}(s_i')} + \sum_{j=1}^{K_{\text{neg}}} e^{-\mathcal{E}_{\phi_3}(s_i^{j'})}}. \tag{7}$$

3.3. Sampling under Energy Guidance

To realize this distribution with diffusion models, we need to calculate its score function in the sampling process, taking into account the energy function designed in Sec. 3.2:

$$\begin{aligned}&\nabla_{(s,a,s')} \log q_\pi(s,a,s') \\ &= \nabla_{(s,a,s')} \log p_\theta(s,a,s') - \nabla_{(s,a,s')} \mathcal{E}(s,a,s')\end{aligned} \tag{8}$$

In the denoising process, we need to obtain the score function at each timestep. Denote the forward distribution at time t starting from $p_0(s,a,s')$ as $p_t(s,a,s')$. Remember that the denoiser model $D_\theta(s,a,s';\sigma)$ is designed to match the score with the expression:

$$\nabla \log p_\theta(s,a,s') = (D_\theta(s,a,s';\sigma) - (s,a,s'))/\sigma^2. \tag{9}$$

Thus, we can obtain the gradient through the denoiser model. Then, the key problem is to obtain the intermediate energy guidance at time t, which is addressed in the following theorem.

Theorem 3.2 (Thm. 3.1 in (Lu et al., 2023b))**.** *Denote $q_t(x_t) := \int q_t(x_t|x_0)q_0(x_0)dx_0$ and $p_t(x_t) := \int p_t(x_t|x_0)p_0(x_0)dx_0$ as the marginal distributions at time t, and define*

$$\mathcal{E}_t(x_t) := \begin{cases} \mathcal{E}(x_0), & t = 0, \\ -\log \mathbb{E}_{q_t(x_0|x_t)}[e^{-\mathcal{E}(x_0)}], & t > 0. \end{cases}$$

Then the score functions satisfy

$$\nabla_{x_t} \log q_t(x_t) = \nabla_{x_t} \log p_t(x_t) - \nabla_{x_t} \mathcal{E}_t(x_t).$$

Let x_0 denote the set of original positive samples. According to this theorem, we aim to formulate the energy function at timestep t. Drawing inspiration from Sec. 3.2, we opt for a contrastive learning approach. Within this method, these positive samples for the energy function at t timestep are derived from the original samples subjected to t steps of noise addition, i.e., $x_t \sim p(\boldsymbol{x}_i;\sigma_t)$, where σ_t is chosen as (Karras et al., 2022):

$$\sigma_t = \left(\sigma_{\max}^{\frac{1}{\rho}} + \frac{t}{T-1}(\sigma_{\min}^{\frac{1}{\rho}} - \sigma_{\max}^{\frac{1}{\rho}})\right)^\rho \quad \text{and} \quad \sigma_T = 0,$$

where ρ, $\sigma_{\max}$ and $\sigma_{\min}$ are pre-defined constant numbers. As for the negative samples, they are generated by the denoiser at time t. Then we start the denoising process defined in Eq. (1). The psuedo-code of EDIS is demonstrated in Appx. B.

4. Theoretical Analysis

In this section, we provide theoretical analysis on the suboptimality bound of our method and previous methods. We follow the assumptions of Fitted-Q-iteration (Chen & Jiang, 2019): Let $\mathcal{F}$ be the Q function class that satisfies realizability: $Q^* \in \mathcal{F}$. $\mathcal{F}$ is closed under Bellman update: $\forall f \in \mathcal{F}, \mathcal{T}^\pi f \in \mathcal{F}$, where $\mathcal{T}^\pi f(s,a) := r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V_f^\pi(s')]$, $V_f^\pi(s') := \mathbb{E}_{a' \sim \pi} f(s',a')$ and π can be any policy. Additionally, we assume $\mathcal{F}$ is L-Lipschitz.

In the offline phase, we initialize a value function Q_0. Based on Q_0, we perform online learning by collecting n samples each iteration. The naive method, which relies solely on the information from n collected samples to update the policy, is characterized by the following suboptimality bound.

Theorem 4.1. *Suppose the learned Q function after the offline phase satisfies $(\|Q^* - Q_0\|_\infty + \|Q^\pi - Q_0\|_\infty) \le \epsilon_Q$, where Q^* is the Q function of the optimal policy π^*. With probability at leaset $1-\delta$, the output policy π after one iteration of online phase satisfies*

$$J(\pi^*) - J(\pi) \le \frac{2\gamma\epsilon_Q}{1-\gamma} + \frac{\sqrt{C_{\pi,\pi^*}} + 1}{1+\gamma} \frac{1}{(1-\gamma)^{2.5}} \sqrt{\frac{56 R_{\max}^2 \log \frac{|\mathcal{F}|^2}{\delta}}{3n}},$$

where $C_{\pi,\pi^*} = \max_{s,a} \frac{\pi_1(a|s)}{\pi_2(a|s)}$, n is the number of collected samples in one iteration.

Considering the performance bound after a single iteration is reasonable, as subsequent iterations can be analyzed by treating the output value function as the new initialization. The suboptimality after one iteration is characterized by the initialization error, augmented by a term of $O\left(\sqrt{\frac{1}{n}}\right)$. Notably, this error is nonnegligible because of the relatively small size of the online interaction samples.

A natural approach for leveraging offline data is augmenting the online dataset with the available offline data. To establish the suboptimality bound for such a method, we need to introduce the concept of concentratability coefficient.

Definition 4.2 (Concentratability Coefficient)**.** The concentratbility coefficient C between state distribution ν and μ is defined as $\forall s \in \mathcal{S}, \frac{\nu(s)}{\mu(s)} \le C$.

Assuming the concentratability coefficient between the state distribution of current policy d^π and offline dataset D is C_d, we have the following theorem:

Theorem 4.3. *Under the conditions of Thm. 4.1, by replaying offline data in the online iteration, the output policy π after one iteration satisfies*

$$J(\pi^*) - J(\pi) \le \frac{2\gamma\epsilon_Q}{1-\gamma} + \frac{\sqrt{\widetilde{C_d} C_{\pi,\pi^*}} + 1}{1+\gamma} \frac{1}{(1-\gamma)^{2.5}} \sqrt{\frac{56 R_{\max}^2 \log \frac{|\mathcal{F}|^2}{\delta}}{3(n+N)}},$$

where $\widetilde{C_d} = \frac{(n+N)C_d}{N+nC_d}$, N is the number of samples in offline dataset.

The error bound is characterized by an order of $O\left(\sqrt{\frac{1}{n+N}}\right)$. Given the substantial difference in size, where N is notably larger than n, replaying offline data shows advantages to the previous method. This result explains the improved performance resulting from online dataset augmentation. However, the introduction of the concentration coefficient C_d in the new method reflects the consequences of the distribution shift issue. This coefficient can assume a significantly large value, particularly if the offline data fails to adequately cover a substantial portion of the state space. It detriments the potential performance improvement attainable through enhanced utilization of prior knowledge.

Based on the diffusion model generator, our new method has access to synthetic samples $(\tilde{s}, \tilde{a}, \tilde{r}, \tilde{s}')$ during the online interaction phase. Let the distribution of $\tilde{s}$ be denoted as $\tilde{d}^\pi$, and the conditional distribution of $\tilde{s}'$ as $\widetilde{T}(\cdot|\tilde{s}, \tilde{a})$. The state distribution error and model error of the diffusion model are defined as:$\epsilon_d = \left\| d^\pi - \tilde{d}^\pi \right\|$, and $\epsilon_m^d = \left\| T(\cdot|s,a) - \widetilde{T}(\cdot|s,a) \right\|_{\tilde{d}^\pi}$.

Theorem 4.4. *Under the conditions of Thm. 4.1, we assume the data generator generates data with state distribution error ϵ_d and model error ϵ_m, then the output policy π after one iteration of online phase satisfies*

$$J(\pi^*) - J(\pi) \leq \frac{2\gamma}{1-\gamma}\epsilon_Q + \left(\sqrt{C_{\pi,\pi^*}} + 1\right)\left(\frac{1}{(1-\gamma)^2}R_{\max}\epsilon_d + \frac{\gamma L}{(1-\gamma)^2}\epsilon_m^d\right),$$

where L is Lipschitz constant of $\mathcal{F}$.

Here, we simplify the analysis by disregarding the generation errors associated with $\tilde{a}$ and $\tilde{r}$. This simplification is justified as modeling these errors is considerably more straightforward compared to modeling the error associated with s and $\tilde{s}$. Additionally, considering these two errors leads to a similar analysis and does not alter the order of the error term. According to the theorem, the error bound is not directly related to the number of samples, but it is highly dependent on the error of the diffusion model. Based on the theory of supervised learning (von Luxburg & Schölkopf, 2011), the model error ϵ_m is $O\left(\sqrt{\frac{1}{n+N}}\right)$. The state distribution error ϵ_d is the distribution matching error, which is also $O\left(\sqrt{\frac{1}{n+N}}\right)$ (Xu et al., 2021). Therefore, our method exhibits the same convergence rate as the previous method without introducing the distribution shift issue identified by the concentration coefficient C_d.

Model-based method can also generate synthetic online samples. Assume the model error of the model is $\epsilon_m^t = \left\| T(\cdot|s,a) - \widetilde{T}(\cdot|s,a) \right\|_{d^\pi}$.

Theorem 4.5. *Under the conditions of Thm. 4.1, we assume the model has an error ϵ_m^t, then the output policy π after one iteration of online phase satisfies*

$$J(\pi^*) - J(\pi) \leq \frac{2\gamma}{1-\gamma}\epsilon_Q + \left(\sqrt{C_{\pi,\pi^*}} + 1\right)\left(\frac{R_{\max}}{(1-\gamma)^3}\epsilon_m^t + \frac{\gamma L}{(1-\gamma)^2}\epsilon_m^t\right).$$

The error of model-based approach is irrelevant to the C_d term, illustrating that incorporating prior knowledge about transitions from the offline dataset can address the issue of distribution shift. Nevertheless, this method is subject to a compounding error, with an error bound expressed as $O\left(\frac{1}{(1-\gamma)^3}\right)$. This limitation implies that model-based methods may not surpass the performance of model-free methods (Rafailov et al., 2023). In contrast, the error bound for EDIS is lower, at $O\left(\frac{1}{(1-\gamma)^2}\right)$, which effectively reduces the impact of compounding error. In summary, EDIS overcomes all the issues previous methods have in theory.

5. Experiments

In this section, we empirically validate the effectiveness of EDIS. Sec. 5.1 showcases the considerable performance enhancement achieved by EDIS when integrated with off-the-shelf offline-to-online algorithms. Sec. 5.2 investigates the reasons for this improvement, highlighting two key factors: i) the robust distribution modeling capability of diffusion models, and ii) the focus on modeling data distribution instead of the transition function, which effectively mitigates compounding errors. Sec. 5.3 includes an ablation study on three energy functions to underscore their critical role in our algorithm. In this section, every experiment result is averaged over five random seeds.

5.1. Enhanced Performance Achieved by EDIS

We evaluate the performance of EDIS on three benchmark tasks from D4RL (Fu et al., 2020): MuJoCo Locomotion, AntMaze Navigation, and Adroit Manipulation. We implement EDIS on top of base algorithms Cal-QL (Nakamoto et al., 2023), a state-of-the-art offline-to-online method that effectively calibrates over-conservatism of CQL (Kumar et al., 2020), and IQL (Kostrikov et al., 2022), which adopts AWR-style policy learning in both phases. The 0.2M fine-tuning results after 1M pre-training are presented in Tab. 1.

Notably, the integration of EDIS with Cal-QL yields a substantial 26.3% improvement in overall performance, demonstrating significant enhancements in the Adroit domain and MuJoCo random datasets. The latter are characterized by

Table 1. Enhanced performance achieved by EDIS after 0.2M online fine-tuning on base algorithms Cal-QL and IQL. Each result is the average score over five random seeds ± standard deviation.

Dataset	Cal-QL		IQL		Avg.	
	Base	Ours	Base	Ours	Base	Ours
hopper-random-v2	17.6±3.1	**98.1±12.3**	10.0±1.7	**12.1±4.0**	13.8±2.4	**55.1±8.2**
hopper-medium-replay-v2	102.2±4.6	**109.9±0.8**	99.0±4.7	**101.1±1.6**	100.6±4.7	**105.5±1.2**
hopper-medium-v2	97.6±1.4	**105.0±4.1**	59.2±7.9	**73.1±4.4**	78.4±4.7	**89.1±4.3**
hopper-medium-expert-v2	107.9±9.6	**109.7±1.4**	90.0±22.1	**105.7±6.7**	99.0±15.9	**107.7±4.1**
halfcheetah-random-v2	74.8±3.2	**86.3±1.8**	36.7±3.0	**38.9±1.9**	55.8±3.1	**62.6±1.9**
halfcheetah-medium-replay-v2	76.6±1.2	**86.7±1.4**	45.6±0.4	**47.1±0.3**	61.1±0.8	**66.9±0.9**
halfcheetah-medium-v2	72.3±2.1	**83.9±1.0**	48.6±0.2	**49.8±0.2**	60.5±1.2	**66.9±0.6**
halfcheetah-medium-expert-v2	91.0±0.6	**98.6±0.5**	**87.9±4.3**	85.4±2.7	89.5±2.5	**92.0±1.6**
walker2d-random-v2	15.1±3.5	**61.6±12.6**	6.5±0.7	**16.2±2.9**	10.8±2.1	**38.9±7.8**
walker2d-medium-replay-v2	87.3±8.5	**112.9±6.4**	83.6±2.0	**95.3±1.4**	85.5±5.3	**104.1±3.9**
walker2d-medium-v2	84.2±0.3	**103.5±1.8**	83.6±2.1	**85.2±1.3**	83.9±1.2	**94.4±1.6**
walker2d-medium-expert-v2	111.1±0.6	**118.5±4.0**	**108.9±2.9**	107.5±4.5	110.0±1.8	**113.0±4.3**
locomotion total	937.7	**1174.7**	759.6	**856.3**	848.7	**1015.5**
antmaze-umaze-v2	96.3±1.4	**98.9±1.3**	79.2±4.1	**81.1±3.4**	87.8±2.8	**90.0±2.4**
antmaze-umaze-diverse-v2	93.4±4.6	**95.9±2.8**	51.3±4.5	**66.7±5.0**	72.4±4.6	**81.3±3.9**
antmaze-medium-diverse-v2	81.4±3.9	**89.3±4.8**	75.6±1.9	**81.8±4.8**	78.5±2.9	**85.6±4.8**
antmaze-medium-play-v2	86.8±1.6	**93.9±2.7**	81.0±2.2	**86.2±1.3**	83.9±1.9	**90.1±2.0**
antmaze-large-play-v2	42.5±5.2	**66.1±8.2**	39.2±7.2	**40.0±5.3**	40.9±6.2	**53.1±6.8**
antmaze-large-diverse-v2	42.3±2.2	**57.1±2.8**	45.0±8.7	**52.1±2.6**	43.7±5.5	**54.6±2.7**
antmaze-umaze-diverse-v2	93.4±4.6	**95.9±2.8**	51.3±4.5	**66.7±5.0**	72.4±4.6	**81.3 ± 3.9**
antmaze-large-diverse-v2	42.3±2.2	**57.1± 2.8**	45.0±8.7	**52.1±2.6**	63.7± 5.5	**54.6±2.7**
antmaze total	442.7	**501.2**	371.3	**407.9**	407.0	**454.6**
relocate-human-v1	-0.4±0.2	**0.2±0.2**	1.4±0.3	**1.7±0.4**	0.5±0.3	**1.0±0.3**
pen-human-v1	68.4±8.7	**95.6±6.2**	91.2±5.2	**103.6±4.2**	79.8±7.0	**99.6±5.2**
door-human-v1	0.1±0.2	**58.4±17.6**	20.7±4.6	**25.6±3.3**	10.4±2.4	**42.0±10.5**
adroit total	68.1	**154.2**	113.3	**130.9**	90.7	**142.3**
total	1448.5	**1830.1**	1244.2	**1395.1**	1346.4	**1612.4**

a scarcity of successful demonstrations, creating a substantial gap between offline and online distributions that poses a challenge to previous methods. EDIS addresses this issue by directly generating online samples, overcoming the distribution shift issue. Even combined with IQL, where the conservatism cannot be entirely dropped during online fine-tuning due to its reliance on in-sample data, EDIS still achieves an average performance improvement of 12.1%. For more information on implementation details, please refer to Appx. C.

5.2. Comparisons between EDIS and Basic Model-based Methods

While EDIS demonstrates superior performance on the D4RL benchmarks, a natural question arises: what contributes to this improvement? We hypothesize two main contributors to EDIS's success: firstly, the superior representational capabilities of diffusion models, and secondly, the innovative modeling approach that focuses on the distribution of (s, a, s') tuples instead of modeling the transition function as is common in traditional model-based methods. To substantiate our hypothesis, we have undertaken further experiments involving two models tasked with representing the transition function of the environments. The models have equivalent capacity as EDIS (i.e., having the same number of parameters), allowing for a direct comparison. These models are then used to rollout current policies within the learned model, thereby augmenting online data. One model employs a multi-layer perceptron (MLP) structure, while the other utilizes a diffusion model. The base algorithm is Cal-QL and the results are shown in Fig. 1. It can be seen that the MLP model does not exhibit enhanced performance compared to the original algorithm. According to the theory outlined in Sec. 4, this suggests that the performance

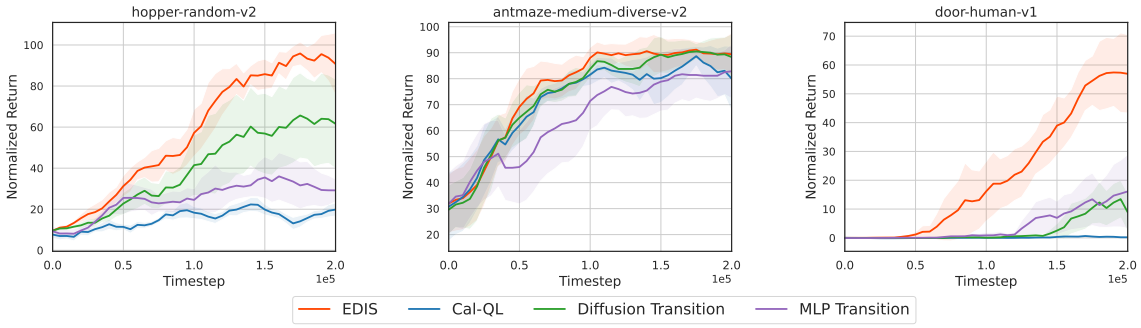

Figure 1. Comparison of EDIS and traditional model-based method. Diffusion transition and MLP transition mean the transition function is modeled by diffusion model and MLP respectively.

drop induced by distribution shift and MLP model error is similar.

In contrast, the diffusion transition model shows some improvement, indicating that diffusion model suffers smaller model error than MLP model because of its superior distribution modeling ability. Despite this, simply employing a powerful diffusion model in conjunction with traditional data augmentation methods could not match the performance achieved by EDIS. This is attributed to the compounding error introduced by rolling out policies in a misspecified transition model. The unique data generation approach of EDIS, which directly generates online samples, proves crucial in achieving its remarkable performance. To visualize the effect of compounding error reduction, we compare the state distribution generated by these models. The result is defered to Appx. C.2 because of space limitation. In summary, the performance enhancement observed with EDIS stems not only from the capability of the diffusion model but also from its distinctive data generation methodology.

5.3. Ablation for Energy Functions in EDIS

The energy functions, as described in Sec. 3.1, align each component in the generated distribution with their online counterparts, including the state distribution, the action distribution under the current policy, and the transition fidelity. We conduct ablation studies on each energy function by omitting its guidance during the reverse-time *SDE* in hopper-random-v2, antmaze-medium-diverse-v2, and door-human-v1. We compare the divergence between the distribution generated by diffusion models and the real environments. For the state divergence, we calculate the JS divergence between the two distributions. The details of JS divergence and its calculation state distribution are shown in Appx. C.4. For action divergence and state divergence, we calculate the mean square loss of the action or next state generated by the diffusion model and the current policy or the real dynamics. As shown in Tab. 2, the guidance of the energy function effectively decreases the divergence from the online sample distribution, real-world transitions, and actions induced by the policy interacting with the environment.

Furthermore, the results in Fig. 2 demonstrate that a larger divergence of the three parts detriments the final performance. Particularly, in environments like hopper-random-v2, EDIS struggles to excel without the state energy function, demonstrating the ineffectiveness of merging unrelated offline data directly into the online fine-tuning phase.

6. Related Work

In this section, we provide a brief summary of related work, the complete version is in Appx. D.

Offline-to-online Reinforcement Learning. Offline-to-online reinforcement learning methods are developed to mitigate the dichotomy between the costly exploration in online RL and the typically suboptimal performance of offline RL. The learning process is typically divided into two phases: Warming-up the policy and value functions in the offline phase and using them as initialization in the online phase (Nair et al., 2020; Wang et al., 2023a; Nakamoto et al., 2023; Lee et al., 2021). These approaches often employ offline RL methods based on policy constraints or pessimism in the offline phase (Fujimoto et al., 2019; Fujimoto & Gu, 2021; Kumar et al., 2020). However, the conservatism conflicts with the online phase and may induce performance degradation. Various strategies have been implemented to tackle this issue, such as policy expansion (Zhang et al., 2023), value function calibration (Nakamoto et al., 2023), Q-ensemble techniques (Lee et al., 2021), and constraint methods (Nair et al., 2020; Kostrikov et al., 2022; Li et al., 2023). Despite the advancements, there has been less focus on the crucial aspect of integrating useful data during the fine-tuning phase to enhance training efficiency. Standard practices include enriching the replay buffer with offline data (Nakamoto et al., 2023; Zhang et al., 2023), adopting balanced sampling methods for managing both online and offline data sources (Lee et al., 2021; Ball et al., 2023), or

Table 2. Divergence comparisons for energy function ablation study (lower is better). Each result is the average score over five random seeds ± standard deviation.

Dataset	State Divergence		Action Divergence		Transition Divergence	
	w/o energy	w/ energy	w/o energy	w/ energy	w/o energy	w/ energy
hopper-radnom-v2	0.85±0.02	**0.73±0.03**	0.51±0.04	**0.39±0.02**	0.69±0.03	**0.66±0.04**
antmaze-medium-diverse-v2	0.98±0.00	**0.91±0.02**	0.38±0.08	**0.27±0.08**	0.75±0.14	**0.64±0.14**
door-human-v1	0.40±0.04	**0.24±0.02**	0.54±0.07	**0.41±0.08**	2.57±0.06	**2.52±0.05**

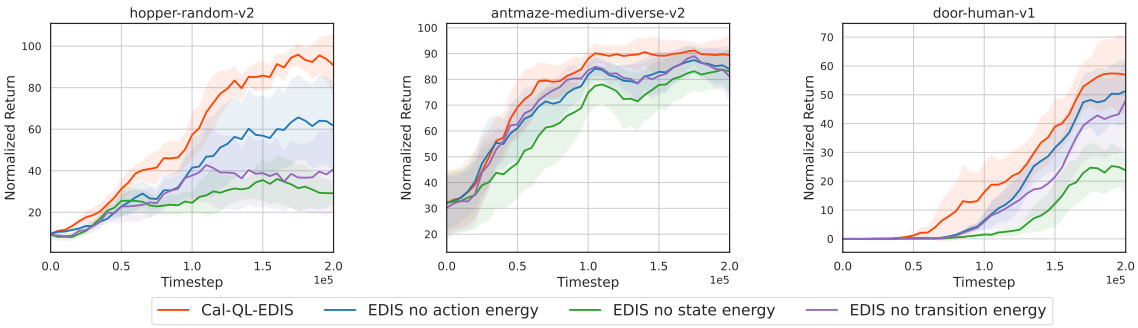

Figure 2. Energy Module Ablation Study of EDIS

building models to performance branch rollout (Rafailov et al., 2023). However, directly replaying the offline data causes a distribution shift, and adopting balanced sampling methods introduces large variance (Nachum et al., 2019) while rolling in the built model suffers from compounding error (Janner et al., 2019). In contrast, our work breaks new ground by proposing a diffusion-based generator specifically designed to generate useful samples by thoroughly leveraging prior knowledge in offline data.

Diffusion Model in Reinforcement Learning. Diffusion models have demonstrated exceptional capabilities in modeling distribution (Saharia et al., 2022; Nichol et al., 2022; Nichol & Dhariwal, 2021). Within the reinforcement learning research, Diffuser (Janner et al., 2022) uses a diffusion model as a trajectory generator and learns a separate return model to guide the reverse diffusion process toward samples of high-return trajectories. Decision Diffuser (Ajay et al., 2023) introduces conditional diffusion with reward or constraint guidance for decision-making tasks, further boosting Diffuser's performance. Expanding the application of diffusion models, SYNTHER (Lu et al., 2023a) focuses on leveraging the diffusion model for upsampling data in both online and offline reinforcement learning scenarios. Recently, several concurrent work also investigates generating samples with certain data distribution by diffusion models. PolyGRAD (Rigter et al., 2023) and PGD (Jackson et al., 2024) embed the policy for classifier-guided trajectory generation, aiming at on-policy world modeling. However, they still use the diffusion model to model the transition function rather than the data distribution directly, which does not eliminate the issue of compounding error. Finally, in the context of offline-to-online reinforcement learning, our research pioneers the utilization of diffusion-based models to actively generate valuable samples. This departure from the passive reuse of offline data marks EDIS as a novel approach, emphasizing the active role of diffusion models in sample generation for offline-to-online RL.

7. Conclusion

This study underscores the crucial role of harnessing offline data effectively to augment online fine-tuning. We introduce Energy-guided Diffusion Sampling (EDIS), a powerful approach leveraging a state-of-the-art generative model with energy guidance. In contrast to existing strategies that primarily rely on initializing with offline-derived policies and Q functions or replaying offline data, EDIS takes a proactive stance by generating synthetic data within the online data distribution, drawing information from all available data sources. Theoretical analysis verifies that EDIS outperforms prior methods in addressing distribution shift challenges and effectively mitigates compounding errors. As a versatile solution, EDIS seamlessly integrates with prevalent offline-to-online frameworks. When combined with methods like Cal-QL and IQL, EDIS showcases substantial performance enhancements. This superior generative modeling capabilities of EDIS mark it as a scalable approach, providing promising potential for designing data-efficient learning strategies in more complex environments with high-dimensional state or action spaces.

Acknowledgement

This work is supported by the National Science Foundation of China (61921006). The authors thank anonymous reviewers for their helpful discussions and suggestions for improving the article.

Impact Statement

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.

References

Ajay, A., Kumar, A., Agrawal, P., Levine, S., and Nachum, O. OPAL: offline primitive discovery for accelerating offline reinforcement learning. In *Proceedings of the 9th International Conference on Learning Representations (ICLR'21)*, Virtual Event, 2021.

Ajay, A., Du, Y., Gupta, A., Tenenbaum, J. B., Jaakkola, T. S., and Agrawal, P. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations (ICLR'23)*, Kigali, Rwanda, 2023.

Ball, P. J., Smith, L. M., Kostrikov, I., and Levine, S. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning (ICML'23)*, Honolulu, USA, 2023.

Chen, H., Lu, C., Ying, C., Su, H., and Zhu, J. Offline reinforcement learning via high-fidelity generative behavior modeling. In *The 11th International Conference on Learning Representations (ICLR'23)*, Kigali, Rwanda, 2023a.

Chen, J. and Jiang, N. Information-theoretic considerations in batch reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, Long Beach, USA, 2019.

Chen, R., Liu, X., Liu, T., Jiang, S., Xu, F., and Yu, Y. Foresight distribution adjustment for off-policy reinforcement learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS'24)*, pp. 317–325, Auckland, New Zealand, 2024.

Chen, X., Yu, Y., Zhu, Z., Yu, Z., Chen, Z., Wang, C., Wu, Y., Qin, R., Wu, H., Ding, R., and Huang, F. Adversarial counterfactual environment model learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS'23)*, New Orleans, LA, 2023b.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS'21)*, Virtual Event, 2021.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, Long Beach, USA, 2019.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NeurIPS'24)*, pp. 2672–2680, Montreal, Canada, 2014.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Proceedings of the 3rd Annual Conference on Robot Learning (CoRL'19)*, Osaka, Japan, 2019.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, Stockholmsmässan, Sweden, 2018.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al. Deep q-learning from demonstrations. In *Proceedings of the 32nd AAAI conference on artificial intelligence (AAAI'18)*, New Orleans, USA, 2018.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Proceedings of the 33rd Neural Information Processing Systems (NeurIPS'20)*, Virtual Event, 2020.

Hubbs, C. D., Perez, H. D., Sarwar, O., Sahinidis, N. V., Grossmann, I. E., and Wassick, J. M. Or-gym: A reinforcement learning library for operations research problems. *CoRR*, abs/2008.06319, 2020.

Jackson, M. T., Matthews, M. T., Lu, C., Ellis, B., Whiteson, S., and Foerster, J. N. Policy-guided diffusion. *CoRR*, abs/2404.06356, 2024.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. In *Proceedings of the 32nd Neural Information Processing Systems (NeurIPS'19)*, Vancouver, Canada, 2019.

Janner, M., Du, Y., Tenenbaum, J., and Levine, S. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, Baltimore, USA, 2022.

Jia, C., Zhang, F., Li, Y., Gao, C., Liu, X., Yuan, L., Zhang, Z., and Yu, Y. Disentangling policy from offline task representation learning via adversarial data augmentation. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS'24)*, pp. 944–953, Auckland, New Zealand, 2024.

Jin, X., Liu, X., Jiang, S., and Yu, Y. Hybrid value estimation for off-policy evaluation and offline reinforcement learning. *CoRR*, abs/2206.02000, 2022.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Proceedings of the 36th Neural Information Processing Systems (NeurIPS'22)*, LA, USA, 2022.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *Proceedings of the 10-th International Conference on Learning Representations (ICLR'22)*, Virtual Event, 2022.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, Virtual Event, 2020.

Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning (CoRL'21)*, London, UK, 2021.

Li, J., Hu, X., Xu, H., Liu, J., Zhan, X., and Zhang, Y. PROTO: iterative policy regularized offline-to-online reinforcement learning. *CoRR*, abs/2305.15669, 2023.

Ling, H., Wang, Z., and Wang, J. Learning to stop cut generation for efficient mixed-integer linear programming. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'24)*, 38(18):20759–20767, Mar. 2024.

Liu, X.-H., Xue, Z., Pang, J., Jiang, S., Xu, F., and Yu, Y. Regret minimization experience replay in off-policy reinforcement learning. In *Proceedings of 34th conference on Neural Information Processing Systems (NeurIPS'21)*, Virtual Event, 2021.

Lu, C., Ball, P. J., and Parker-Holder, J. Synthetic experience replay. *CoRR*, abs/2303.06614, 2023a.

Lu, C., Chen, H., Chen, J., Su, H., Li, C., and Zhu, J. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning(ICML'23)*, volume 202, pp. 22825–22855, Honolulu, HI, 2023b.

Mazoure, B., Talbott, W., Bautista, M. Á., Hjelm, R. D., Toshev, A., and Susskind, J. M. Value function estimation using conditional diffusion models for control. *CoRR*, abs/2306.07290, 2023.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Nachum, O., Chow, Y., Dai, B., and Li, L. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Proceedings of the 32nd Neural Information Processing Systems (NeurIPS'19)*, pp. 2315–2325, Vancouver, Canada, 2019.

Nair, A., Dalal, M., Gupta, A., and Levine, S. Accelerating online reinforcement learning with offline datasets. *CoRR*, abs/2006.09359, 2020.

Nakamoto, M., Zhai, Y., Singh, A., Mark, M. S., Ma, Y., Finn, C., Kumar, A., and Levine, S. Cal-ql: Calibrated offline RL pre-training for efficient online fine-tuning. *CoRR*, abs/2303.05479, 2023.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, Virtual Event, 2021.

Nichol, A. Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, Baltimore, USA, 2022.

Peng, X. B., Coumans, E., Zhang, T., Lee, T. E., Tan, J., and Levine, S. Learning agile robotic locomotion skills by imitating animals. In *Proceedings of the 14th Robotics: Science and Systems (RSS'20)*, Virtual Event, 2020.

Rafailov, R., Hatch, K. B., Kolev, V., Martin, J. D., Phielipp, M., and Finn, C. Moto: Offline pre-training to online fine-tuning for model-based robot learning. In *Conference on Robot Learning (CoRL'23)*, pp. 3654–3671, 2023.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of the 12th Robotics: Science and Systems (RSS'18)*, PA, USA, 2018.

Rigter, M., Yamada, J., and Posner, I. World models via policy-guided trajectory diffusion. *CoRR*, abs/2312.08533, 2023.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, S. K. S., Lopes, R. G., Ayan, B. K., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding. In *Proceedings of the 35th Neural Information Processing Systems (NeurIPS'22)*, New Orleans, USA, 2022.

Schaal, S. Learning from demonstration. *Advances in neural information processing systems*, 9, 1996.

Sinha, S., Song, J., Garg, A., and Ermon, S. Experience replay with likelihood-free importance weights. In *Learning for Dynamics and Control Conference (L4RC'22)*, Stanford, USA, 2022.

Song, Y., Zhou, Y., Sekhari, A., Bagnell, J. A., Krishnamurthy, A., and Sun, W. Hybrid rl: Using both offline and online data can make rl efficient. In *The 11th International Conference on Learning Representations (ICLR'22)*, Virtual Event, 2022.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tarasov, D., Nikulin, A., Akimov, D., Kurenkov, V., and Kolesnikov, S. CORL: Research-oriented deep offline reinforcement learning library. In *3rd Offline RL Workshop: Offline RL as a "Launchpad"*, 2022.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Vecerik, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., and Riedmiller, M. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *CoRR*, abs/1707.08817, 2017.

von Luxburg, U. and Schölkopf, B. Statistical learning theory: Models, concepts, and results. In *Inductive Logic*, volume 10, pp. 651–706. Elsevier, 2011.

Wang, J., Wang, Z., Li, X., Kuang, Y., Shi, Z., Zhu, F., Yuan, M., Zeng, J., Zhang, Y., and Wu, F. Learning to cut via hierarchical sequence/set model for efficient mixed-integer programming. *CoRR*, abs/2008.06319, 2024.

Wang, S., Yang, Q., Gao, J., Lin, M. G., Chen, H., Wu, L., Jia, N., Song, S., and Huang, G. Train once, get a family: State-adaptive balances for offline-to-online reinforcement learning. *CoRR*, abs/2310.17966, 2023a.

Wang, X., Chen, Y., Yang, J., Wu, L., Wu, Z., and Xie, X. A reinforcement learning framework for explainable recommendation. In *Proceedings of the 18th International Conference on Data Mining (ICDM'18)*, Singapore, 2018.

Wang, Z., Wang, J., Zhou, Q., Li, B., and Li, H. Sample-efficient reinforcement learning via conservative model-based actor-critic. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'22)*, 36(8):8612–8620, Jun. 2022.

Wang, Z., Hunt, J. J., and Zhou, M. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations (ICLR'23)*, Kigali, Rwanda, 2023b.

Wang, Z., Li, X., Wang, J., Kuang, Y., Yuan, M., Zeng, J., Zhang, Y., and Wu, F. Learning cut selection for mixed-integer linear programming via hierarchical sequence model. In *The Eleventh International Conference on Learning Representations (ICLR'23)*, 2023c.

Xu, T., Li, Z., and Yu, Y. Nearly minimax optimal adversarial imitation learning with known and unknown transitions. *CoRR abs/2106.10424*, 2021.

Zhang, H., Xu, W., and Yu, H. Policy expansion for bridging offline-to-online reinforcement learning. In *The Eleventh International Conference on Learning Representations(ICLR'23)*, Kigali, Rwanda, 2023.

Zhao, X., Zhang, L., Ding, Z., Xia, L., Tang, J., and Yin, D. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th International Conference on Knowledge Discovery & Data Mining (KDD'18)*, London, UK, 2018.

Zhao, Y., Boney, R., Ilin, A., Kannala, J., and Pajarinen, J. Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning. *CoRR*, abs/2210.13846, 2022.

Zhu, H., Gupta, A., Rajeswaran, A., Levine, S., and Kumar, V. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *Proceedings of the 36th International Conference on Robotics and Automation (ICRA'19)*, Montreal, Canada, 2019.

Zhu, Y., Wang, Z., Merel, J., Rusu, A. A., Erez, T., Cabi, S., Tunyasuvunakool, S., Kramár, J., Hadsell, R., de Freitas, N., and Heess, N. Reinforcement and imitation learning for diverse visuomotor skills. In *Proceedings of the 12th Robotics: Science and Systems (RSS'18)*, PA, USA, 2018.

A. Proofs and additional Theory

A.1. Proof of Thm. 3.1

Because $e^{\mathcal{E}_1(s)} \propto \frac{p_\theta(s)}{d^\pi(s)}$, $e^{\mathcal{E}_2(a)} \propto \frac{p_\theta(a|s)}{\pi(a|s))}$, $e^{\mathcal{E}_3(s')} \propto \frac{p_\theta(s')}{T^\pi(s'|s,a))}$, we have $e^{\mathcal{E}_1(s)} = k_1 \frac{p_\theta(s)}{d^\pi(s)}$, $e^{\mathcal{E}_2(a)} = k_2 \frac{p_\theta(a|s)}{\pi(a|s)}$, $e^{\mathcal{E}_3(s')} = k_3 \frac{p_\theta(s'|s,a)}{T(s'|s,a)}$, where k_1, k_2 and k_3 are arbitrary positive constants.

$$\begin{aligned} & e^{-(\mathcal{E}_1(s)+\mathcal{E}_2(a)+\mathcal{E}_3(s'))} \\ = & k_1 k_2 k_3 \frac{d^\pi(s)\pi(a|s)T(s'|s,a)}{p_\theta(s)p_\theta(a|s)p_\theta(s'|s,a)} \\ = & k_1 k_2 k_3 \frac{q_\pi(s,a,s')}{p_\theta(s,a,s')}, \end{aligned}$$

where the last equation comes from $q_\pi(s,a,s') = d^\pi(s)\pi(a|s)T(s'|s,a)$ and $p_\theta(s,a,s') = p_\theta(s)p_\theta(a|s)p_\theta(s'|s,a)$. Because $k_1 k_2 k_3 > 0$, we conclude $e^{-(\mathcal{E}_1(s)+\mathcal{E}_2(a)+\mathcal{E}_3(s'))} \propto \frac{q_\pi(s,a,s')}{p_\theta(s,a,s')}$, then $p_\theta(s,a,s')e^{-\mathcal{E}(s,a,s')} \propto q_\pi(s,a,s')$, and thus Eq. (3) holds.

A.2. Proof of Thm. 4.1

Before proof this theorem, we introduce some new notations.

$$\mathcal{T}^\pi f(s,a) := r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V_f^\pi(s')],$$

where $V_f^\pi(s') := \mathbb{E}_{a' \sim \pi} f(s',a')$. Recursively define $(\mathcal{T}^\pi)^j = (\mathcal{T}^\pi)(\mathcal{T}^\pi)^{j-1}$.

Similarly, we define

$$\mathcal{T} f(s,a) := r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V_f(s')],$$

where $V_f(s') := \max_{a'} f(s',a')$.

Given a function $f : X \mapsto \mathbb{R}$, define its norm $\|\cdot\|_\mu$ with respect to distribution μ as:

$$\|f\|_\mu = \sqrt{\sum_x f^2(x)\mu(x)}.$$

Suppose $\nu(s)$ is a state state distribution $\pi(a|s)$ is an action distribution based on given state s, then $\nu \times \pi(s,a) = \nu(s)\pi(a|s)$.

Lemma A.1 (Theorem 1 of (Sinha et al., 2022))**.** *The Bellman operator $\mathcal{T}$ is a γ-contraction with respect to the $\|\cdot\|_{d^\pi}$ norm, i.e.,*

$$\|\mathcal{T}^\pi f - \mathcal{T}^\pi f'\|_{d^\pi} \le \gamma \|f - f'\|_{d^\pi}, \quad \forall f, f' \in \mathcal{F}.$$

Lemma A.2. *Let ν be any admissible state distribution, π_1 and π_2 be two policies, and $C_{\pi_1,\pi_2} = \max_{s,a} \frac{\pi_1(a|s)}{\pi_2(a|s)}$, $\|\cdot\|_{\nu\times\pi_1} \le \sqrt{C_{\pi_1,\pi_2}} \|\cdot\|_{\nu\times\pi_2}$.*

Proof. For any function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, we have

$$\begin{aligned} \|f\|_{\nu\times\pi_1} &= \left(\sum_{s,a} |f(s,a)|^2 \nu(s)\pi_1(a|s)\right)^{1/2} \\ &\le \left(\sum_{s,a} |f(s,a)|^2 C_{\pi_1,\pi_2} \nu(s)\pi_2(a|s)\right)^{1/2} \\ &= \sqrt{C_{\pi_1,\pi_2}} \left(\sum_{s,a} |f(s,a)|^2 \nu(s)\pi_2(a|s)\right)^{1/2} \\ &= \sqrt{C_{\pi_1,\pi_2}} \|f\|_{\nu\times\pi_2}. \end{aligned}$$

□

To simplify the notation, we use $\widehat{\mathcal{T}}^\pi$ to denote the Bellman operator under limited data, i.e.,

$$\widehat{\mathcal{T}}^\pi f = \arg\min_g \sum_{i=1}^N (g - r - \gamma V_f^\pi)^2.$$

Lemma A.3. *If $f \in \mathcal{F}$, with probability at least $1-\delta$, we have*

$$\left\|(\widehat{\mathcal{T}}^\pi)^j f - (\mathcal{T}^\pi)^j f\right\|_{d^\pi} \le \frac{1-\gamma^j}{1-\gamma} \frac{56 V_{\max}^2 \log \frac{|\mathcal{F}|^2}{\delta}}{3n}.$$

Proof. The proof is based on Lem. 16 of (Chen & Jiang, 2019).

First, we consider $\left\|\widehat{\mathcal{T}}^\pi f - \mathcal{T}^\pi f\right\|_{d^\pi}$. Define

$$L_{d^\pi}(g,h) = \mathbb{E}_{s,a\sim d^\pi}(g - r - \gamma V_h^\pi)^2$$
$$L_D(g,h) = \mathbb{E}_{s,a\sim D}(g - r - \gamma V_h^\pi)^2$$

Then we have $\mathcal{T} f = \arg\min_g L_{d^\pi}(g,f)$ and $\widehat{\mathcal{T}} f = \arg\min_g L_D(g,f)$.

To simplify the notations, we define

$$X(g,f,g^*) := (g(s,a) - r - \gamma V_f^\pi(s'))^2 - (g^*(s,a) - r - V_f^\pi(s'))^2$$

Next, we bound the variance of $X(g,f,g^*)$,

$$\begin{aligned}
\mathbb{V}\left[X(g,f,g^*)\right] &\le \mathbb{E}\left[\left\|X^2(g,f,g^*)\right\|\right] \\
&= \mathbb{E}\left[\left((g(s,a) - r - \gamma V_f^\pi(s'))^2 - (g^*(s,a) - r - V_f^\pi(s'))^2\right)\right] \\
&= \mathbb{E}\left[(g(s,a) - g^*(s,a))^2 (g(s,a) + g^*(s,a) - 2r - 2\gamma V_f^\pi(s'))^2\right] \\
&\le 4V_{\max}^2 \mathbb{E}\left[(g(s,a) - g^*(s,a))^2\right] \\
&= 4V_{\max}^2 \left\|g - g^*\right\|_{d^\pi}^2
\end{aligned}$$

Note that

$$\begin{aligned}
\|g - g^*\|_{d^\pi}^2 &\overset{(a)}{\le} 2(\|g - r - \gamma V_f^\pi\|_{d^\pi}^2 + \|r + \gamma V_f^\pi - g^*\|_{d^\pi}^2) \\
&= 2\mathbb{E}[X(g,f,g^*)],
\end{aligned}$$

where (a) holds because $(a+b)^2 \le 2(a^2+b^2)$.

Finally, we apply Bernstein's inequality and union bound over all $f \in \mathcal{F}$. With probability at least $1-\delta$, we have

$$\begin{aligned}
&\mathbb{E}[X(g,f,g^*) - \frac{1}{n}\sum_{i=1}^n X_i(g,f,g^*) \\
&\le \sqrt{\frac{2\mathbb{V}\left[X(g,f,g^*)\right] \log \frac{|\mathcal{F}|^2}{\delta}}{n}} + \frac{4V_{\max}^2 \log \frac{|\mathcal{F}|^2}{\delta}}{3n} \\
&\le \sqrt{\frac{16 V_{\max}^2 \mathbb{E}[X(g,f,g^*) \log \frac{|\mathcal{F}|^2}{\delta}}{n}} + \frac{4V_{\max}^2 \log \frac{|\mathcal{F}|^2}{\delta}}{3n}
\end{aligned}$$

Since $\widehat{\mathcal{T}}^\pi f$ minimizes $L_D(\cdot, f)$, it also minimizes $\frac{1}{n} X_i(\cdot, f, g^*)$. This is because the two objectives only differ by a constant $\mathcal{L}_D(\cdot, f)$. Therefore, we have

$$\frac{1}{n}\sum_{i=1}^n X_i(\widehat{\mathcal{T}}^\pi f, f, \mathcal{T}^\pi f) \le \frac{1}{n}\sum_{i=1}^n X_i(\mathcal{T}^\pi f, f, \mathcal{T}^\pi f) = 0.$$

Then

$$\mathbb{E}[X(\widehat{\mathcal{T}}^\pi, f, \mathcal{T}^\pi f) \leq \sqrt{\frac{16V_{\max}^2 \mathbb{E}[X(g,f,g^*)\log\frac{|\mathcal{F}|^2}{\delta}}{n}} + \frac{4V_{\max}^2 \log\frac{|\mathcal{F}|^2}{\delta}}{3n}.$$

Solving for the quadratic formula, we have

$$\mathbb{E}[X(\widehat{\mathcal{T}}^\pi, f, \mathcal{T}^\pi f) \leq \frac{56V_{\max}^2 \log\frac{|\mathcal{F}|^2}{\delta}}{3n}.$$

Noticing that

$$\begin{aligned}\left\|\widehat{\mathcal{T}}^\pi f - \mathcal{T}^\pi f\right\|_{d^\pi}^2 &= \mathcal{L}_{d^\pi}(\widehat{\mathcal{T}}^\pi f, f) + \mathcal{L}_{d^\pi}(\mathcal{T}^\pi f, f) \\ &= \mathbb{E}[X(\widehat{\mathcal{T}}^\pi f, f, \mathcal{T}^\pi f)] \leq \epsilon,\end{aligned} \tag{10}$$

where $\epsilon = \frac{56V_{\max}^2 \log\frac{|\mathcal{F}|^2}{\delta}}{3n}$.

For $\left\|(\widehat{\mathcal{T}}^\pi)^2 f - (\mathcal{T}^\pi)^2 f\right\|_{d^\pi}^2$, note that

$$\begin{aligned}\left\|(\widehat{\mathcal{T}}^\pi)^2 f - (\mathcal{T}^\pi)^2 f\right\|_{d^\pi}^2 &= \left\|\widehat{\mathcal{T}}^\pi(\widehat{\mathcal{T}}^\pi f) - \mathcal{T}^\pi(\mathcal{T}^\pi f)\right\|_{d^\pi}^2 \\ &\leq \left\|\widehat{\mathcal{T}}^\pi(\widehat{\mathcal{T}}^\pi f) - \mathcal{T}^\pi(\widehat{\mathcal{T}}^\pi f)\right\|_{d^\pi}^2 + \left\|\mathcal{T}^\pi(\widehat{\mathcal{T}}^\pi f) - \mathcal{T}^\pi(\mathcal{T}^\pi f)\right\|_{d^\pi}^2 \\ &\overset{(a)}{\leq} \epsilon + \left\|\mathcal{T}^\pi(\widehat{\mathcal{T}}^\pi f) - \mathcal{T}^\pi(\mathcal{T}^\pi f)\right\|_{d^\pi}^2 \\ &\overset{(b)}{\leq} \epsilon + \gamma^2 \left\|\widehat{\mathcal{T}}^\pi f - \mathcal{T}^\pi f\right\|_{d^\pi}^2 \\ &\overset{(c)}{\leq} \epsilon + \gamma^2\epsilon,\end{aligned}$$

where (a) and (c) use Eq. (10), (b) use Lem. A.1.

Recursively, we have

$$\left\|(\widehat{\mathcal{T}}^\pi)^2 f - (\mathcal{T}^\pi)^2 f\right\|_{d^\pi}^2 \leq \sum_{i=0}^{j} \gamma^{2i}\epsilon = \frac{1-\gamma^{2j}}{1-\gamma^2}\frac{56V_{\max}^2 \log\frac{|\mathcal{F}|^2}{\delta}}{3n}.$$

□

Lemma A.4 (Lemma 14 in (Chen & Jiang, 2019)). *Assume $f, f' \in \mathcal{F}$ and define $\pi_{f,f'} := \arg\max_{a\in\mathcal{A}} \max\{f(s,a), f'(s,a)\}$. Then we have $\forall \nu \in \Delta(\mathcal{S}\times\mathcal{A})$,*

$$\|\mathcal{T}f - \mathcal{T}f'\|_\nu \leq \gamma \|f - f'\|_{\nu\times\pi_{f,f'}}.$$

Suppose π is induced by Q_0,

$$\begin{aligned}J(\pi^*) - J(\pi) &= \sum_{t=0}^{\infty} \mathbb{E}_{s\sim d^\pi}\gamma^t[V^*(s) - Q^*(s,\pi)] \\ &\leq \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{s\sim d^\pi}[Q^*(s,\pi^*) - \widetilde{Q}(s,\pi^*) + \widetilde{Q}(s,\pi) - Q^*(s,\pi)] \\ &\leq \sum_{t=0}^{\infty}\left(\sum_{s,a} d^\pi(s)\pi^*(a|s)\left|Q^*(s,a) - \widetilde{Q}(s,a)\right| + \sum_{s,a} d^\pi(s)\pi(a|s)\left|Q^*(s,a) - \widetilde{Q}(s,a)\right|\right) \\ &\leq \sum_{t=0}^{\infty}\gamma^t\left(\left\|Q^* - \widetilde{Q}\right\|_{d^\pi\times\pi^*} + \left\|Q^* - \widetilde{Q}\right\|_{d^\pi}\right)\end{aligned} \tag{11}$$

$$
\begin{aligned}
&\left\|Q^* - \widetilde{Q}\right\|_\nu \\
&= \left\|Q^* - \mathcal{T}Q_0 + \mathcal{T}Q_0 - Q^\pi + Q^\pi - \widetilde{Q}\right\|_\nu \\
&\leq \|Q^* - \mathcal{T}Q_0\|_\nu + \|\mathcal{T}Q_0 - Q^\pi\|_\nu + \left\|Q^\pi - \widetilde{Q}\right\|_\nu \\
&= \|Q^* - \mathcal{T}Q_0\|_\nu + \|Q^\pi - (\mathcal{T}^\pi)^\infty Q_0\|_\nu + \left\|(\mathcal{T}^\pi)^\infty Q_0 - \widetilde{Q}\right\|_\nu + \|\mathcal{T}Q_0 - Q^\pi\|_\nu \\
&= \|\mathcal{T}Q^* - \mathcal{T}Q_0\|_\nu + \|Q^\pi - (\mathcal{T}^\pi)^\infty Q_0\|_\nu + \left\|(\mathcal{T}^\pi)^\infty Q_0 - \widetilde{Q}\right\|_\nu + \|\mathcal{T}Q_0 - Q^\pi\|_\nu \\
&\overset{(a)}{\leq} \gamma \|Q^* - Q_0\|_{\nu \times \pi_{Q^*,Q_0}} + \|Q^\pi - (\mathcal{T}^\pi)^\infty Q_0\|_\nu + \left\|(\mathcal{T}^\pi)^\infty Q_0 - \widetilde{Q}\right\|_\nu + \|\mathcal{T}Q_0 - Q^\pi\|_\nu \\
&= \gamma \|Q^* - Q_0\|_{\nu \times \pi_{Q^*,Q_0}} + \|Q^\pi - (\mathcal{T}^\pi)^\infty Q_0\|_\nu + \left\|(\mathcal{T}^\pi)^\infty Q_0 - (\widehat{\mathcal{T}}^\pi)^\infty Q_0\right\|_\nu + \|\mathcal{T}Q_0 - Q^\pi\|_\nu ,
\end{aligned}
\tag{12}
$$

where (a) uses Lem. A.4. The last equality comes from the fact that $\widetilde{Q} = (\widehat{\mathcal{T}}^\pi)^\infty Q_0$.

When $\nu = d^\pi \times \pi^*$, we can use Lem. A.2 to obtain

$$\|\cdot\|_\nu \leq \sqrt{C_{\pi,\pi^*}} \|\cdot\|_{d^\pi} .$$

When $\nu = d^\pi \times \pi_{Q^*,Q_0}$, note that π_{Q^*,Q_0} is more similar to π than π^*, so

$$\|\cdot\|_{\nu \times \pi_{Q^*,Q_0}} \leq \sqrt{C_{\pi,\pi^*}} \|\cdot\|_{d^\pi} .$$

Therefore, we only need to consider how to bound these terms when $\nu = d^\pi$.

According to Lem. A.1,

$$\|\mathcal{T}Q_0 - Q^\pi\|_{d^\pi} \overset{(b)}{=} \|\mathcal{T}Q_0 - \mathcal{T}^\pi Q^\pi\|_{d^\pi} = \|\mathcal{T}^\pi Q_0 - \mathcal{T}^\pi Q^\pi\|_{d^\pi} \overset{(c)}{\leq} \gamma \|Q^\pi - Q_0\|_{d^\pi},$$

where (b) comes from $\pi(a|s) = \arg\max Q_0(s,a)$, (c) uses Lem. A.1.

Let $j \to \infty$, we can apply Lem. A.3 to term $\left\|(\mathcal{T}^\pi)^\infty Q_0 - (\widehat{\mathcal{T}}^\pi)^\infty Q_0\right\|_{d^\pi}$,

$$\left\|(\mathcal{T}^{\pi_k})^\infty Q_0 - (\widehat{\mathcal{T}}^\pi)^\infty Q_0\right\|_{d^\pi} \leq \epsilon',$$

where $\epsilon' = \sqrt{\frac{1}{1-\gamma^2} \frac{56 V_{\max}^2 \log \frac{|\mathcal{F}|}{\delta}}{3n}}$ and the inequality holds with probability at least $1-\delta$.

For the term $\|Q^\pi - (\mathcal{T}^\pi)^\infty Q_0\|_{d^\pi}$, note that

$$
\begin{aligned}
\|Q^\pi - (\mathcal{T}^\pi)^\infty Q_0\|_{d^\pi} &= \|\mathcal{T}^\pi Q^\pi - \mathcal{T}^\pi (\mathcal{T}^\pi)^\infty Q_0\|_{d^\pi} \\
&\leq \gamma \|Q^\pi - (\mathcal{T}^\pi)^\infty Q_0\|_{d^\pi}
\end{aligned}
$$

The inequality uses Lem. A.1. Recursively, we get $\|Q^\pi - (\mathcal{T}^\pi)^\infty Q_0\|_{d^\pi} = 0$.

To sum up, we have

$$
\begin{aligned}
J(\pi^*) - J(\pi) &\leq \sum_{t=0}^{\infty} \gamma^t \left(\left\|Q^* - \widetilde{Q}\right\|_{d^\pi \times \pi^*} + \left\|Q^* - \widetilde{Q}\right\|_{d^\pi} \right) \\
&\leq \frac{2\gamma}{1-\gamma} \left(\|Q^* - Q_0\|_{d^\pi \times \pi_{Q^*,Q_0}} + \|Q^\pi - Q_0\|_{d^\pi} \right) + \frac{\sqrt{C_{\pi,\pi^*}} + 1}{1+\gamma} \frac{1}{(1-\gamma)^{2.5}} \sqrt{\frac{56 R_{\max}^2 \log \frac{|\mathcal{F}|^2}{\delta}}{3n}} \\
&\leq \frac{2\gamma}{1-\gamma} \left(\|Q^* - Q_0\|_\infty + \|Q^\pi - Q_0\|_\infty \right) + \frac{\sqrt{C_{\pi,\pi^*}} + 1}{1+\gamma} \frac{1}{(1-\gamma)^{2.5}} \sqrt{\frac{56 R_{\max}^2 \log \frac{|\mathcal{F}|^2}{\delta}}{3n}}.
\end{aligned}
$$

Then we conclude the proof.

A.3. Proof of Thm. 4.3

Let the state distribution of the combined buffer as $d(s)$, the concentratability coefficient between d^π and d as C_d, i.e., $C_d = \max_s \frac{d^\pi(s)}{d(s)}$. We introduce the following lemmas.

Lemma A.5. *D_1, D_2 are two dataset containing N samples and n samples, respectively. If the distribution of D_1 is μ and D_2 is ν, the concentratabilty coefficient between ν and μ is C, then the concentratability coefficient between ν and combined buffer ρ is $\frac{(N+n)C}{N+nC}$.*

Proof. The distribution of combined buffer is $\rho = \frac{N\mu(s)+n\nu(s)}{N+n}$.

$$\begin{aligned}\max_s \frac{\nu(s)}{\rho(s)} &= \max_s \frac{(N+n)\nu(s)}{N\mu(s)+n\nu(s)} \\ &= \max_s \frac{N+n}{N\frac{\mu(s)}{\nu(s)}+n} \\ &= \frac{(N+n)C}{N+nC}.\end{aligned}$$

□

Lemma A.6 (Lemma 12 in (Chen & Jiang, 2019))**.** *Let $\mu(s)$ be any admissible distribution, C be the concentratablility coefficient of $\mu(s)$ and $\nu(s)$, then $\|\cdot\|_\nu \le \sqrt{C}\|\cdot\|_\mu$.*

Note that Eq. (11) and (12) hold whatever the distribution of training data is. By replacing ν in Eq. (12) with $d \times \pi^*$ and $d \times \pi_{Q^*,Q_0}$, and apply Lem. A.6, Lem. A.1 and Lem. A.5:

$$\begin{aligned}\|\cdot\|_{d\times\pi^*} &\le \sqrt{\widetilde{C_d} C_{\pi,\pi^*}}\,\|\cdot\|_{d^\pi}, \\ \|\cdot\|_{d\times\pi_{Q^*,Q_0}} &\le \sqrt{\widetilde{C_d} C_{\pi,\pi^*}}\,\|\cdot\|_{d^\pi},\end{aligned}$$

where $\widetilde{C_d} = \frac{(N+n)C_d}{N+nC_d}$.

Similar to the derivation of last section, we have

$$J(\pi^*) - J(\pi) \le \frac{2\gamma}{1-\gamma}\left(\|Q^* - Q_0\|_\infty + \|Q^\pi - Q_0\|_\infty\right) + \frac{\sqrt{\widetilde{C_d} C_{\pi,\pi^*}} + 1}{(1-\gamma)^4}\frac{56R_{\max}^2 \log\frac{|\mathcal{F}|^2}{\delta}}{3(n+N)}$$

A.4. Proof of Thm. 4.4

Define th transition function learned by the model is $\widetilde{T}$. To simplify the notation, we use $\widetilde{\mathcal{T}}^\pi$ to denote the Bellman operator under learned model, i.e.,

$$\widehat{\mathcal{T}}^\pi f = \arg\min_g \sum_{i=1}^N (g(s,a) - r(s,a) - \gamma V_f^\pi(\tilde{s}'))^2,$$

where $\tilde{s}' \sim \widetilde{T}(s'|s,a)$.

Lemma A.7. $\left\|(\mathcal{T}^\pi)^\infty Q_0 - (\widetilde{\mathcal{T}}^\pi)^\infty Q_0\right\|_{\hat{d}^\pi} \le \frac{\gamma}{1-\gamma} L\epsilon_m^d$

Proof. Note that

$$\left\|\mathcal{T}^\pi Q_0 - \widetilde{\mathcal{T}}^\pi Q_0\right\|_{\hat{d}^\pi} = \gamma \left\|Q_0(s',\pi) - Q_0(\tilde{s}',\pi)\right\|_{\hat{d}^\pi},$$

where $s' \sim T$, $\tilde{s}' \sim \widetilde{T}$.

Since Q_0 is L-Lipschitz,

$$\|Q_0(s',\pi) - Q_0(\tilde{s}',\pi)\|_{\hat{d}^\pi} \le L \|s' - \tilde{s}'\|_{\hat{d}^\pi} \le L\epsilon_m^d$$

Then,

$$\begin{aligned}
&\left\|(\mathcal{T}^\pi)^2 Q_0 - (\widetilde{\mathcal{T}}^\pi)^2 Q_0\right\|_{\hat{d}^\pi} \\
&= \gamma \left\|(\mathcal{T}^\pi Q_0)(s',\pi) - (\widetilde{\mathcal{T}}^\pi Q_0)(\tilde{s}',\pi)\right\|_{\hat{d}^\pi} \\
&\le \gamma \left\|(\mathcal{T}^\pi Q_0)(s',\pi) - (\widetilde{\mathcal{T}}^\pi Q_0)(s',\pi)\right\|_{\hat{d}^\pi} + \left\|(\widetilde{\mathcal{T}}^\pi Q_0)(s',\pi) - (\widetilde{\mathcal{T}}^\pi Q_0)(\tilde{s}',\pi)\right\|_{\hat{d}^\pi} \\
&\overset{(a)}{=} \gamma \left\|(\mathcal{T}^\pi Q_0) - (\widetilde{\mathcal{T}}^\pi Q_0)\right\|_{\hat{d}^\pi} + \left\|(\widetilde{\mathcal{T}}^\pi Q_0)(s',\pi) - (\widetilde{\mathcal{T}}^\pi Q_0)(\tilde{s}',\pi)\right\|_{\hat{d}^\pi} \\
&\le \gamma^2 L\epsilon_m^d + \gamma L\epsilon_m^d,
\end{aligned}$$

where (a) is because d^π is stationary distribution.

Similarly, we have

$$\left\|(\mathcal{T}^\pi)^\infty Q_0 - \widetilde{\mathcal{T}}^\infty Q_0\right\|_{\hat{d}^\pi} \le \sum_{t=1}^{\infty} \gamma^t L\epsilon_m^d = \frac{\gamma}{1-\gamma} L\epsilon_m^d.$$

□

Lemma A.8. *Let ν and μ be any state distribution, $f : \mathcal{S} \to \mathbb{R}$ be any function with $|f| \le C$, $\left|\|f\|_\nu - \|f\|_\mu\right| \le C\|\nu - \mu\|$.*

Proof.

$$\begin{aligned}
\left|\|f\|_\nu - \|f\|_\mu\right| &= \left|\sqrt{\sum_s |f(s)|^2 \nu(s)} - \sqrt{\sum_s |f(s)|^2 \mu(s)}\right| \\
&\overset{(a)}{=} \frac{\left|\|f\|_\nu^2 - \|f\|_\mu^2\right|}{\|f\|_\nu + \|f\|_\mu} \\
&\overset{(b)}{\le} \sqrt{\left|\|f\|_\nu^2 - \|f\|_\mu^2\right|} \\
&\le \sqrt{C^2 |\nu(s) - \mu(s)|} \\
&= C\sqrt{\|\nu - \mu\|}
\end{aligned}$$

(a) is because $a^{1/2} - b^{1/2} = \frac{a-b}{\sqrt{a}+\sqrt{b}}$. (b) is because

$$\sqrt{\left|\|f\|_\nu^2 - \|f\|_\mu^2\right|} \le \|f\|_\nu + \|f\|_\mu.$$

□

Similar to the analysis in the previous section, $\left\|Q^* - \widetilde{Q}\right\|_\nu$ can be decomposed into three terms.

$$\left\|Q^* - \widetilde{Q}\right\|_\nu \le \gamma \|Q^* - Q_0\|_{\nu \times \pi_{Q^*, Q_0}} + \|Q^\pi - (\mathcal{T}^\pi)^\infty Q_0\|_\nu + \left\|(\mathcal{T}^\pi)^\infty Q_0 - (\widetilde{\mathcal{T}}^\pi)^\infty Q_0\right\|_\nu \tag{13}$$

According to last section, we only need to consider $\nu = d^\pi$, and obtain that $\|Q^\pi - (\mathcal{T}^\pi)^\infty Q_0\|_{d^\pi} = 0$.

Therefore, we focus on the last term $\left\|(\mathcal{T}^\pi)^\infty Q_0 - (\widetilde{\mathcal{T}}^\pi)^\infty Q_0\right\|_{d^\pi}$.

$$\left\|(\mathcal{T}^\pi)^\infty Q_0 - (\widetilde{\mathcal{T}}^\pi)^\infty Q_0\right\|_{d^\pi} \overset{(a)}{\leq} V_{\max}\epsilon_d + \left\|(\mathcal{T}^\pi)^\infty Q_0 - (\widetilde{\mathcal{T}}^\pi)^\infty Q_0\right\|_{\hat{d}^\pi} \overset{(b)}{\leq} V_{\max}\epsilon_d + \frac{\gamma}{1-\gamma} L\epsilon_m^d,$$

where (a) uses Lem. A.8, (b) uses Lem. A.7.

Then we have

$$\begin{aligned} J(\pi^*) - J(\pi) &\leq \frac{2\gamma}{1-\gamma}\left(\|Q^* - Q_0\|_\infty + \|Q^\pi - Q_0\|_\infty\right) + \left(\sqrt{C_{\pi,\pi^*}} + 1\right)\left(\frac{1}{1-\gamma} V_{\max}\epsilon_d + \frac{\gamma}{(1-\gamma)^2} L\epsilon_m^d\right) \\ &= \frac{2\gamma}{1-\gamma}\left(\|Q^* - Q_0\|_\infty + \|Q^\pi - Q_0\|_\infty\right) + \left(\sqrt{C_{\pi,\pi^*}} + 1\right)\left(\frac{1}{(1-\gamma)^2} R_{\max}\epsilon_d + \frac{\gamma}{(1-\gamma)^2} L\epsilon_m^d\right) \end{aligned}$$

A.5. Proof of Thm. 4.5

Suppose the dataset generated by the traditional model is $\{(s, a, r, s')\}$. Let the distribution of s in the dataset be $\tilde{d}^\pi$. The next lemma aims to bound the error between $\tilde{d}^\pi$ and the real distribution d^π:

Lemma A.9 (Lemma B.2 in (Janner et al., 2019))**.** *The distance in the state marginal distribution is bounded as:*

$$\left\|\tilde{d}^\pi - d^\pi\right\| \leq \frac{1}{1-\gamma}\epsilon_m^t.$$

By replacing ϵ_d with $\frac{1}{1-\gamma}\epsilon_m^t$, we get the final result:

$$J(\pi^*) - J(\pi) \leq \frac{2\gamma}{1-\gamma}\left(\|Q^* - Q_0\|_\infty + \|Q^\pi - Q_0\|_\infty\right) + \left(\sqrt{C_{\pi,\pi^*}} + 1\right)\left(\frac{1}{(1-\gamma)^3} R_{\max}\epsilon_m^t + \frac{\gamma}{(1-\gamma)^2} L\epsilon_m^t\right)$$

B. Pseudo-Code

C. Experimental Details

C.1. Task Description

Adroit Manipulation. Our empirical evaluation on Adroit manipulation contains 3 domains: pen, door, relocate, where the RL agent is required to solve dexterous manipulation tasks including rotating a pen in specific directions, opening a door, and moving a ball, respectively. The offline datasets are human-v1 datasets in D4RL (Fu et al., 2020) benchmark, which only contain a few successful non-markovian human demonstrations and thus is pretty difficult for most offline RL approaches to acquire reasonable pre-training performances.

AntMaze Navigation. Our tests on Antmaze navigation benchmark consists of four datasets, namely umaze-v2, medium-diverse-v2, medium-play-v2 and large-play-v2 from the D4RL (Fu et al., 2020). The objective is for an ant to learn how to walk and navigate from the starting point to the destination in a maze environment, with only sparse rewards provided. This task poses a challenge for online RL algorithms to explore high-quality data effectively without the support of offline datasets or additional domain knowledge.

MuJoCo Locomotion. MuJoCo locomotion encompasses several standard locomotion tasks commonly utilized in RL research, such as Hopper, Halfcheetah, Walker2d. In each task, the RL agent is tasked with controlling a robot to achieve forward movement. The D4RL (Fu et al., 2020) benchmark provides four types of datasets with varying quality for each task: random-v2, medium-v2, medium-replay-v2, medium-expert-v2.

C.2. Details on Comparison with Model-based Methods

Algorithm 1 Energy-Guided Diffusion Sampling in Offline-to-Online Reinforcement Learning

Require: Offline phase loss function $\{\mathcal{L}^{Q_\psi}_{\text{offline}}, \mathcal{L}^{\pi_\omega}_{\text{offline}}\}$, online phase loss function $\{\mathcal{L}^{Q_\psi}_{\text{online}}, \mathcal{L}^{\pi_\omega}_{\text{online}}\}$, energy-based models $\{\mathcal{E}_{\phi_1}, \mathcal{E}_{\phi_2}, \mathcal{E}_{\phi_3}\}$, noise prediction model D_θ.
Initialize $\psi, \theta, \phi_1, \phi_2, \phi_3, \omega$, offline replay buffer $\mathcal{D}_{\text{offline}}$, online replay buffer $\mathcal{D}_{\text{online}}$, diffusion replay buffer $\mathcal{D}_{\text{diffusion}}$.
while in *offline training phase* **do**
  % offline policy training using batches from the offline replay buffer $\mathcal{D}_{\text{offline}}$
  $\psi \leftarrow \psi - \lambda_Q \nabla_\psi \mathcal{L}^Q_{\text{offline}}(\psi), \omega \leftarrow \omega - \lambda_\pi \nabla_\omega \mathcal{L}^\pi_{\text{offline}}(\omega)$.
end while
while in *online training phase* **do**
  for each environment step **do**
    $\mathcal{D}_{\text{online}} \leftarrow \mathcal{D}_{\text{online}} \cup \{(s, a, s', r)\}$
  end for
  if step meets D_θ update frequency **then**
    Sample data from $\mathcal{D}_{\text{offline}} \cup \mathcal{D}_{\text{online}}$.
    Update D_θ by minimizing Eq. (2).
    Sample positive samples from $\mathcal{D}_{\text{online}}$ and generate negative samples with D_θ based on the positive samples.
    Update $\mathcal{E}_{\phi_1}$ by minimizing Eq. (5).
    Sample positive samples from $\mathcal{D}_{\text{online}} \cup \mathcal{D}_{\text{offline}}$ and generate negative samples.
    Update $\mathcal{E}_{\phi_2}$ by minimizing Eq. (6).
    Sample positive samples from $\mathcal{D}_{\text{online}} \cup \mathcal{D}_{\text{offline}}$ and generate negative samples.
    Update $\mathcal{E}_{\phi_3}$ by minimizing Eq. (7).
    Score-based sampling with energy guidance and store them as $\mathcal{D}^\epsilon$, $\mathcal{D}_{\text{diffusion}} \leftarrow \mathcal{D} \cup \mathcal{D}^\epsilon$.
  end if
  for each gradient step **do**
    Sample data from $\mathcal{D}_{\text{online}} \cup \mathcal{D}_{\text{diffusion}}$.
    $\psi \leftarrow \psi - \lambda_Q \nabla_\psi \mathcal{L}^Q_{\text{online}}(\psi)$
    $\omega \leftarrow \omega - \lambda_\pi \nabla_\omega \mathcal{L}^\pi_{\text{online}}(\omega)$
  end for
end while

We conduct the experiment on comparison of distribution differences between the real online distribution and the distribution generated by models a Maze environment, which is visualized in Figure 3. With going up, down, left, and right as selectable actions, the agent starts at the upper left corner and the exit is at the lower right corner. The black blocks are occupied and inaccessible. The agent's goal is to reach the exit as quickly as possible, with every step the agent incurs a penalty or, when finally reaching the exit, a reward.

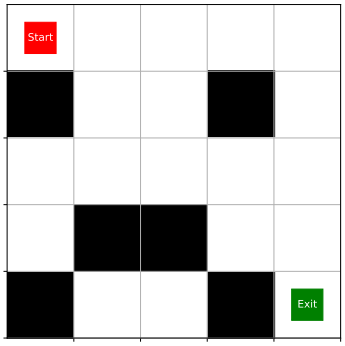

Figure 3. Visualization of the Maze MDP.

We visualize the distribution difference in Fig. 4. To quantify this difference, we compute the distribution difference as the deviation in the number of times the agent visited each state, normalized by a factor of 1000. Table C.3 explicitly illustrates the divergence values, underscoring that our EDIS effectively generates the intended distribution. In contrast, model-based approaches, particularly those utilizing MLP, struggle to accurately emulate the actual online distribution. While the distribution generated by the transition modeled by diffusion model shows better result, verifying diffusion model has better capability of modeling distributions. It is noteworthy that the offline dataset distribution exhibits a substantial divergence from the online distribution, rendering direct replay of the offline dataset impractical. Despite the online buffer having a distribution comparable to the real one, its limited dataset size poses a challenge to achieving optimal sample efficiency.

C.3. Details and Hyperparameters for EDIS

We use the PyTorch implementation of Cal-QL and IQL from https://github.com/tinkoff-ai/CORL, and primarily followed the author's recommended parameters (Tarasov et al., 2022). The hyperparameters used in our EDIS module are detailed in the Tab. 3:

Table 3. Hyperparameters and their values in EDIS

Hyperparameter	Value
Network Type (Denoising)	Residual MLP
Denoising Network Depth	6 layers
Denoising Steps	128 steps
Denoising Network Learning Rate	3×10^{-4}
Denoising Network Hidden Dimension	1024 units
Denoising Network Batch Size	256 samples
Denoising Network Activation Function	ReLU
Denoising Network Optimizer	Adam
Learning Rate Schedule (Denoising Network)	Cosine Annealing
Training Epochs (Denoising Network)	$50,000$ epochs
Training Interval Environment Step (Denoising Network)	Every $10,000$ steps
Energy Network Hidden Dimension	256 units
Negative Samples (Energy Network Training)	10
Energy Network Learning Rate	1×10^{-3}
Energy Network Activation Function	ReLU
Energy Network Optimizer	Adam

(a) Online (Real) Distribution (b) MLP Transition Distribution (c) Diffusion Distribution

(d) Diffusion Transition Distribution (e) Offline Distribution (f) Online Buffer Distribution

Figure 4. Illustration of the distribution differences between the actual online distribution and the three generated distributions in a Maze environment.

Table 4. Comparison of distribution divergence between the actual online distribution and generated data distributions in different methods

	Offline	Interaction	MLP Transition	Diffusion Transition	Diffusion
Divergence	0.29	0.18	0.44	0.24	0.30

Table 5. Divergence comparisons for energy function ablation study

Dataset	State Divergence		Action Divergence		Transition Divergence	
	w/o energy	w/ energy	w/o energy	w/ energy	w/o energy	w/ energy
hopper-radnom-v2	0.85±0.02	**0.73±0.03**	0.51±0.04	**0.39±0.02**	0.69±0.03	**0.66±0.04**
halfcheetah-medium-replay-v2	0.98±0.01	**0.50±0.05**	0.43±0.15	**0.31±0.03**	1.01±0.08	**0.88±0.07**
walker2d-medium-expert-v2	0.91±0.02	**0.65±0.12**	1.62±0.06	**1.32±0.03**	0.51±0.05	**0.33±0.08**
antmaze-medium-play-v2	0.99±0.01	**0.95±0.02**	0.68±0.09	**0.54±0.05**	2.05±0.13	**1.85±0.14**
antmaze-medium-diverse-v2	0.98±0.00	**0.91±0.02**	0.38±0.08	**0.27±0.08**	0.75±0.14	**0.64±0.14**
door-human-v1	0.40±0.04	**0.24±0.02**	0.54±0.07	**0.41±0.08**	2.57±0.06	**2.52±0.05**

C.4. Additional Ablation Study

We conducted additional ablation studies on each energy function by omitting their guidance during the reverse-time *SDE* in a total of six environments, including hopper-random-v2, halfcheetah-medium-replay-v2, antmaze-medium-play-v2, antmaze-medium-diverse-v2, and door-human-v1. It can be observed that each energy function plays a functional role and reduces the corresponding divergence value.

The state divergence here is the JS divergence between the two state distributions, which is defined as:

$$D_{\mathrm{JS}}(p, q) = \frac{1}{2}\left(D_{\mathrm{KL}}(p, (p+q)/2) + D_{\mathrm{KL}}(q, (p+q)/2)\right),$$

where p and q are two distributions and D_{KL} is

$$D_{\mathrm{KL}}(p, q) = \int \log(p(x)/q(x))dx.$$

If the two distributions are similar, JS divergence will approach zero, or it will approach one. To calculate it, we apply the techniques of GAN (Goodfellow et al., 2014). We learn a discriminator by minimizing the following loss function

$$V(D) = \mathbb{E}_{x\sim p}[\log D(x)] + \mathbb{E}_{z\sim q}[\log(1 - D(z)],$$

and JS divergence can be derived according to the following formula:

$$\max_D V(D) = -\log 4 + 2D_{\mathrm{JS}}(p, q).$$

The aggregated learning curves, summarizing outcomes across all environments from Tab.1, are displayed in Fig 6 and Fig. 7. They compare the performance of Cal-QL and IQL, both augmented with EDIS, against their respective base algorithms. For the evaluation of the normalized score in sparse reward domains, we computed a metric that represents the goal achievement rate for each method. For example, in the adroit environment door-human, we assessed the success rate of opening the door.

C.5. Experiments on Visual Environment

We conduct extra experiments on DMC, which is pixel-based state observations. Our EDIS operates on encoded representations, which are relatively low-dimensional. By integrating EDIS with CQL+SAC, a commonly used offline to online baseline, we observed significant improvements on walker-walk and cheetah-run tasks. In future work, we will explore even more complex environments. This expansion will help further refine our approach and validate its generalization.

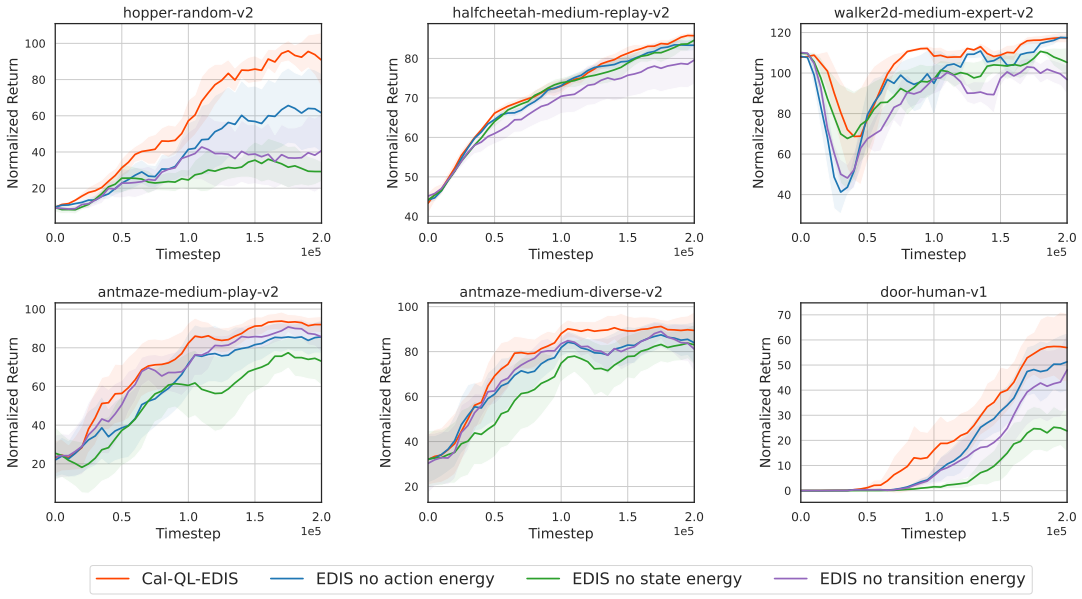

Figure 5. Energy Module Abaltion Study of EDIS

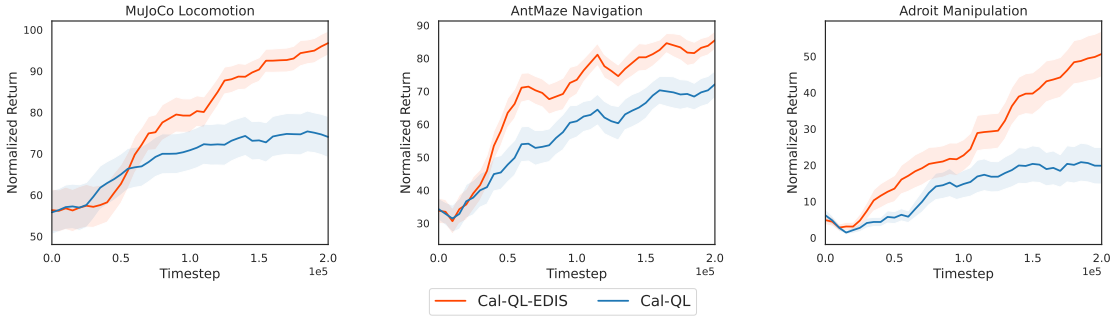

Figure 6. Aggregated learning curves of Cal-QL and Cal-QL-EDIS on MuJoCo Locomotion, AntMaze Navigation and Adroit Manipulation tasks

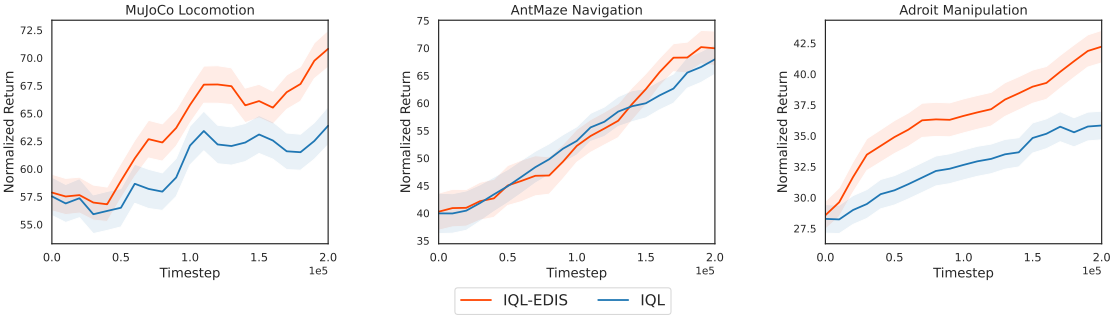

Figure 7. Aggregated learning curves of IQL and IQL-EDIS on MuJoCo Locomotion, AntMaze Navigation and Adroit Manipulation tasks

Table 6. Experiments on the remaining antmaze environments.

Environment	Cal-QL	Cal-QL-EDIS	IQL	IQL-EDIS
antmaze-umaze-diverse-v2	93.4±4.6	**95.9±2.8**	51.3±4.5	**66.7±5.0**
antmaze-large-diverse-v2	42.3±2.2	**57.1± 2.8**	45.0±8.7	**52.1±2.6**

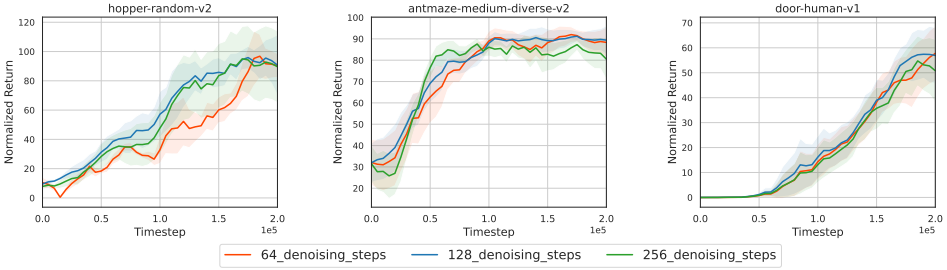

Figure 8. Sensitivity analysis on the training steps for the diffusion model.

C.6. Experiments on Sensitivity Analysis

In our sensitivity analysis, we evaluated the effect of varying the diffusion model's denoising steps and the number of negative samples. We tested denoising steps within the range of 64 to 256 and found comparable performance across this spectrum, indicating that EDIS is robust to changes in this hyperparameter. On the other hand, reducing the number of negative samples from 10 to 5 resulted in a noticeable decline in performance. However, maintaining a minimum of 10 negative samples ensures that performance remains consistent and unaffected.

C.7. Computational Resources

We train EDIS integrated with base algorithms on an NVIDIA RTX 4090, with approximately 4 hours required for 0.2M fine-tuning on MuJoCo Locomotion and Adroit Manipulation, while 6 hours for AntMaze Navigation. Specifically, the time cost for each time training our EDIS every time for 50000 epochs is about 10 minutes. The detailed computational consumption is shown in Tab. 6. As pointed out in (Karras et al., 2022), the sampling time is faster than prior diffusion designs, which is much shorter compared with training. The introduction of the diffusion model does indeed entail an inevitable increase in computational and time costs. However, this tradeoff between improved performance and higher computational cost is a common consideration in diffusion model research. As the field progresses, we anticipate that better solutions will emerge. In our future work, we aim to further refine and optimize the extra costs.

D. Additional Related Work

D.1. Offline-to-online Reinforcement Learning.

Offline-to-online reinforcement learning methods are developed to mitigate the dichotomy between the costly exploration in online RL and the typically suboptimal performance of offline RL. Previous existing methods do this in a variety of

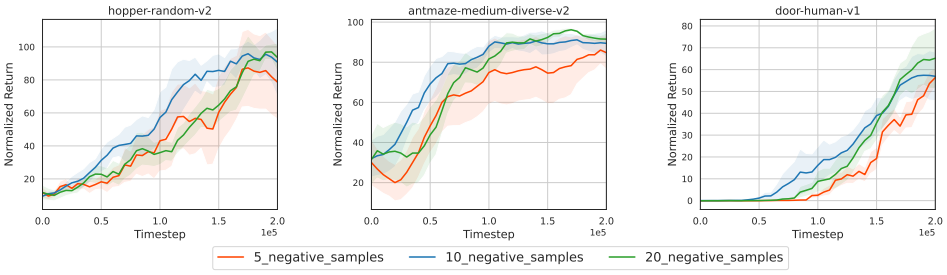

Figure 9. Sensitivity analysis on the number of negative samples.

Table 7. Computational consumption of different algorithms.

Algorithm	Online phase training time	Maximal GPU memory
Cal-QL (MuJoCo)	3h	2GB
Cal-QL-EDIS (MuJoCo)	5h	2GB
Cal-QL (AntMaze)	3h	2GB
Cal-QL-EDIS (AntMaze)	5h	3GB
Cal-QL (Adroit)	3h	2GB
Cal-QL-EDIS (Adroit)	6h	3GB
IQL (MuJoCo)	2h	2GB
IQL-EDIS (MuJoCo)	3h	3GB
IQL (AntMaze)	3h	2GB
IQL-EDIS (AntMaze)	7h	3GB
IQL (Adroit)	2h	2GB
IQL-EDIS (Adroit)	5h	3GB

ways: incorporating the offline data into the replay buffer of online RL (Schaal, 1996; Vecerik et al., 2017; Song et al., 2022; Hester et al., 2018), imitating offline policy as auxiliary losses (Zhu et al., 2018; Rajeswaran et al., 2018; Zhu et al., 2019), or extracting a high-level skill space for downstream online RL (Gupta et al., 2019; Ajay et al., 2021). Although these methods improve the sample efficiency of online RL from scratch, they cannot eliminate the need to actively rollout poor policies for data collection (Nakamoto et al., 2023).

Another line of work, typically divide the learning process into two phases. Warming up the policy and value functions in the offline phase and use them as initialization in the online phase (Zhao et al., 2022; Nair et al., 2020; Wang et al., 2023a; Nakamoto et al., 2023; Lee et al., 2021; Kostrikov et al., 2022). These approaches often employ offline RL methods based on policy constraints or pessimism (Fujimoto et al., 2019; Fujimoto & Gu, 2021; Kumar et al., 2020) in the offline phase. However, the conservatism conflicts with the online phase and may induce performance degradation. Various strategies have been implemented to tackle this issue, such as policy expansion (Zhang et al., 2023), value function calibration (Nakamoto et al., 2023), Q-ensemble techniques (Lee et al., 2021), and constraint methods (Nair et al., 2020; Kostrikov et al., 2022; Li et al., 2023).

Despite the advancements, there has been less focus on the crucial aspect of integrating useful data during the fine-tuning phase to enhance training efficiency. Standard practices includes enriching the replay buffer with offline data (Nakamoto et al., 2023; Zhang et al., 2023), adopting balanced sampling methods for managing both online and offline data sources (Lee et al., 2021; Ball et al., 2023), or building models to performance branch rollout (Rafailov et al., 2023). However, directly replaying the offline data causes a distribution shift, and adopting balanced sampling methods introduces large variance (Nachum et al., 2019) while rolling in the built model suffers from compounding error (Janner et al., 2019). In contrast, our work breaks new ground by proposing a diffusion-based generator specifically designed to generate useful samples by thoroughly leveraging prior knowledge in offline data.

D.2. Diffusion Model in Reinforcement Learning

Diffusion models have demonstrated exceptional capabilities in modeling distribution (Saharia et al., 2022; Nichol et al., 2022; Nichol & Dhariwal, 2021). Within the reinforcement learning research, Diffuser (Janner et al., 2022) uses a diffusion model as a trajectory generator and learns a separate return model to guide the reverse diffusion process toward samples of high-return trajectories. The consequent work, Decision Diffuser (Ajay et al., 2023) introduces conditional diffusion with reward or constraint guidance for decision-making tasks, further boosting Diffuser's performance. Diffusion-QL (Wang et al., 2023b) tracks the gradients of the actions sampled from the behavior diffusion policy to guide generated actions to high Q-value area. SfBC (Chen et al., 2023a) and Diffusion-QL both employ the technique of resampling actions from multiple behavior action candidates using predicted Q-values as sampling weights. Expanding the application of diffusion models, SYNTHER (Lu et al., 2023a) focuses on leveraging the diffusion model for upsampling data in both online and offline reinforcement learning scenarios. DVF (Mazoure et al., 2023) introduces a diffusion model to learn the transition dynamics, which can then be used to estimate the value function. Recently, several concurrent work also investigates generating samples with certain data distribution by diffusion models. PolyGRAD (Rigter et al., 2023) and PGD (Jackson et al., 2024) embed the policy for classifier-guided trajectory generation, aiming at on-policy world modeling. However, these studies do not focus on the offline-to-online setting, and they model the transition function rather than the distribution directly, which does not eliminate the issue of compounding error. In the context of offline-to-online reinforcement learning,

our research pioneers the utilization of diffusion-based models to actively generate valuable samples. This departure from passive reuse of offline data marks a novel approach, emphasizing the active role of diffusion models in sample generation for enhanced learning outcomes.