Off-dynamics Conditional Diffusion Planners

Wen Zheng Terence Ng, Jianda Chen, Tianwei Zhang
Nanyang Technological University, Singapore
{ngwe0099, jianda001}@e.ntu.edu.sg, tianwei.zhang@ntu.edu.sg

***Abstract*— Offline Reinforcement Learning (RL) offers an attractive alternative to interactive data acquisition by leveraging pre-existing datasets. However, its effectiveness hinges on the quantity and quality of the data samples. This work explores the use of more readily available, albeit off-dynamics datasets, to address the challenge of data scarcity in Offline RL. We propose a novel approach using conditional Diffusion Probabilistic Models (DPMs) to learn the joint distribution of the large-scale off-dynamics dataset and the limited target dataset. To enable the model to capture the underlying dynamics structure, we introduce two contexts for the conditional model: (1) a *continuous dynamics score* allows for partial overlap between trajectories from both datasets, providing the model with richer information; (2) an *inverse-dynamics context* guides the model to generate trajectories that adhere to the target environment's dynamic constraints. Empirical results demonstrate that our method significantly outperforms several strong baselines. Ablation studies further reveal the critical role of each dynamics context. Additionally, our model demonstrates that by modifying the context, we can interpolate between source and target dynamics, making it more robust to subtle shifts in the environment.**

I. Introduction

Conventional Reinforcement Learning (RL) excels at learning from live interactions within its environment, but real-time data acquisition can be resource-intensive, dangerous or infeasible in critical domains. *Offline RL* emerges as a powerful alternative, leveraging pre-existing datasets without live interactions to train control policies [1], [2]. This approach opens doors to training intelligent agents across a wide range of applications, including life-saving surgical robots [3], [4], data-driven financial trading [5], [6], improved medical diagnosis [7], [8], and autonomous vehicles [9]–[11]. Despite its potential, its effectiveness heavily relies on the quantity and quality of available data, with performance deteriorating sharply when data is scarce [12], [13]. However, collecting large-scale expert-level data from offline sources remains challenging despite not requiring live interaction with the environment.

This work explores the use of more accessible source datasets to address the challenge of data scarcity in Offline RL. We draw inspiration from the success of transfer learning in supervised learning [14], [15], where researchers have increasingly utilized data from additional, easier-to-obtain sources [16]–[18]. In the context of Offline RL, we apply this principle by employing a readily available *off-dynamics* source dataset that aligns with the target task's objective but possesses different transition dynamics compared to the target environment. This strategy can be well applicable to real-world settings. For instance, self-driving cars can leverage data from diverse cities or vehicle models; medical diagnosis can benefit from similar but non-identical illness data; surgical robots can initially train on artificial organs; financial trading algorithms can utilize data from larger, related markets to inform decisions in smaller ones.

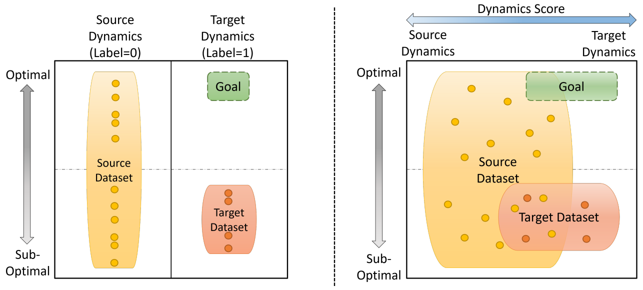

Fig. 1: (Left) We utilize an accessible off-dynamics source dataset to enhance a limited target dataset for Offline RL. Our goal is to generate optimal trajectories within the green region. (Right) By conditioning a diffusion planner with our proposed continuous dynamics score, we enable the model to capture the underlying dynamics structure within the latent space through overlapping dynamics information.

Several approaches have been proposed to leverage off-dynamics samples to boost RL training. One popular method is to introduce a reward bonus to incentivize agents for taking actions which end up resembling the target environment [13], [19]–[22]. However, this incurs extra cost and renders the learning of the primary task to be less effectively due to the embedded nature of the reward bonus. An alternative method is to apply transformations to the source domain [23]–[25], making it behave like the target domain. This allows the agent to learn as if it were in the target domain. However, this requires *online interactions* with the target environment, making it unsuitable for our problem setting.

To address these limitations, we propose to utilize the flexibility and expressiveness of *diffusion probabilistic models* (DPMs) [26]–[28] to learn a joint distribution of both source and target data for greater data efficiency. DPMs have demonstrated impressive capabilities in image generation [29], [30], audio generation [31], video generation [32] and more recently in Offline RL [33]–[35]. To enable our model to generate trajectories for the target environment, we utilize classifier-free guidance [36] by conditioning our model with dynamics-related contexts. For this context, we propose a *continuous dynamics score* as an alternative to the discrete dynamics labels as seen in Figure 1. This "soft score" allows for greater coverage, enabling overlap between trajectories from different datasets. Furthermore, we incorporate an inverse-dynamics context that measures the closeness to the target dynamics, ensuring the generated trajectories adhere to the target environment's dynamic constraints. Together, the context facilitates capturing the underlying dynamics structure within the latent space more effectively, leading to improved generation performance.

We conduct comprehensive experiments to evaluate the effectiveness of our method on a diverse set of challenging off-dynamics settings. Empirical results demonstrate that our method outperforms several strong baselines. The simplicity of the proposed dynamics-related contexts coupled with the powerful capabilities of DPMs allow us to effectively leverage an accessible off-dynamics source dataset. Lastly, we demonstrate that by modifying the context, we can interpolate between source and target dynamics, making the model more robust to subtle shifts.

II. Related Work

A. Off-dynamics Reinforcement Learning

Coined by [19], *Off-dynamics RL* refers to a unique RL scenario where the transition dynamics in the source domain, used to train the policy, differ from those in the target domain. It can be applied to two settings, as described below.

Off-dynamics Online RL. In the online setting, it is assumed that online samples from both domains are accessible, typically with a limited amount available in the target domain. In [19], a reward adjustment is proposed to encourage agents to favour transitions that are similar to the target dynamics when selecting actions during source domain training. The reward adjustment is based on auxiliary classifiers that are trained separately to differentiate between source-domain and target-domain transitions. This concept of utilizing dynamics classifiers has also been adapted to other off-dynamics problems such as model-based RL [20], inverse RL [37] and unsupervised RL [22].

A special case of off-dynamics RL involves the use of simulators as the source domain [23]–[25], [38]. Instead of reward adjustments, these approaches *ground* the simulator by applying a transformation on the source domain [23]–[25] to make it behave like the target domain. In this case, the agent can learn as if it were in the target domain. These transformation models employ forward and/or inverse dynamics models of the source and/or target domain. Instead of learning any dynamics models, some approaches [38], [39] involve using a classifier by redefining the task as an adversarial learning problem [40], [41].

Off-dynamics Offline RL. In the offline setting, Niu et al. [21] introduce a method that combines offline samples from the target domain with off-dynamics online samples from the simulator. Lastly, the most relevant work to our setting, [13], exclusively uses offline samples from both the source and target domains. Both of these approaches closely align with the method described in [19], applying a reward adjustment to the offline samples to facilitate learning.

B. Diffusion Models

Diffusion Probabilistic Models. These generative models [26]–[28] have gained significant attention for their ability to produce high-quality, diverse samples across various domains such as images [29], [30], [42], audio [31], [43] and video [32]. They learn to gradually construct data by reversing a process that adds noise to the data over several steps. Some of the key benefits include: (1) enabling the creation of flexible models that can accommodate arbitrary data structures, (2) enabling training in a tractable manner, and (3) exhibiting high stability during training and lower susceptibility to mode collapse, unlike in Generative Adversarial Networks (GANs) [44], [45].

Conditional DPMs. To enhance the capabilities of DPMs, conditional DPMs are introduced through classifier-guided sampling [42] and a classifier-free approach [36]. They have been applied to a broad spectrum of tasks. A notable application involves generating images from textual descriptions [29], [30], [42], where the generative model produces visuals based on the provided text input. Besides utilizing textual contexts for conditioning, other contexts have been explored for conditional image generation, e.g., low resolution contexts for super-resolution [46], binary masks contexts for in-painting [47] and semantic masks, human poses contexts for ControlNet [48]. Beyond image generation, conditional DPMs have also achieved success in various fields. e.g. 2D molecular graph contexts for 3D graph generation [49], partial point-cloud observations contexts for point-cloud synthesis [50], and linguistic features contexts for audio generation [31].

C. Diffusion-based Planners.

Recently, diffusion-based planners [33], [34] utilize DPMs to generate trajectories which address the *Offline RL* challenges as discussed in [1], [51]. By leveraging the powerful generative capabilities of DPMs, diffusion-based planners are able to outperform existing offline RL methods while having the additional benefits of specifying flexible constraints or composing multiple skills [33]–[35]. They have been adopted for various problem settings or tasks, such Multi-agent RL [52], Meta-RL [53], Multi-task RL [54] and Safe RL [55]. To maximize the return of the generated trajectories in these diffusion-based planners, Diffuser [33] uses classifier-guided sampling [42] whereas Decision-Diffuser [34] adopts a classifier-free approach [36].

One limitation of diffusion-based planners is the long inference time, attributed to the slow sampling process of DPMs. Researchers have proposed several methods to accelerate this process [56], [57]. While numerous follow-up works on diffusion-based planners exist, none specifically focus on utilizing an off-dynamics dataset to train the diffusion-based planners. The most relevant work to ours is [53], where a diffusion-planner uses trajectories generated from a large number of off-dynamics environments for meta-training, aiming to generalize to the target domain. In contrast, *our approach concentrates on adaptation rather than generalization, utilizing only a single source domain.*

III. Preliminaries

A. Reinforcement Learning

Markov Decision Process. In RL [58], an environment is characterized by a Markov Decision Process (MDP) defined as $\mathcal{M} = (S, A, P, R, \gamma, d_0)$, where S and A represent the state and action spaces, respectively . The transition dynamics are denoted by $P : S \times A \times S \to [0,1]$, and the reward function is represented by $R : S \times A \times S \to \mathbb{R}$. The discount factor is a scalar $\gamma \in [0,1)$ representing how much the agent prioritizes immediate rewards over future rewards and d_0 defines the initial state distribution. An agent's policy $\pi : S \to A$ induces a probability distribution over trajectories $\tau = (s_t, a_t, r_t)_{t \geq 0}$, where s_t, a_t, r_t are the state, action and reward at timestep t, respectively. For a planning horizon of H, the expected return for a trajectory is given by $R(\tau) = \sum_{t=0}^{H} \gamma^t r_t$. The goal of RL is to find the optimal policy π^* that maximizes the expected cumulative discounted reward with the objective: $\pi^* = \arg\max_\pi \mathbb{E}_{\tau \sim p_\pi}[R(\tau)]$.

Offline RL. In contrast to traditional RL, the offline RL framework [1], [59] employs a static dataset $\mathcal{D} = \{(s, a, r, s')\}$ for learning. Here, s' denotes the subsequent state following the application of action a from state s, bypassing the need for data collection through direct environment interaction using a policy, π. This dataset is collected by an unknown behavior policy μ, and is utilized for the learning of a new policy entirely from $\mathcal{D}$, without any online interactions with the environment. This resembles supervised learning, where $\mathcal{D}$ serves as the training set. A significant problem in offline RL is the distribution shift between the behavior policy and the learnt policy [1], resulting in overly optimistic value estimates when evaluating unseen states or actions. To address this, prior works explicitly constrain the learnt policy to be close to the behavior policy [60], [61], or reduce the optimism of the value functions [62], [63]. An alternative approach involves bypassing the value estimate by framing offline RL as a sequence modeling problem [33], [51], [64].

B. Diffusion Models

Diffusion Probabilistic Models. DPMs [26]–[28] form a class of generative models that learn data distributions $q(x)$ through a process that inversely models the addition of noise. Specifically, data points sampled from $p_{\text{data}}(x)$ undergo a forward noising sequence to produce a Markov chain $x_0 : K$ defined as $x_k \sim \mathcal{N}(\sqrt{\alpha_k} x_{k-1}, (1-\alpha_k)I)$, with $\alpha_{0:K}$ being the noise schedule and K being the number of diffusion steps. This chain is reversed by a variational process $p_\theta(x_{k-1}|x_k) = \mathcal{N}(x_{k-1}|\mu_\theta(x_k,k), (1-\alpha_k)I)$, beginning with $x_K \sim \mathcal{N}(0,I)$ and iteratively denoised until x_0 is recovered. The procedure can be optimized through a surrogate loss [28]:

$$\mathcal{L}(\theta) = \mathbb{E}_{k \sim [1,K], x_0 \sim q, \varepsilon \sim \mathcal{N}(0,I)} \left\| \varepsilon - \varepsilon_\theta(x_k, k) \right\|^2 \tag{1}$$

where the mean of the reverse Gaussian is given by $\mu_\theta(x_k,k) = \frac{x_k - \sqrt{1-\alpha_k}\varepsilon_\theta(x_k,k)}{\sqrt{\alpha_k}}$, with $\bar{\alpha}_k = \prod_{s=1}^{k} \alpha_s$.

Conditional Diffusion Models. The DPM method can be extended to a conditional generative model $p_\theta(x_{t-1}|x_t, c)$ using c as an input context through classifier-free guidance [36]. During sampling, the predicted noise is adapted to a weighted combination of conditional and non-conditional sampling, $\hat{\varepsilon}_\theta(x_t \mid c) = (1+w)\varepsilon_\theta(x_t \mid c) + \varepsilon_\theta(x_t \mid \emptyset)$, where $\emptyset$ represents the null context. and w regulates the trade-off between sample quality and diversity by balancing the conditioned and unconditioned models. In practice, the unconditioned model is obtained by applying dropout on the context embedding. While training a conditional model with classifier-guided sampling [42] is possible, classifier-free guidance provides advantages such as improved control over generation and superior performance [30], [34].

Diffusion Planning By framing Offline RL as a sequence modeling problem, we can simplify this task as a supervised learning and planning framework [64]. Building on this concept, *Diffuser* [33] is introduced as a diffusion-based trajectory planning model that utilizes expressive DPMs to model trajectories in the following form:

$$\tau = \begin{bmatrix} s_0 & s_1 & \dots & s_H \\ a_0 & a_1 & \dots & a_H \end{bmatrix}, \tag{2}$$

where H is the planning horizon. The model is optimized based on Equation 1, with $\varepsilon_\theta(\tau_k, k)$ being modeled by U-Nets [65], chosen for their non-autoregressive, temporally local, and equivariant characteristics. Conditional sampling is employed to generate trajectories that maximize return. By defining $\mathcal{O}_t$ as a binary random variable indicating the optimality of timestep t in a trajectory, where $p(\mathcal{O}_t = 1) = \exp(\gamma^t r(s_t, a_t))$, it is sufficient to guide the trajectory with the gradient of the return: $\nabla \mathcal{J} = \sum_{t=0}^{T} \nabla_{\mathbf{s}_t, \mathbf{a}_t} r(\mathbf{s}_t, \mathbf{a}_t) = \nabla_\tau \log p(\mathcal{O}_{1:T} \mid \tau)$. A separate model $\mathcal{J}_\phi$ is trained to predict the cumulative rewards, and the gradients of $\mathcal{J}_\phi$ are used to guide the trajectory following the classifier-guided sampling procedure [42]. Following *Diffuser*, *Decision-Diffuser* [34] adopts a classifier-free approach [36], utilizing reward information as context. In this work, we adopt the classifier-free approach due to the ease of incorporating additional contexts beyond return.

C. Problem Formulation: Off-Dynamics Offline RL

In this paper, we aim to enhance a limited target dataset offline using an accessible off-dynamics source dataset. We go beyond the standard offline RL framework of using a single fixed static offline dataset $\mathcal{D}_{\text{target}}$. Formally, similarly defined in [13], we incorporate an additional source dataset $\mathcal{D}_{\text{source}} = (s, a, r, s')$, collected by another unknown behavior policy μ_{source}, where $\mu_{\text{source}} \neq \mu_{\text{target}}$. A key distinction is that the source dataset is derived from an easily accessible source MDP $\mathcal{M}_{\text{source}}$, which exhibits different transition dynamics from the target MDP $\mathcal{M}_{\text{target}}$, i.e., $\exists (s, a, s') : P_{\text{source}}(s'|s,a) \neq P_{\text{target}}(s'|s,a)$. We assume that μ_{source} is nearly optimal, and $\mathcal{D}_{\text{source}}$ sampled under $\mathcal{M}_{\text{source}}$ is abundant but off-dynamics. Conversely, the target dataset is presumed to have limited and suboptimal trajectories sampled under $\mathcal{M}_{\text{target}}$. Our goal is to enable the transfer of knowledge between the offline datasets $\mathcal{D}_{\text{source}}$ and $\mathcal{D}_{\text{target}}$ to diminish the data dependency of $\mathcal{D}_{\text{target}}$ for learning an optimal policy for $\mathcal{M}_{\text{target}}$.

IV. Approach

We present our novel approach to improve a limited, suboptimal target dataset, $\mathcal{D}_{\text{target}}$, by leveraging a larger, diverse but off-dynamics source dataset, $\mathcal{D}_{\text{source}}$. We achieve this by training a conditional DPM to learn the joint distribution of both datasets. Naively training both datasets would be ineffective, as it would bias the model towards the source dynamics due to the larger size and optimality of $\mathcal{D}_{\text{source}}$. This bias would cause the generated trajectories not to align with the target environment's dynamics.

To address this challenge, the model is conditioned on the following two dynamics-related contexts:

- Continuous dynamics score: This replaces discrete labels (Figure 1) with a "soft" score, allowing for smoother transitions and overlap between trajectories from different datasets.
- Inverse-dynamics context: This measures how closely generated trajectories align with the target environment's dynamics, ensuring the generated samples adhere to its specific constraints.

Although both contexts are related to dynamics information, they serve different purposes. The continuous dynamics score focuses on aligning the source and target dynamics, whereas the inverse dynamics context is used to generate feasible actions based on the target dynamics. Together, these contexts enable the model to learn the joint distribution while adhering the inherent dynamics within the latent space.

A. Dynamics Score Context

We aim to introduce a context that can effectively capture both the differences and similarities between $\mathcal{D}_{\text{source}}$ and $\mathcal{D}_{\text{target}}$ in terms of dynamics information. A straightforward method is assigning the context as a discrete one-hot label. However, this does not provide sufficient information due to the potential overlaps in dynamics between the datasets. For instance, there could exist transitions in both $\mathcal{D}_{\text{source}}$ and $\mathcal{D}_{\text{target}}$ that are identical, indicating a shared dynamic characteristic across both domains. Given this, a discrete labeling system could fail to capture the nuanced differences and shared attributes adequately.

To address this limitation, we propose replacing the discrete context with a continuous score. This score should also be symmetric, ensuring equal representation for trajectories originating from $\mathcal{D}_{\text{source}}$ and $\mathcal{D}_{\text{target}}$. To achieve this, for each trajectory τ in the form of Equation 2, we define the dynamics score as

$$c_{\text{dyn\_score}}(\tau) = \frac{1}{\kappa H}\sum_{t=0}^{H-1}\Big\{\log[P_{\text{target}}(s_t,a_t,s_{t+1})+\varepsilon] \\ - \log[P_{\text{source}}(s_t,a_t,s_{t+1})+\varepsilon]\Big\}. \tag{3}$$

where P_{source} and P_{target} represent the probability of a given transition originating from the source and target datasets respectively, ε is a small value to prevent infinities and κ is a scaling parameter to keep the score within $[-1,1]$. We apply logarithmic scaling to enhance the dynamic range, particularly at the extreme values of the output probabilities. Overall, a smaller score indicates trajectories follow source-like dynamics while a larger score implies target-like dynamics, and a score of zero suggests equal likelihood of both. Details of modelling P_{target} will be elaborated in Section IV-C.

B. Inverse-dynamics Context

When attempting to maximize the out-of-distribution return context through the diffusion sampling process, the generated trajectory may not fully adhere to the dynamics constraints of the underlying MDP. To better enforce these constraints, one approach is to execute an inverse action based on the inverse dynamics to maintain the agent's dynamic constraints, as seen in [66]. Specifically, a separately trained inverse-dynamics model is used to predict the action from two consecutive states that best conform to the underlying MDP. Similar to this approach, we integrate the inverse-dynamics information into our method. The key difference, however, is that instead of using the inverse action as a post-processing method, we directly use it as a context for the conditional model. The rationale is as follows: if the inverse action is computed after the trajectory generation, the consecutive states could already violate the dynamics constraints, and the inverse model cannot recover the correct action. On the other hand, incorporating the inverse dynamics constraints during training compels the trajectory to adhere as closely as possible during the conditional sampling process, thus avoiding violation of the dynamics constraints in the first place. Formally, for each trajectory τ in the form of Equation 2, the inverse context, c_{inverse} is computed as follows:

$$c_{\text{inverse}}(\tau) = \frac{1}{H}\sum_{t=0}^{H-1}\log[1+\|I_{\text{target}}(s_t,s_{t+1})-a_t\|_2], \tag{4}$$

where I_{target} is the target inverse dynamics model, H is the planning horizon. The logarithm function helps to compress the range of the large-valued errors.

C. Practical Algorithm

We begin by outlining the process of learning P_{target}, P_{source} and I_{inverse}. To model P_{target}, we parameterize a binary classifier, $p_\phi = p(s_t,a_t,s_{t+1};\phi)$, with a multi-layer perceptron (MLP). For P_{source}, we simply use $1-p_\phi$. We fit p_ϕ by minimizing the standard binary cross-entropy loss:

$$L(\phi) = -\mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{D}_{\text{source}}\cup\mathcal{D}_{\text{target}}} \\ \left[y\log(p_\phi)+(1-y)\log(1-p_\phi)\right], \tag{5}$$

where $y = 0,1$ represents the source and target labels respectively.

Next, to model the inverse dynamics I_{target}, we parameterize it by ψ with another MLP, and minimize the standard mean-squared error between the predicted action and true action a_t:

$$L(\psi) = \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{D}_{\text{target}}}\left[(I_{\text{target}}(s_t,s_{t+1};\psi)-a_t)^2\right]. \tag{6}$$

Algorithm 1 Training: Off-dynamics Conditional Diffusion Planners

```
Input: Target dataset $\mathcal{D}_{\text{target}}$, Source dataset $\mathcal{D}_{\text{source}}$
Input: Number of training updates $N$,
        Number of diffusion time steps $T$
Initialize dynamics score model $q_\phi$ and inverse model $f_\psi$
Fit $q_\phi$ using Loss in Eq.5 over $\mathcal{D}_{\text{target}} \cup \mathcal{D}_{\text{source}}$
Fit $f_\psi$ using Loss in Eq.6 over $\mathcal{D}_{\text{target}}$
// Start Conditional Diffusion Training
Initialize Conditional U-Nets $\varepsilon_\theta$
for $n$ in $1, 2, ..., N$ do
    Sample stratified batch $\tau_{\mathcal{B}} \in \mathcal{D}_{\text{target}} \cup \mathcal{D}_{\text{source}}$
    Compute context $c_{dyn\_score}(\tau_{\mathcal{B}}) = q_\phi(\tau_{\mathcal{B}})$
    Compute context $c_{inverse}(\tau_{\mathcal{B}}) = f_\psi(\tau_{\mathcal{B}})$
    Set full context $y(\tau_{\mathcal{B}}) = [R(\tau_{\mathcal{B}}), c_{dyn\_score}(\tau_{\mathcal{B}}), c_{\text{inverse}}(\tau_{\mathcal{B}})]$.
    for $t$ in $1, 2, ..., T$ do
        Update $\theta$ with $\varepsilon_\theta(\tau_{\mathcal{B}}, t, y(\tau_{\mathcal{B}}))$ using Loss in Eq.8
    end for
end for
```

Now, we introduce a practical algorithm that combines all components for training a off-dynamics conditional diffusion-based planner. Our model follows the classifier-free approach with input trajectories τ, following Equation 2. The models are conditioned on the full context $y(\tau)$, which consists of the dynamics score, the inverse dynamics and the normalised return $R(\tau) \in [0, 1]$ as follows:

$$y(\tau) = [R(\tau), c_{dyn\_score}(\tau), c_{\text{inverse}}(\tau)]. \tag{7}$$

This context summarizes the optimality and the dynamics information of each trajectory in a continuous form as movitated in Fig. 1. By leveraging dynamics information, the conditional model gains the ability to flexibly learn from both $\mathcal{D}_{\text{source}}$ and $\mathcal{D}_{\text{target}}$ with greater data efficiency. It also enables the conditional model to generate optimal trajectories from either domain in a seamless manner. The objective for the conditional diffusion process is,

$$\max_\theta \mathbb{E}_{\tau \sim (\mathcal{D}_{\text{source}} \cup \mathcal{D}_{\text{target}})} \left[\log p_\theta \left(x_0(\tau) \mid y(\tau)\right)\right]$$

with the loss given by

$$\mathcal{L}(\theta) = \mathbb{E}_{k \sim [1,K], x_0 \sim q, \varepsilon \sim \mathcal{N}(0,I)} \left\| \varepsilon - \varepsilon_\theta(x_k(\tau), k, y(\tau)) \right\|^2 \tag{8}$$

The complete training procedure is detailed in Algorithm 1. During planning, we set the target context $y(\tau) = [1, 1, 0]$ to generate trajectories that maximize the reward for the target environment. This aligns with Equations 3 and 4, where $P_{\text{target}}(s_t, a_t, s_{t+1}) = 1$ and $I_{\text{target}}(s_t, s_{t+1}) = a_t$, ensuring the generated trajectories adhere to the target dynamics.

V. Experiments

We design and conduct comprehensive experiments to thoroughly compare the effectiveness of our method with existing ones. We begin by outlining the experimental setup, which involves creating a diverse and challenging set of off-dynamics datasets for training and evaluation. Nine distinct settings are used to compare our approach with several strong baselines. Additionally, an ablation study is conducted to investigate the specific contribution of each dynamics context to the performance improvement. Finally, we investigate the robustness and capacity of our model to seamlessly interpolate between source and target dynamics.

A. Experimental Setup

Datasets. We formulate our experimental datasets based on our proposed framework, illustrated in Figure 1 (left). We require a large and diverse offline source dataset, and select the Hopper, Walker2d, and Halfcheetah medium-expert datasets from D4RL [59], each consisting of around 2 million samples. For the target dataset, we need limited and sub-optimal samples with different dynamics compared to the source. To achieve this, we first create multiple off-dynamics variants by modifying parameters like mass, size, control range, friction, and gear torques within each environment. Subsequently, we collect 10,000 samples from each modified environment using a behavioral policy trained with Diffuser in the respective source environment. This diverse and challenging off-dynamics setting allows for comprehensive evaluation of our method.

Baselines. We benchmark against various data-driven control algorithms for off-dynamics offline RL. We apply the off-dynamics reward compensation method DARA [13] to several well-performing offline RL algorithms. Among these offline RL algorithms, we include model-free methods like BCQ [60] and CQL [62], as well as diffusion-based methods like Diffuser [33]. In addition to DARA, we include the fine-tuning method, which performs 10,000 additional updates on a pre-trained source model using the target dataset.

Implementation details. For the hyper-parameters of DPM, we follow the default settings in Diffuser [33]. For both the classifier and inverse models, we use basic MLPs with 2 hidden layers and 32 nodes, each trained for 200k updates. We apply *min-max normalization* to the outputs of these models prior to context computations. During training, we apply a context *dropout* of $p = 0.5$ for each context independently. To address the imbalanced dataset, we ensure a 1:1 ratio of source and target data in each batch during sampling. During sampling, we use a conditional sampling weight of $w = 0.9$ for all environments. We evaluate our models over 300 episodes across 3 seeds (total 900 episodes) and compute the normalized score over the target environments [59].

B. Main Results

We conduct experiments on three different environments, each with three variations in their physical properties, resulting in a total of nine different settings. Table I displays the normalized scores of different methods over four training types: 1) source only, 2) target only, 3) source pretraining with target finetuning, and 4) joint source and target training. For the source only setting, which uses a Diffuser trained on optimal samples from the source dataset ("$\mathcal{D}_{\text{source}}$" column), the performance is significantly decreased across all environments when evaluated on the target environment. This finding highlights the inherent challenges presented by our off-dynamics environments. We use this behavioral

Environment	Property	Source	Target	$\mathcal{D}_{\text{source}}$	$\mathcal{D}_{\text{target}}$	$\mathcal{D}_{\text{source}}$ pretrain + $\mathcal{D}_{\text{target}}$ finetune			$\mathcal{D}_{\text{source}} \cup \mathcal{D}_{\text{target}}$ Joint Training		
				Diffuser	Diffuser	Diffuser	CQL	BCQ	CQL+DARA	BCQ+DARA	Proposed
Half-cheetah	total mass	14	7	30.2	30.5	32.6	26.3	28.7	27.3	19.8	**56.8** ± 5.9
	torso size	0.046	0.092	43.6	43.9	38.2	32.0	46.8	45.9	32.2	**54.1** ± 7.9
	control range	[-1,1]	[-0.5,0.5]	41.7	43.1	25.2	52.2	2.2	53.2	53.1	**61.5** ± 6.6
Walker2d	thigh action	Enabled	Disabled	22.6	24.4	39.1	28.8	-0.3	32.2	10.2	**58.5** ± 9.0
	foot gear torque	100	70	42.9	43.1	36.7	46.8	51.0	-0.2	44.2	**63.4** ± 12.2
	foot length	0.1	0.25	42.7	42.9	41.6	48.3	43.6	-0.2	5.3	**63.0** ± 10.7
Hopper	foot friction	2.0	1.7	50.7	51.3	52.5	32.1	40.2	49.7	109.5	**124.2** ± 31.5
	torso stiffness	0	3	73.9	75.8	72.2	65.7	72.5	51.8	42.9	**95.0** ± 6.9
	leg size	0.03	0.04	74.2	76.0	57.3	36.8	82.8	75.6	93.1	**95.9** ± 9.3
			Average	46.9	47.9	43.9	41.0	40.8	37.3	45.6	**74.7**

TABLE I: **Mean normalized scores evaluated on the target environment over 900 episodes across 9 diverse settings.** The columns 'Source' and 'Target' represent the dynamics settings of $\mathcal{D}_{\text{source}}$ and $\mathcal{D}_{\text{target}}$, respectively. Our proposed method is a conditional diffusion planner with contexts according to Equation 7, trained with Algorithm 1.

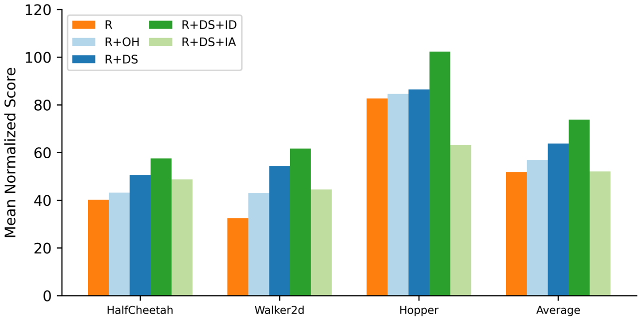

Fig. 2: **Ablation: models trained with different contexts following Algorithm 1.** We report the mean normalized score across different settings per environment. ('R', Orange) represents the base model conditioned on only the return. ('R+OH', LightBlue) adds on the one-hot source/target label as contexts to base model. ('R+DS', Blue) adds on the dynamics score as contexts to base model. ('R+DS+ID', Green) further adds on the inverse-dynamics context. ('R+DS+IA', LightGreen) applies inverse action on 'R+DS'.

policy to collect samples for our target dataset. Next, for the target-only setting, we trained a model with Diffuser using exclusively target samples ("$\mathcal{D}_{\text{target}}$" column). This resulted in a slight improvement over the model trained solely on source data. This score serves as a baseline for the target dataset's performance.

Moving on to finetuning ("$\mathcal{D}_{\text{source}}$ pretrain+$\mathcal{D}_{\text{target}}$ finetune" column), we observe that all three methods (Diffuser, CQL, BCQ) generally perform worse than the source-only Diffuser. A possible reason is that the model struggles to balance learning dynamics from the target dataset while retaining the knowledge of optimal behaviors learned from the source dataset during the finetuning process. Lastly, in the joint training setting ("$\mathcal{D}_{\text{source}} \cup \mathcal{D}_{\text{target}}$ Joint Training" column), both DARA methods face difficulties in this low-data regime despite the dynamics-aware reward adjustment. On the other hand, our proposed conditional DPM with dynamics contexts outperform all baselines by a significant margin. This is attributed to the powerful DPM's ability to generate high-reward trajectories for the target environment, guided by the dynamics-related contexts. Below we break down the contribution of each individual context.

C. Contexts Analysis

To better understand each context's contribution, we conducted additional experiments and present the mean normalized scores per environment in Figure 2.

Source/Target Dynamics We evaluated the impact of using discrete labels ('R+OH', LightBlue) versus our proposed soft score ('R+DS', Blue), as detailed in Section IV-A. While discrete labels showed some improvement over the baseline ('R', Orange), our soft score consistently outperformed both, confirming its superior ability to capture the underlying dynamics structure.

Inverse Dynamics Incorporating the inverse dynamics context ('R+DS+ID', Green) from Section IV-B, in addition to the dynamics score, yielded further improvements. This suggests that the inverse dynamics context provides valuable guidance during sampling, leading to even better adherence to target dynamics. However, directly applying the inverse action on R+DS proved detrimental ('R+DS+IA', LightGreen), underlining our hypothesis that consecutive states generated by R+DS might violate dynamics constraints, hindering the inverse model's ability to recover correct actions. This was particularly pronounced with Hopper, where using the inverse action led to performance worse than relying solely on the return context.

In summary, these ablation studies highlight the importance of both our soft score for dynamics alignment and the inverse dynamics context for generating feasible actions. They also demonstrate the limitations of directly applying inverse actions in the presence of potential constraint violations.

D. Robustness

Inspired by the success of diffusion models in other domains, where they achieve smooth interpolation within the latent space by controlling contexts, we investigate whether our model exhibits similar behaviors. This ability becomes crucial in real-world scenarios, as target dynamics often undergo subtle shifts. To explore this, we leverage one of our experimental settings involving Halfcheetah with varying masses (Table I, first row). The source and target datasets correspond to masses of $m = 14$ and $m = 7$, respectively. Using the models trained with these two datasets, we further

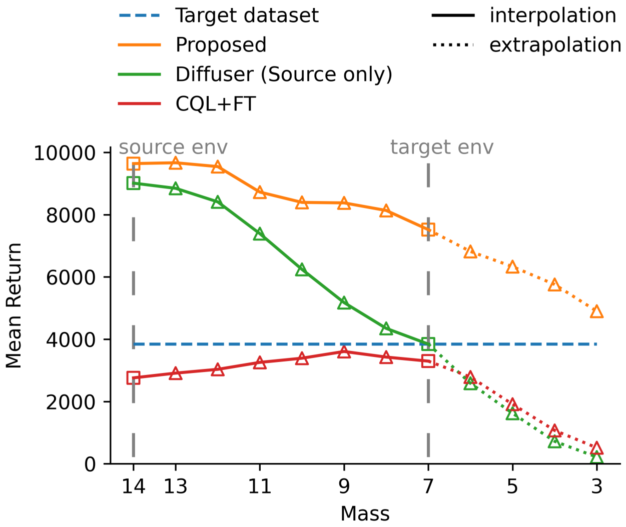

Fig. 3: **Plot of the generalisation capabilities for Halfcheetah.** Models are trained on $\mathcal{D}_{\text{source}}$ $(m = 14)$, $\mathcal{D}_{\text{target}}$ $(m = 7)$. Models are evaluated at interpolated masses $8 \leq m \leq 13$ and extrapolated masses $3 \leq m \leq 6$. Mean returns are shown due to varying normalizing score factors across different masses.

evaluate the masses in between (interpolation: $8 \leq m \leq 13$) and beyond (extrapolation: $3 \leq m \leq 6$). We evaluate our model, alongside two baselines: (1) Diffuser, using only source data; (2) CQL+FT, source pre-train with target fine-tuning. For our approach, we apply a simple linear scale on the dynamics-related contexts based on mass for intermediate evaluations. During extrapolation $3 \leq m \leq 6$, we utilize the context corresponding to the target mass $m = 7$.

Figure 3 presents the results. Our method (orange line) exhibits robust performance across the interpolated range $(8 \leq m \leq 13)$, even without training data following these environment's dynamics. Compared to CQL+FT (red line), the model maintains similar performance for $m = 8, 9$, but exhibits a gradual decline beyond those values. This aligns with the expectation that the fine-tuned model has forgotten most of its source information. While sharing a similar architecture, the Diffuser trained on source data (green line) experiences a rapid performance drop beyond the source mass of $m = 7$. This suggests that the dataset with matching dynamics, despite its small quantity, remains crucial for effectively training our model. Finally, when extrapolating masses outside the range of the training data $(3 \leq m \leq 6)$, all three models exhibit similar limitations in their capabilities. This is evident in the similar rate of performance degradation observed for all models in this region.

VI. Conclusion

In this paper, we propose a new approach, which utilizes a conditional DPM with dynamics-related contexts to address the challenge of data scarcity in offline RL. We introduce a continuous dynamics score and an inverse-dynamics context to effectively capture the underlying dynamics structure within the latent space, enabling the model to learn from both a larger off-dynamics source dataset and a limited, sub-optimal target dataset. Experimental results demonstrate that our method significantly outperforms various baselines. Ablation studies further reveal the critical role of each dynamics context in improving performance. Additionally, our model exhibits promising robustness in handling interpolation scenarios, showcasing its potential for real-world applications with dynamic shifts in the target environment. Our future work aims to expand the applicability of our method by incorporating multiple off-dynamics datasets, enabling adaptation to changes in robot embodiment, and ultimately validating its effectiveness in real-world robotics scenarios.

References

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

[2] S. Lange, T. Gabel, and M. Riedmiller, "Batch reinforcement learning," in *Reinforcement learning: State-of-the-art*. Springer, 2012.

[3] M. Zare, P. Mohsenzadeh Kebria, and A. Khosravi, "Leveraging optimal transport for enhanced offline reinforcement learning in surgical robotic environments," *Abbas, Leveraging Optimal Transport for Enhanced Offline Reinforcement Learning in Surgical Robotic Environments*.

[4] K. Fan, Z. Chen, G. Ferrigno, and E. De Momi, "Learn from safe experience: Safe reinforcement learning for task automation of surgical robot," *IEEE Transactions on Artificial Intelligence*, 2024.

[5] R.-J. Qin, X. Zhang, S. Gao, X.-H. Chen, Z. Li, W. Zhang, and Y. Yu, "Neorl: A near real-world benchmark for offline reinforcement learning," *NeurIPS*, vol. 35, pp. 24 753–24 765, 2022.

[6] N. Lee and J. Moon, "Offline reinforcement learning for automated stock trading," *IEEE Access*, vol. 11, pp. 112 577–112 589, 2023.

[7] H.-H. Tseng, Y. Luo, S. Cui, J.-T. Chien, R. K. Ten Haken, and I. E. Naqa, "Deep reinforcement learning for automated radiation adaptation in lung cancer," *Medical physics*, 2017.

[8] X. Nie, E. Brunskill, and S. Wager, "Learning when-to-treat policies," *Journal of the American Statistical Association*, 2021.

[9] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *2019 ICRA*. IEEE, 2019, pp. 8248–8254.

[10] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *IEEE ICRA*, 2018.

[11] L. Sun, C. Peng, W. Zhan, and M. Tomizuka, "A fast integrated planning and control framework for autonomous driving via imitation learning," in *Dynamic Systems and Control Conference*, vol. 51913. American Society of Mechanical Engineers, 2018, p. V003T37A012.

[12] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Conference on Robot Learning*. PMLR, 2022.

[13] J. Liu, Z. Hongyin, and D. Wang, "Dara: Dynamics-aware reward augmentation in offline reinforcement learning," in *ICLR*, 2021.

[14] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[15] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*. Springer, 2018, pp. 270–279.

[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[17] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf, "Transfer learning in natural language processing," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, 2019, pp. 15–18.

[18] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, “Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection,” *Construction and building materials*, vol. 157, pp. 322–330, 2017.

[19] B. Eysenbach, S. Chaudhari, S. Asawa, S. Levine, and R. Salakhutdinov, “Off-dynamics reinforcement learning: Training for transfer with domain classifiers,” in *ICLR*, 2020.

[20] B. Eysenbach, A. Khazatsky, S. Levine, and R. R. Salakhutdinov, “Mismatched no more: Joint model-policy optimization for model-based rl,” *NeurIPS*, vol. 35, pp. 23 230–23 243, 2022.

[21] H. Niu, Y. Qiu, M. Li, G. Zhou, J. HU, X. Zhan, *et al.*, “When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning,” *NeurIPS*, vol. 35, pp. 36 599–36 612, 2022.

[22] J. Liu, H. Shen, D. Wang, Y. Kang, and Q. Tian, “Unsupervised domain adaptation with dynamics-aware rewards in reinforcement learning,” *NeurIPS*, vol. 34, pp. 28 784–28 797, 2021.

[23] J. Hanna and P. Stone, “Grounded action transformation for robot learning in simulation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[24] H. Karnan, S. Desai, J. P. Hanna, G. Warnell, and P. Stone, “Reinforced grounded action transformation for sim-to-real transfer,” in *2020 IEEE/RSJ IROS*. IEEE, 2020, pp. 4397–4402.

[25] P. F. Christiano, Z. Shah, I. Mordatch, J. Schneider, T. Blackwell, J. Tobin, P. Abbeel, and W. Zaremba, “Transfer from simulation to real world through learning deep inverse dynamics model,” *ArXiv*, vol. abs/1610.03518, 2016.

[26] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *ICML*. PMLR, 2015, pp. 2256–2265.

[27] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *ICLR*, 2021.

[28] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *NeurIPS*, vol. 33, pp. 6840–6851, 2020.

[29] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, *et al.*, “Photorealistic text-to-image diffusion models with deep language understanding,” *NeurIPS*, vol. 35, pp. 36 479–36 494, 2022.

[30] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, “Glide: Towards photorealistic image generation and editing with text-guided diffusion models,” *arXiv preprint arXiv:2112.10741*, 2021.

[31] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “Diffwave: A versatile diffusion model for audio synthesis,” *arXiv preprint arXiv:2009.09761*, 2020.

[32] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, “Video diffusion models,” *arXiv:2204.03458*, 2022.

[33] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, “Planning with diffusion for flexible behavior synthesis,” in *ICML*, 2022.

[34] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, “Is conditional generative modeling all you need for decision making?” in *The Eleventh ICLR*, 2023.

[35] J. Hu, Y. Sun, S. Huang, S. Guo, H. Chen, L. Shen, L. Sun, Y. Chang, and D. Tao, “Instructed diffuser with temporal condition guidance for offline reinforcement learning,” *arXiv preprint arXiv:2306.04875*, 2023.

[36] J. Ho, “Classifier-free diffusion guidance,” *ArXiv*, vol. abs/2207.12598, 2022.

[37] Y. Kang, J. Liu, X. Cao, and D. Wang, “Off-dynamics inverse reinforcement learning from hetero-domain,” *arXiv preprint arXiv:2110.11443*, 2021.

[38] S. Desai, I. Durugkar, H. Karnan, G. Warnell, J. Hanna, and P. Stone, “An imitation from observation approach to transfer learning with dynamics mismatch,” *NeurIPS*, vol. 33, pp. 3917–3929, 2020.

[39] Y. Jiang, T. Zhang, D. Ho, Y. Bai, C. K. Liu, S. Levine, and J. Tan, “Simgan: Hybrid simulator identification for domain adaptation via adversarial reinforcement learning,” in *2021 IEEE ICRA*. IEEE, 2021, pp. 2884–2890.

[40] J. Ho and S. Ermon, “Generative adversarial imitation learning,” *NeurIPS*, vol. 29, 2016.

[41] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *NeurIPS*, vol. 27, 2014.

[42] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *NeurIPS*, vol. 34, pp. 8780–8794, 2021.

[43] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, “Grad-tts: A diffusion probabilistic model for text-to-speech,” in *ICML*. PMLR, 2021, pp. 8599–8608.

[44] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *ICML*. PMLR, 2017, pp. 214–223.

[45] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” *NeurIPS*, vol. 30, 2017.

[46] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, “Cascaded diffusion models for high fidelity image generation,” *The Journal of Machine Learning Research*, vol. 23, pp. 2249–2281, 2022.

[47] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, “Repaint: Inpainting using denoising diffusion probabilistic models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 461–11 471.

[48] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.

[49] M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon, and J. Tang, “Geodiff: A geometric diffusion model for molecular conformation generation,” in *ICLR*, 2021.

[50] Z. Lyu, Z. Kong, X. Xudong, L. Pan, and D. Lin, “A conditional point diffusion-refinement paradigm for 3d point cloud completion,” in *ICLR*, 2021.

[51] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, “Decision transformer: Reinforcement learning via sequence modeling,” *NeurIPS*, vol. 34, pp. 15 084–15 097, 2021.

[52] Z. Zhu, M. Liu, L. Mao, B. Kang, M. Xu, Y. Yu, S. Ermon, and W. Zhang, “Madiff: Offline multi-agent learning with diffusion models,” *arXiv preprint arXiv:2305.17330*, 2023.

[53] F. Ni, J. Hao, Y. Mu, Y. Yuan, Y. Zheng, B. Wang, and Z. Liang, “Metadiffuser: Diffusion model as conditional planner for offline meta-rl,” *arXiv preprint arXiv:2305.19923*, 2023.

[54] H. He, C. Bai, K. Xu, Z. Yang, W. Zhang, D. Wang, B. Zhao, and X. Li, “Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning,” *arXiv preprint arXiv:2305.18459*, 2023.

[55] Y. Zheng, J. Li, D. Yu, Y. Yang, S. E. Li, X. Zhan, and J. Liu, “Safe offline reinforcement learning with feasibility-guided diffusion model,” *arXiv preprint arXiv:2401.10700*, 2024.

[56] Z. Dong, J. Hao, Y. Yuan, F. Ni, Y. Wang, P. Li, and Y. Zheng, “Diffuserlite: Towards real-time diffusion planning,” *arXiv preprint arXiv:2401.15443*, 2024.

[57] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *Robotics: Science and Systems (RSS)*, 2023.

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

[59] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, “D4rl: Datasets for deep data-driven reinforcement learning,” *arXiv preprint arXiv:2004.07219*, 2020.

[60] S. Fujimoto, D. Meger, and D. Precup, “Off-policy deep reinforcement learning without exploration,” in *ICML*. PMLR, 2019, pp. 2052–2062.

[61] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, “Stabilizing off-policy q-learning via bootstrapping error reduction,” *NeurIPS*, vol. 32, 2019.

[62] A. Kumar, A. Zhou, G. Tucker, and S. Levine, “Conservative q-learning for offline reinforcement learning,” *NeurIPS*, vol. 33, pp. 1179–1191, 2020.

[63] I. Kostrikov, A. Nair, and S. Levine, “Offline reinforcement learning with implicit q-learning,” in *ICLR*, 2021.

[64] M. Janner, Q. Li, and S. Levine, “Offline reinforcement learning as one big sequence modeling problem,” *NeurIPS*, vol. 34, pp. 1273–1286, 2021.

[65] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[66] Z. Liang, Y. Mu, M. Ding, F. Ni, M. Tomizuka, and P. Luo, “Adaptdiffuser: Diffusion models as adaptive self-evolving planners,” in *ICML*, 2023.