ReDiffuser: Reliable Decision-Making Using a Diffuser with Confidence Estimation

Nantian He [1] Shaohui Li [1] Zhi Li [1] Yu Liu [2] You He [2]

Abstract

The diffusion model has demonstrated impressive performance in offline reinforcement learning. However, non-deterministic sampling in diffusion models can lead to unstable performance. Furthermore, the lack of confidence measurements makes it difficult to evaluate the reliability and trustworthiness of the sampled decisions. To address these issues, we present ReDiffuser, which utilizes confidence estimation to ensure reliable decision-making. We achieve this by learning a confidence function based on Random Network Distillation. The confidence function measures the reliability of sampled decisions and contributes to quantitative recognition of reliable decisions. Additionally, we integrate the confidence function into task-specific sampling procedures to realize adaptive-horizon planning and value-embedded planning. Experiments show that the proposed ReDiffuser achieves state-of-the-art performance on standard offline RL datasets.

1. Introduction

Offline reinforcement learning (RL) methods learn policies from previously collected data instead of interacting with the environment. Traditional offline RL methods suffer from the *deadly triad*, *limited data*, and *reward sparsity* problems. Instead, recent work formulates offline RL as a sequence modeling problem and achieves superior performance on various tasks. In particular, diffusion-based offline RL methods (Janner et al., 2022; Ajay et al., 2023; Wang et al., 2022; Liang et al., 2023; Ni et al., 2023; Zhao & Grover, 2023; Li et al., 2023; Chen et al., 2024) show impressive improvement by exploiting the powerful modeling ability of diffusion models. In these works, the diffusion models generate state and action trajectories through iterative denoising.

The diffusion models originally designed for image and video generation are highly non-deterministic. The iterative denoising process is decided by dozens of random samples from noisy distributions. Thus, the generative model provides different images and videos for the given prompts. However, in offline decision making, the non-determinism results in significant variations in the decisions if there is a slight change in any of the sampling steps. Thus, the decision policy can be significantly influenced by the random seed and the computing platform, which can be an issue in critical scenarios such as robotic manipulation and medical surgery.

The reliability of the diffusion model has recently been studied in image generation. These methods (Angelopoulos et al., 2022; Horwitz & Hoshen, 2022; Teneggi et al., 2023) offer the lower and upper bounds for each generated pixel, and the ground truth is guaranteed to fall within these bounds with a given probability. However, these methods cannot be directly applied in offline RL since the previously collected data may be suboptimal and lack the ground truth.

In this paper, we present ReDiffuser, a diffusion-based offline reinforcement learning method that achieves reliable decision-making by leveraging Random Network Distillation (RND) (Burda et al., 2019). An RND-based confidence function is learned from the historical trajectories of pretrained Diffuser (Janner et al., 2022), which evaluates the similarity between the current decision and statistically reliable decisions. Specifically, we propose task-specific methods for gathering historical trajectories on goal-conditioned and reward-maximization tasks. For goal-conditioned tasks, we collect historical trajectories with diverse horizons to accommodate variable-length planning. Thus, the confidence measurement can adaptively choose the decisions with appropriate horizons. For reward-maximization tasks, historical trajectories are selected from the high-value trajectories. Therefore, we can achieve value-embedded planning that selects decisions with low risks and high values. Evaluation of the standard benchmark D4RL (Fu et al., 2020) shows that the proposed ReDiffuser outperforms Diffuser

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China [2] Department of Electronics, Tsinghua University, Beijing, China. Correspondence to: Yu Liu <liuyu77360132@126.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

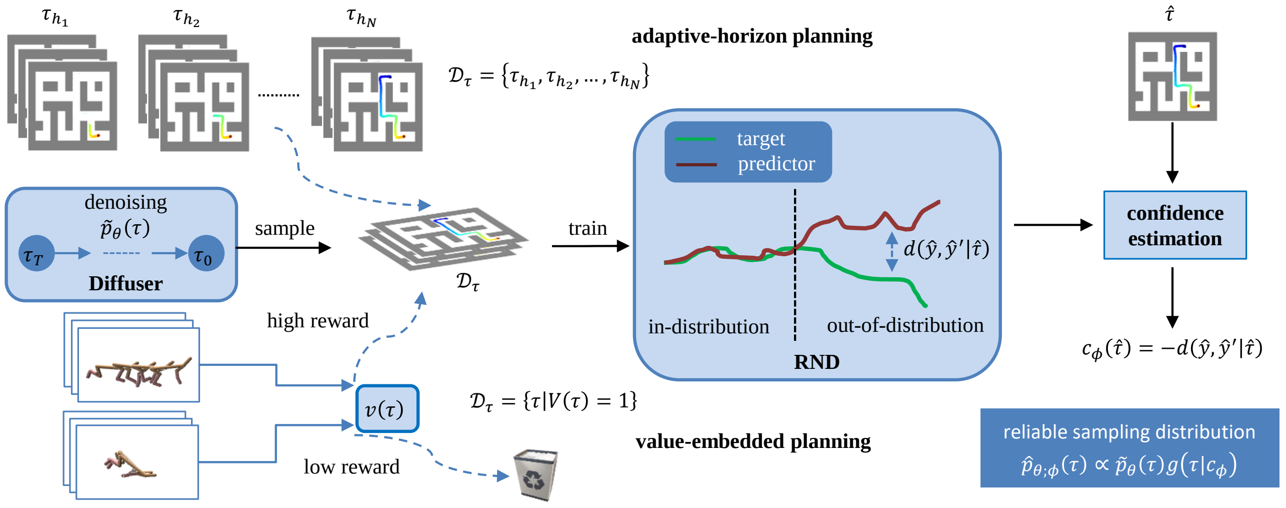

τ_{h_1}
τ_{h_2}
....
τ_{h_N}
adaptive-horizon planning
$\mathcal{D}_\tau = \{\tau_{h_1}, \tau_{h_2}, \dots, \tau_{h_N}\}$
$\hat{\tau}$
target
predictor
denoising
$\tilde{p}_\theta(\tau)$
τ_T
τ_0
Diffuser
sample
$\mathcal{D}_\tau$
train
$d(\hat{y}, \hat{y}'|\hat{\tau})$
in-distribution
out-of-distribution
RND
confidence estimation
$c_\phi(\hat{\tau}) = -d(\hat{y}, \hat{y}'|\hat{\tau})$
high reward
$v(\tau)$
$\mathcal{D}_\tau = \{\tau|V(\tau) = 1\}$
low reward
value-embedded planning
reliable sampling distribution
$\hat{p}_{\theta;\phi}(\tau) \propto \tilde{p}_\theta(\tau) g(\tau|c_\phi)$

Figure 1. The overall framework of ReDiffuser. The RND-based confidence function captures the statistical characteristics in historical decisions, and provides confidence scores based on the distance between the outputs of the target and predictor networks. Based on the naive confidence estimation, adaptive-horizon planning and value-embedded planning are realized by adjusting the sampling process of historical decisions to train RND.

and achieves 18. 2% improvement in Maze2D.

The contributions of our work are as follows:

- We leverage RND to learn a confidence function measuring the reliability of the non-deterministic decisions of diffusion-based offline RL.
- We propose adaptive-horizon planning and value-embedded planning for goal-conditioned tasks and reward-maximization tasks on the foundations of confidence estimation.
- We conduct extensive experiments to evaluate the effectiveness of the proposed confidence estimation and ReDiffuser. [1].

2. Preliminary

2.1. Diffusion Probabilistic Model

Diffusion probabilistic models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are designed to learn the complicated distribution lied behind high-dimensional signals such as images and videos, achieved through novel diffusion and denoising processes. The diffusion process progressively disturbs the raw signal, until it becomes indistinguishable from Gaussian noise. Conversely, the denoising process progressively removes additive noise to recover the raw signal. Theoretically, each step of the denoising process follows a Gaussian distribution, as shown in Equation (1).

$$p_\varphi(\tau_{i-1} \mid \tau_i) = \mathcal{N}(\tau_{i-1}; \mu_\varphi(\tau_i, i), \Sigma_\varphi(\tau_i, i)), \qquad (1)$$

where τ_{i-1} is denoised from τ_i, $\tau = \tau_0$ represents the noiseless raw signal, and φ represents the learnable parameters of the parameterized Gaussian distribution.

2.2. Diffusion-based Offline Reinforcement Learning

Offline reinforcement learning targets on learning powerful policies from the offline trajectory datasets collected by a demonstrative behavior policy. The objective of an offline RL algorithm is to learn a policy π_θ^* that maximizes the cumulative reward when interacting with the environment, as shown in Equation (2).

$$\pi_\theta^* = \arg\max_{\pi_\theta} \sum_{t=0}^{T-1} \mathbb{E}_{a_t \sim \pi_\theta(\cdot|s_t)}[r(s_t, a_t)], \qquad (2)$$

where t is the time step, an episode contains T time steps, s_t and a_t are the state and action of t-th time step, and $r(\cdot)$ is the reward. The main challenge of offline RL is the distributional shift (Levine et al., 2020; Prudencio et al., 2023), where the novel states and actions outside the dataset are with inaccessible rewards. Since the exploration on novel states is limited, it is inevitable that the learned policies show unwanted actions on unseen states. A common solution is to balance between constraining the learned policy π_θ close to the behavior policy π_β and maximizing the cumulative reward.

[1] Source code is available at https://github.com/he-nantian/ReDiffuser

Diffusion probabilistic models are employed in offline RL due to their advantages in imitation learning and guided sampling process (Janner et al., 2022). *1) Imitation learning:* Diffusion probabilistic models have powerful ability to learn complicated distributions. Thus, they can exploit the distribution of the behavior policy π_β from the collected dataset. *2) Conditional sampling:* The sampling process can be easily disturbed by injecting a task-specific guidance into the progressive denoising process. Therefore, the learned policy $\tilde{p}_\theta(\tau)$ can be represented by a task-guided distribution, as shown in Equation (3).

$$\tilde{p}_\theta(\tau) \propto p_\theta(\tau)h(\tau), \tag{3}$$

where $p_\theta(\tau)$ is trained from the offline dataset to imitate the behavior policy, and $h(\tau)$ contains the preference of the current task. In goal-conditioned tasks, the guidance $h(\tau)$ is inpainting that completes the trajectories from the start to the end in each denoising step. In reward-maximization tasks, the guidance is a classifier that guides the sampling process toward high-value regions by adjusting the means μ of each denoising step. At inference time, a decision $\hat{\tau}$ is sampled from the task-guided distribution to interact with the environment, *i.e.*, $\hat{\tau} \sim \tilde{p}_\theta(\tau)$. However, diffusion probabilistic models still suffer from distributional shift, where unsafe actions are inevitably for unseen states. The decisions sampled from $\tilde{p}_\theta(\tau)$ are with no reliability guarantees. Therefore, it is necessary to employ the confidence estimation on these decisions, which is realized by random network distillation in this work.

2.3. Random Network Distillation

Random network distillation (RND) offers a method to recognize samples within and outside the distribution of training set (Burda et al., 2019). RND employs two randomly initialized networks, of which the parameters are denoted by ψ and ϕ for convenience. Thus, the two networks realize mappings $p(y|\tau;\phi)$ and $p(y'|\tau;\psi)$ between the input and output spaces. Without loss of generality, we refer TO $p(y'|\tau;\psi)$ as the target network, and $p(y|\tau;\phi)$ as the predictor network. Given a training set $\mathcal{D}_\tau$, RND minimizes the distance between $p(y|\tau;\phi)$ and $p(y'|\tau;\psi)$ over $\mathcal{D}_\tau$ by optimizing the predictor network. Formally, the optimal predictor could be obtained as FOLLOWING.

$$\phi^* = \arg\min_\phi D(p(y \mid \tau;\phi) \parallel p(y' \mid \tau;\psi)), \text{ with } \tau \sim \mathcal{D}_\tau, \tag{4}$$

where $D(\cdot \parallel \cdot)$ represents the distance between two probability distributions. On the one hand, the two networks would output close results on in-distribution samples due to the optimization process presented in Equation (4). On the other hand, the two networks would generate distinguishable results (with extremely high probabilities) on out-of-distribution samples because the target and predictor networks are initialized independently.

3. Method

3.1. Overview

Diffusion-based offline RL is expected to make stable decisions under all circumstances. However, the task-guided distribution that represents the policy brings about issues of reliability and trustworthiness. Furthermore, there is a lack of measurements to evaluate the decisions sampled from the black-box distribution.

In this paper, we introduce ReDiffuser, a method for improving the reliability of decision-making in diffusion-based RL based on confidence estimation. Figure 1 illustrates the framework of ReDiffuser. We employ a confidence function $c_\phi(\tau)$ to estimate the confidence of the sampled decision τ. Based on the confidence function, we build a confidence guidance $g(\tau \mid c_\phi)$ to tune the task-guided distribution for a more reliable sampling procedure, as shown in Equation (5).

$$\tau \sim \hat{p}_{\theta;\phi}(\tau) \propto \tilde{p}_\theta(\tau)g(\tau \mid c_\phi), \tag{5}$$

where $\tilde{p}_\theta(\tau)$ is the task-guided distribution that can be found in Equation (3), and the confidence guidance function $g(\tau \mid c_\phi)$ is derived from the confidence function c_ϕ. The task-guided distribution $\tilde{p}_\theta(\tau)$ is disturbed by the confidence guidance $g(\tau \mid c_\phi)$, increasing the possibility of selecting sampled decisions with higher confidence. In Section 3.2, we introduce the proposed RND-based confidence function $c_\phi(\tau)$. In Section 3.3, we elaborate the exact forms of the confidence guidance $g(\tau \mid c_\phi)$ and the implementations towards two kinds of decision-making tasks (*i.e.*, adaptive-horizon planning and value-embedded planning).

3.2. RND for Confidence Estimation

We extend of the application of RND (Burda et al., 2019) for confidence estimation in diffusion-based offline RL, where reliability and trustworthiness are key considerations. Motivated by the imitation learning nature of offline RL methods, we measure the reliability of the sampled decision by the similarity between the historical decisions and it. We use RND to measure the similarity, where historical decisions are employed as the training set of the predictor network.

The training set of RND, denoted by $\mathcal{D}_\tau$, consists of the historical decisions generated by a pretrained Diffuser performed on tasks contained the offline datasets. For example, we use the initial states of the locomotion datasets, and use Diffuser to generate candidate decisions. Given a decision $\hat{\tau}$ as input, the predictor network and the target network each output an M-dimensional vector. We denote these vectors as y and y', respectively. The output of RND is the distance between y and y', which is measured by the Euclidean distance in this work. Thus, the following equation holds.

$$\begin{aligned} D(p(y \mid \tau;\phi) \parallel p(y' \mid \tau;\psi)) &= \mathbb{E}_{\tau\sim\mathcal{D}_\tau} d(y, y'|\tau) \\ &= \mathbb{E}_{\tau\sim\mathcal{D}_\tau} \|y - y'\|_2^2\,, \end{aligned} \tag{6}$$

Therefore, after training, RND can distinguish those decisions that are similar to the historical decisions (*i.e.*, in-distribution) from others (*i.e.*, out-of-distribution).

During test, we feed the sampled decision $\hat{\tau}$ to the trained RND. The output of RND, *i.e.*, the distance between the outputs of the predictor and target networks, is used to measure whether the sampled decision is an in-distribution decision. In short, the decision with a value close to zero is considered to be an in-distribution decision, and that with a large value is considered to be an out-of-distribution decision. The confidence function can be defined as the negative of RND output, as shown in Equation (7).

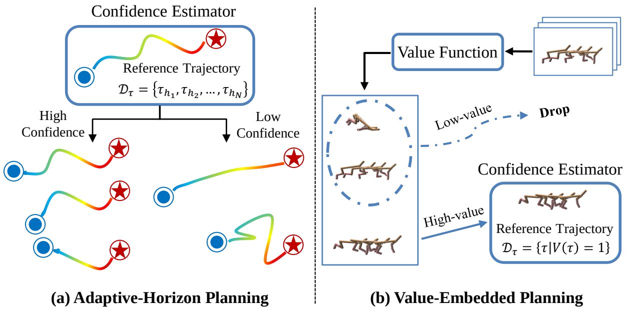
$$c_\phi(\hat{\tau}) = -d(\hat{y}, \hat{y}'|\hat{\tau}), \tag{7}$$

where $\hat{y}$ and $\hat{y}'$ denote the outputs of the predictor network and the target network, respectively. Thus the confidence function $c_\phi(\hat{\tau}) \in (-\infty, 0]$ is positively correlated with the similarity between the current decision $\hat{\tau}$ and historical decisions $\tau \in \mathcal{D}_\tau$. To be brief, we refer to the output of the confidence function as confidence score and denote it as c_ϕ.

Since Diffuser produces the entire trajectories as the decisions, we specify the target and predictor networks as follows. The target network contains one-dimensional convolutional layers and several fully connected layers, which efficiently extracts temporal features efficiently. The predictor network uses the target network as a backbone, and adopts 1 or 2 additional fully connected layers on the top of the backbone. The design of the predictor network guarantees that the predictor network is able to efficiently fit the target network to the training data. Since the target and predictor networks are much smaller than Diffuser, the inference time for confidence estimation is negligible, allowing real-time evaluation for decisions when interacting with the environment.

3.3. Confidence Guidance for Reliable Decision-Making

In this section, we introduce the forms of the confidence guidance $g(\tau \mid c_\phi)$ for different kinds of tasks, which are realized based on the confidence function $c_\phi(\tau)$ in Equation (7). In goal-conditioned tasks, we propose adaptive-horizon planning to enable more flexible planning based on the confidence scores. In reward-maximization tasks, we propose value-embedded planning, which integrates the value function $v(\tau)$ into $c_\phi(\tau)$ to select reliable and valuable decision. These two forms of the confidence guidance are illustrated in Figure 2.



(a) Adaptive-Horizon Planning

(b) Value-Embedded Planning

Figure 2. **(a) Adaptive-Horizon Planning.** Synthesis planned trajectories with diverse horizons are generated from Diffuser's samples. These trajectories are all used as reference trajectories for the training procedure of RND, allowing for adaptive horizon selection. Selecting a short horizon when the target point is distant or a long horizon when the target point is nearby will result in low confidence due to the lack of occurrences in the training samples. **(b) Value-Embedded Planning.** We embed the value function into the confidence function by retaining only the high-value planned trajectories as reference trajectories. At inference time, a planned trajectory with both high-reward and low-risk is evaluated with high confidence.

3.3.1. Adaptive-Horizon Planning

In goal-conditioned tasks, Diffuser outputs trajectories from the initial state to the goal state as decisions. The horizon of the trajectory is determined by a hyperparameter, which is fixed during inference. A shorter horizon always means either reaching the goal state earlier to get more reward, or being too challenging to complete the task and even violating the dynamics. In practice, Diffuser is trained with a relatively long horizon to guarantee the reachability (Janner et al., 2022; Liang et al., 2023). However, the task will be better accomplished if shorter horizons are available for selection at inference time, when the goal state is not far from the initial state. To achieve this, it is crucial to determine the horizon adaptively, to enable shorter horizons for higher rewards and longer horizons for reachability.

We propose adaptive-horizon planning that can adaptively choose an appropriate horizon based on the distance between the initial and target states. To realize it, we first collect a training set $\mathcal{D}_\tau$ with diverse horizons. For simplicity, we denote a decision with a horizon length h as τ_h. We specify the a group of horizons $\{h_1, h_2, \ldots, h_N\}$, and collect the output trajectories with these horizons as $\mathcal{D}_\tau$. Thus,

$$\mathcal{D}_\tau = \{\tau_{h_1}, \tau_{h_2}, \ldots, \tau_{h_N}\}, \tag{8}$$

where $\tau_{h_i} \sim \tilde{p}_\theta(\tau_{h_i})$ is sampled from the original sampling distribution of the pretrained Diffuser. After training RND using $\mathcal{D}_\tau$, the confidence score can be used to measure the reliability of planned trajectories with diverse horizons. In practice, we truncate trajectories with shorter horizons directly from those with the longest horizon in the training set. We achieve the truncation by setting the midpoint of these trajectories as the new initial state, and keeping the trajectory towards the goal state. Thus, the time for collecting historical trajectories could be reduced. Moreover, all the truncated trajectories have guaranteed reachability.

Then the confidence guidance for adaptive-horizon planning can be formulated in Equation (9).

$$g(\tau_h \mid c_\phi) = e^{\frac{\beta}{\alpha(h)} \cdot c_\phi(\tau_h)}, \tag{9}$$

where $c_\phi(\tau_h)$ represents the estimated confidence score of τ_h, the constant $\beta \in [0, +\infty)$ is referred to as confidence proportion, and function $\alpha(h) \in (0, 1]$ is confidence discount. In experiments, we use $\beta = 20$, and $\alpha(h) = \left(\frac{h}{h_N}\right)^p$ with $p \in \{0.2, 0.4, 0.6\}$. The confidence proportion β determines the extent to which confidence guides the sampling distribution. The confidence discount increases with respect to h and improves the sampling probability of decisions with short horizons. The necessity and validity of these two parameters are further demonstrated in the ablation study in Section 4.4. In general, ReDiffuser chooses a reliable trajectory with suitable horizon using the confidence guidance presented in Equation (9).

3.3.2. Value-Embedded Planning

In reward-maximization tasks, there exists a value function $v(\tau)$ that estimates an action's cumulative reward for classifier-guided sampling (Nichol et al., 2022; Nichol & Dhariwal, 2021). However, the value of the unseen actions may be overestimated during extrapolation, which is also the motivation of classical BCQ method (Fujimoto et al., 2019). To alleviate the notorious overestimation problem, we integrate $v(\tau)$ into the confidence function $c_\phi(\tau)$ by collecting high-value data to train RND. In other words,

$$\mathcal{D}_\tau = \{\tau \mid V(\tau) = 1\}, \tag{10}$$

where $V(\tau)$ is binary with 1 indicating a high value and 0 indicating a low value for a decision τ. In practice, we recognize the decision with the highest value $v(\tau)$ in one sampling as the high-value decision. After trained on $\mathcal{D}_\tau$, RND can capture the statistical characteristics of historical decisions with high value. Therefore, decisions that are incorrectly assigned with high values are excluded from the selection due to their distinguishable features compared to reliable ones.

Thus, the form of confidence guidance for value-embedded planning is defined in Equation (11).

$$g(\tau_v \mid c_\phi) = e^{\beta \cdot c_\phi(\tau_v)}, \tag{11}$$

where β is the confidence proportion, as defined previously.

Table 1. **KUKA Block Stacking.** We evaluate RND-based confidence estimation in the KUKA block stacking task. We report the mean and bias of success rate over 10 random seeds and each seed corresponds to 1000 planning episodes.

Environment	BCQ	CQL	Diffuser	ReDiffuser
KUKA	0.0	24.4	60.5 ±1.1	**67.0** ±1.5
Enhancement	_	_	_	**6.5**

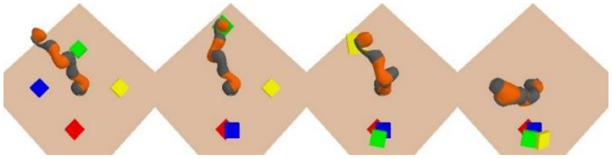


Figure 3. Visualization of KUKA block stacking task. It has three *pick-and-place* operations during an episode to stack the four blocks together.

4. Experiments

We construct our experiments to evaluate the proposed confidence estimation, adaptive-horizon planning, and value-embedded planning described in Section 3. First, we evaluate the RND-based confidence estimation method in robotic manipulation tasks. Second, we evaluate adaptive-horizon planning in maze navigation tasks. Third, we evaluate value-embedded planning in robot locomotion tasks. Finally, we design two ablation studies to further elaborate on the effectiveness of our methodology. The experimental results reported in this section are normalized cumulative rewards on the corresponding tasks.

4.1. RND-based Confidence Estimation

We conduct experiments using Diffuser with and without confidence estimation on KUKA block stacking task (Schreiber et al., 2010) to evaluate the effectiveness of confidence estimation. The goal of this task is to stack four blocks as tall as possible with three *pick-and-place* operations in an episode. Ideally, the robot arm should stack the four blocks together to get the sparse reward with the normalized score of 100. The offline dataset is generated by an expert policy PDDLStream (Garrett et al., 2020). The visualization of this task can be seen in Figure 3.

In KUKA, the distance between the initial state and the expected final state is relatively fixed during an episode, since the robot arm is always reset vertically upward and the four blocks are placed on the ground at the beginning of each episode. Additionally, value function is not separately

Table 2. **Maze2D.** The performance of ReDiffuser, Diffuser and several model-free RL algorithms in the Maze2D environment. We report the mean and bias of scores over 1∼10 random seeds and each seed corresponds to 1000 planning episodes. We have bolded the maximum score per task.

Environment	MPPI	CQL	IQL	Diffuser	ReDiffuser
Single-Umaze	33.2	5.7	47.4	121.8±0.6	**145.3** ±1.2
Single-Medium	10.2	5.0	34.9	130.7±0.8	**140.3** ±1.8
Single-Large	5.1	12.5	58.6	130.0±1.7	**159.6** ±3.0
Average	16.2	7.7	47.0	127.5	**148.4**
Multi-Umaze	41.2	_	24.8	129.2 ±1.3	**154.8** ±1.8
Multi-Medium	15.4	_	12.1	128.7 ±1.4	**149.4** ±2.3
Multi-Large	8.0	_	13.9	142.3 ±1.6	**176.1** ±3.4
Average	21.5	_	16.9	133.4	**160.1**

learned since the Diffuser is only used to clone the behavior policy on an offline demonstration dataset. Therefore, we use naive RND-based confidence estimation instead of adaptive-horizon planning or value-embedded planning.

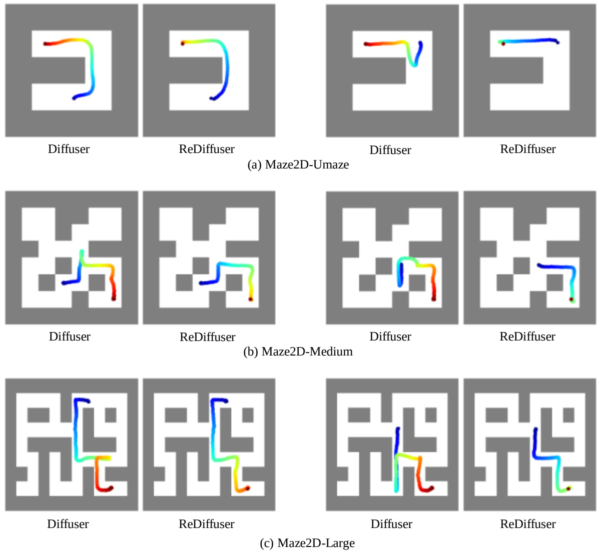
As shown in Table 1, ReDiffuser can increase the normalized score by 6.5 compared with Diffuser, indicating the effectiveness of the naive RND-based confidence estimation. BCQ (Fujimoto et al., 2019) and CQL (Kumar et al., 2020) have poor performances for lack of behavior cloning ability.

4.2. Goal-conditioned Reliable Planning

We evaluate adaptive-horizon planning on Maze2D tasks (Fu et al., 2020). The mazes used in these tasks are categorized into three types: Umaze, Medium, and Large, ordered from easiest to most difficult. Besides, there exist two kinds of task settings, *i.e.*, single and multiple task, are conducted in these mazes. In the single-task setting, the initial location is reset, and the target location is fixed at the beginning of each episode. In the multi-task setting, both the initial and target locations are reset at the beginning of each episode. The goal of the agent is to navigate from the initial location to the target location as quickly as possible. A score of 100 corresponds to the performance of a reference expert policy.

To implement adaptive-horizon planning, we use historic trajectories with different horizons, all of which are multiples of 32. The longest horizons in the training set are 128, 256, and 384 for Umaze, Medium, and Large, respectively. Additionally, the output dimensions of the predictor networks are 240, 336, and 384 for Umaze, Medium, and Large, respectively.

As shown in Table 2, ReDiffuser significantly outperforms Diffuser and other existing model-free RL algorithms (Williams et al., 2015; Kumar et al., 2020; Kostrikov et al., 2021). The improvement originates from confidence estimation over adaptive-horizon trajectories. Thus, ReDiffuser can choose the trajectory with a suitable horizon.



(a) Maze2D-Umaze

(b) Maze2D-Medium

(c) Maze2D-Large

Figure 4. Visualization of planned trajectories in Maze2D tasks by Diffuser and ReDiffuser. The trajectories are drawn based on the ground-truth navigation point when interacting with the environment online. The agent starts from the blue point and navigates to the red point.

Figure 4 illustrates the trajectories planned by Diffuser and ReDiffuser, which shows that adaptive-horizon planning can avoid the “wandering” phenomenon and renders smoother trajectories to achieve the goal. In addition, we compare our method with other diffusion-based offline RL algorithms that are reported comparative in Maze2D tasks, which can be found in Appendix A. Generally, our method achieves the state-of-the-art performance in Maze2D tasks.

4.3. Reward-maximization Reliable Planning

We evaluate value-embedded planning in locomotion tasks based on the MuJoCo engine (Todorov et al., 2012), which is commonly adopted in evaluating other offline RL algorithms. The goal of these tasks is to control a robot to move forward without falling down. Each task corresponds to three different levels of behavior policies to generate the offline dataset: Medium-Expert, Medium and Medium-Replay.

In locomotion tasks, the states of a robot include positions and orientations of the base link and joints. At the start of each episode, we reset the state randomly. To collect the high-value training data for RND to achieve value-embedded planning, we sample 64 decisions for each randomly reset condition and only retain the best one with the highest value. We find that the performance of Diffuser on

Table 3. **Locomotion.** Performance in the MuJoCo environment on the D4RL locomotion tasks. We report the mean and bias of the scores over 1∼10 random seeds and each seed corresponds to 10 planning episodes. We have bolded those scores that are above 95 percent of the maximum per task.

Dataset	**Environment**	**BC**	**CQL**	**IQL**	**DT**	**TT**	**MOPO**	**MOReL**	**MBOP**	**Diffuser**	**ReDiffuser**
Medium-Expert	HalfCheetah	55.2	91.6	86.7	86.8	95.0	63.3	53.3	**105.9**	86.9 ±5.2	88.3 ±3.1
Medium-Expert	Hopper	52.5	**105.4**	91.5	**107.6**	**110.0**	23.7	**108.7**	55.1	102.1 ±12.0	**104.8** ±9.4
Medium-Expert	Walker2d	**107.5**	**108.8**	**109.6**	**108.1**	101.9	44.6	95.6	70.2	**104.9** ±4.8	**107.4** ±1.4
Medium	HalfCheetah	42.6	44.0	**47.4**	42.6	**46.9**	42.3	42.1	44.6	**45.2** ±0.5	**45.2** ±0.7
Medium	Hopper	52.9	58.5	66.3	67.6	61.1	28.0	**95.4**	48.8	92.9 ±12.9	**100.2** ±6.0
Medium	Walker2d	75.3	72.5	78.3	74.0	**79.0**	17.8	77.8	41.0	**80.0** ±1.7	**82.5** ±2.3
Medium-Replay	HalfCheetah	36.6	45.5	44.2	36.6	41.9	**53.1**	40.2	42.3	36.6 ±2.8	38.6 ±1.7
Medium-Replay	Hopper	18.1	**95.0**	**94.7**	82.7	**91.5**	67.5	**93.6**	12.4	**93.2** ±1.4	**94.0** ±1.7
Medium-Replay	Walker2d	26.0	77.2	73.9	66.6	**82.6**	39.0	49.8	9.7	**80.6** ±13.2	**82.2** ±10.7
Average		51.9	77.6	77.0	74.7	**78.9**	42.1	72.9	47.8	**80.3**	**82.6**

Table 4. Ablation study of the extent to confidence guidance in the KUKA task and the Maze2D Multi-Large task. We have bolded the maximum score per task.

Environment	$\boldsymbol{\beta = 0.0}$	$\boldsymbol{\beta = 1.0}$	$\boldsymbol{\beta = 2.0}$	$\boldsymbol{\beta = 4.0}$	$\boldsymbol{\beta = 10.0}$	$\boldsymbol{\beta = 20.0}$
KUKA	60.5 ±1.1	62.1 ±0.6	62.9 ±0.9	64.1 ±1.3	65.8 ±1.3	**67.0** ±1.5
Multi-Large	142.3 ±1.6	171.9 ±1.0	172.3 ±1.2	173.0 ±1.4	173.9 ±1.7	**174.9** ±2.1

Hopper-Medium-Replay can be improved by adjusting the hyperparameter *scale*. Thus, we use a *scale* of 1.0, instead of 0.1 in original paper of Diffuser (Janner et al., 2022).

As shown in Table 3, ReDiffuser exhibits consistent performance improvements over Diffuser. Besides, ReDiffuser is either superior or competitive to a variety of existing algorithms across all locomotion settings, and its average score is ranked first. The comparison with other diffusion-based offline RL methods in locomotion tasks can be found in Appendix A.

4.4. Ablation Study

We design two ablation experiments to further demonstrate the effectiveness of ReDiffuser. First, we prove the essence of confidence guidance by changing the value of the confidence proportion constant β in Equation (9) and Equation (11). Second, we prove the necessity of confidence discount applied in adaptive-horizon planning by adjusting the power constant p of the confidence discount function $\alpha(h)$ in Equation (9).

4.4.1. Confidence Proportion

As shown in Equation (9), the confidence proportion β determines the extent to which confidence scores affect the task. Specifically, it governs how much the confidence function influences the sampling distribution. We conduct an ablation experiment regarding the confidence proportion employed in the KUKA and Maze2D Multi-Large tasks. When $\beta = 0$, the sampling process is free from confidence guidance, which performs as Diffuser. The results in Table 4 demonstrate that scores steadily increase as the confidence guidance becomes more significant, suggesting the effectiveness of confidence guidance in performance during inference.

4.4.2. Confidence Discount

In adaptive-horizon planning, we introduce a confidence discount function $\alpha(h)$ to reduce the confidence of planned trajectories with shorter horizons. We define it as a power function $\alpha(h) = (\frac{h}{h_N})^p$, where $p \in [0, 1]$ is a constant that adjusts the effect of trajectory horizons. Higher p renders a higher preference on longer trajectories. We conduct an ablation experiment on p in Maze2D Single tasks. When $p = 0$, trajectories with different lengths are equally selected. As shown in Table 5, the value of p needs to increase to obtain a high score when the maze size increases. This result could be explained as follows. In larger mazes, longer trajectories are expected to guarantee the achievement.

5. Related Work

5.1. Offline Reinforcement Learning

Offline reinforcement learning learns policies on static dataset without interacting with the environment. Due to the limited dataset, offline reinforcement learning faces the

Table 5. Ablation study on confidence discount for adaptive-horizon planning in Maze2D Single tasks. We have bolded the maximum score per task.

Environment	$p = 0.0$	$p = 0.2$	$p = 0.4$	$p = 0.6$	$p = 0.8$	$p = 1.0$
Single-Umaze	143.2 ±1.6	**145.3** ±1.2	131.5 ±1.0	128.9 ±1.0	122.3 ±0.6	121.7 ±0.6
Single-Medium	81.5 ±4.9	101.9 ±5.3	**140.3** ±1.8	135.3 ±1.2	131.5 ±1.0	130.6 ±1.1
Single-Large	89.2 ±6.3	125.3 ±4.7	157.0 ±4.5	**159.6** ±3.0	143.9 ±1.8	139.3 ±2.0

distributional shift problem (Levine et al., 2020; Prudencio et al., 2023). Policy constraint methods (Fujimoto et al., 2019; Kumar et al., 2019; Fujimoto & Gu, 2021) are proposed to solve this problem by keeping the learned policy close to the behavior policy that is used to collect the offline data. Furthermore, imitation learning methods (Chen et al., 2020; Wang et al., 2020; Emmons et al., 2021) are designed to imitate the behavior policy in a supervised manner. The learned policy can be improved by learning from good trajectories or conditional modeling. Trajectory optimization methods (Chen et al., 2021; Janner et al., 2021; Xu et al., 2022; Yamagata et al., 2023) have pioneered a new paradigm in offline reinforcement learning that views RL as a sequence modeling problem. It performs well in long-horizon credit assignment by assembling high-capacity sequence model architectures (Vaswani et al., 2017; Radford et al., 2019).

5.2. Diffusion Probabilistic Models in Offline RL

Most existing diffusion-based offline RL methods focus on improving the effectiveness and generalization of the sequential models. For example, the pioneering Diffuser (Janner et al., 2022) optimizes trajectory by iterative denoising using a diffusion model. Decision Diffuser (Ajay et al., 2023) employs conditional diffusion models to obtain return-maximizing trajectories and composed skills. Diffusion Q-learning (Wang et al., 2022) utilizes the diffusion model to learn highly-expressive policies. MetaDiffuser (Ni et al., 2023) formulates the generalization problem as conditional trajectory generation and utilizes a conditional diffusion model to overcome it. AdaptDiffuser (Liang et al., 2023) generates diverse synthetic data with a generator-discriminator paradigm, improving generalization on unseen tasks. Decision Stacks (Zhao & Grover, 2023) decompose goal-conditioned policy agents into three independent generative modules simulating observations, rewards, and actions. Some works employ DPMs in hierarchical reinforcement learning (Sutton et al., 1999) to solve long-horizon decision-making problems (Li et al., 2023; Chen et al., 2024). These works verify the effectiveness of diffusion models in offline decision-making.

5.3. Confidence Estimation Methods

To measure the reliability and trustworthiness of diffusion models, there are some methods (Angelopoulos et al., 2022; Horwitz & Hoshen, 2022; Teneggi et al., 2023) trying to derive statistically rigorous confidence intervals with a user-defined confidence guarantee called conformal prediction (Angelopoulos & Bates, 2021). It means the output result is ensured to fall into an interval with a given probability. These methods focus on the performance of diffusion models solely in image-to-image regression tasks. However, in reinforcement learning tasks, conformal prediction is often impractical due to the lack of ground-truth data.

Random Network Distillation (RND) (Burda et al., 2019) is an exploration approach in reinforcement learning that first surpasses human's performance on Montezuma's Revenge task (Bellemare et al., 2016). The novelty of a state is measured and then used as an intrinsic reward term to encourage the agent to explore the environment. As a heuristic approach to encourage the agent to explore the states that has never been seen yet, Random Network Distillation (RND) can be used for confidence estimation by fitting random priors (Osband et al., 2018; 2019; Ciosek et al., 2019), which is the focus of this work.

6. Conclusion

We introduce ReDiffuser, a method that enhances existing diffusion-based offline RL with confidence estimation to achieve reliable decision-making. We implement confidence estimation using Random Network Distillation (RND), providing quantitative criteria for decisions sampled from diffusion models. We also incorporate confidence guidance in goal-conditioned and reward-maximization tasks to adjust the sampling distribution toward more reliable regions. Experimental results in KUKA, Maze2D, and Locomotion demonstrate the effectiveness of ReDiffuser.

Limitation. ReDiffuser is an offline RL method and lacks of exploration capability. Thus, its performance is highly affected by the quality of the offline demonstration dataset. Furthermore, the current validations of ReDiffuser are performed on simple navigation and locomotion tasks, we would extend it to more complicated tasks.

Future work. We plan to extend the proposed method on more practical and challenging tasks such as robotic manipulation and autonomous driving (Chi et al., 2023; Yang et al., 2024; Liu et al., 2024). Furthermore, we consider sim-to-real transfer into consideration to demonstrate the effectiveness of ReDiffuser in addressing reliability and trustworthiness issues in real-world applications.

Acknowledgements

This work was partially supported by grants from the National Natural Science Foundation of China (No.62293544), China Postdoctoral Innovation Talents Support Program (BX20230184), and China Postdoctoral Science Foundation (No.2022M720077).

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 which we feel must be specifically highlighted here.

References

Ajay, A., Du, Y., Gupta, A., Tenenbaum, J., Jaakkola, T., and Agrawal, P. Is conditional generative modeling all you need for decision making? In *International Conference on Learning Representations*, 2023.

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Angelopoulos, A. N., Kohli, A. P., Bates, S., Jordan, M., Malik, J., Alshaabi, T., Upadhyayula, S., and Romano, Y. Image-to-image regression with distribution-free uncertainty quantification and applications in imaging. In *International Conference on Machine Learning*, pp. 717–730. PMLR, 2022.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

Chen, C., Fei, D., Kenji, K., Gulcehre, C., and Sungjin, A. Simple hierarchical planning with diffusion. In *International Conference on Learning Representations*, 2024.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Chen, X., Zhou, Z., Wang, Z., Wang, C., Wu, Y., and Ross, K. Bail: Best-action imitation learning for batch deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:18353–18363, 2020.

Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Ciosek, K., Fortuin, V., Tomioka, R., Hofmann, K., and Turner, R. Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representations*, 2019.

Emmons, S., Eysenbach, B., Kostrikov, I., and Levine, S. Rvs: What is essential for offline rl via supervised learning? In *International Conference on Learning Representations*, 2021.

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

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

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

Garrett, C. R., Lozano-Pérez, T., and Kaelbling, L. P. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pp. 440–448, June 2020.

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

Horwitz, E. and Hoshen, Y. Conffusion: Confidence intervals for diffusion models. *arXiv preprint arXiv:2211.09795*, 2022.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34: 1273–1286, 2021.

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*, pp. 9902–9915. PMLR, 2022.

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

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

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

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

Li, W., Wang, X., Jin, B., and Zha, H. Hierarchical diffusion for offline decision making. In *International Conference on Machine Learning*, pp. 20035–20064. PMLR, 2023.

Liang, Z., Mu, Y., Ding, M., Ni, F., Tomizuka, M., and Luo, P. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. *arXiv preprint arXiv:2302.01877*, 2023.

Liu, J., Hang, P., Zhao, X., Wang, J., and Sun, J. Ddm-lag : A diffusion-based decision-making model for autonomous vehicles with lagrangian safety enhancement, 2024.

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

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 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*, pp. 16784–16804. PMLR, 2022.

Osband, I., Aslanides, J., and Cassirer, A. Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Osband, I., Van Roy, B., Russo, D. J., Wen, Z., et al. Deep exploration via randomized value functions. *J. Mach. Learn. Res.*, 20(124):1–62, 2019.

Prudencio, R. F., Maximo, M. R., and Colombini, E. L. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Schreiber, G., Stemmer, A., and Bischoff, R. The fast research interface for the kuka lightweight robot. In *IEEE workshop on innovative robot control architectures for demanding (Research) applications how to modify and enhance commercial controllers (ICRA 2010)*, pp. 15–21. Citeseer, 2010.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

Teneggi, J., Tivnan, M., Stayman, W., and Sulam, J. How to trust your diffusion model: A convex optimization approach to conformal risk control. In *International Conference on Machine Learning*, pp. 33940–33960. PMLR, 2023.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, Z., Novikov, A., Zolna, K., Merel, J. S., Springenberg, J. T., Reed, S. E., Shahriari, B., Siegel, N., Gulcehre, C., Heess, N., et al. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33: 7768–7778, 2020.

Wang, Z., Hunt, J. J., and Zhou, M. Diffusion policies as an expressive policy class for offline reinforcement learning. *International Conference on Learning Representations*, 2022.

Williams, G., Aldrich, A., and Theodorou, E. Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*, 2015.

Xu, M., Shen, Y., Zhang, S., Lu, Y., Zhao, D., Tenenbaum, J., and Gan, C. Prompting decision transformer for few-shot policy generalization. In *international conference on machine learning*, pp. 24631–24645. PMLR, 2022.

Yamagata, T., Khalil, A., and Santos-Rodriguez, R. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In *International Conference on Machine Learning*, pp. 38989–39007. PMLR, 2023.

Yang, B., Su, H., Gkanatsios, N., Ke, T.-W., Jain, A., Schneider, J., and Fragkiadaki, K. Diffusion-es: Gradient-free planning with diffusion for autonomous driving and zero-shot instruction following, 2024.

Zhao, S. and Grover, A. Decision stacks: Flexible reinforcement learning via modular generative models. *arXiv preprint arXiv:2306.06253*, 2023.

A. Comparison with Other Diffuser Algorithms

Table 6. **Maze2D.** The performance of ReDiffuser, Diffuser and several diffusion-based offline reinforcement learning algorithms in the Maze2D environment. We report the mean and bias of scores over 1∼10 random seeds and each seed corresponds to 1000 planning episodes. 100 corresponds to a reference expert policy. We have bolded the maximum score per task.

Environment	**Diffuser**	**AdaptDiffuser**	**DS**	**HDMI**	**HD**	**ReDiffuser**
Single-Umaze	121.8±0.6	135.1	111.3	120.1	128.4	**145.3** ±1.2
Single-Medium	130.7±0.8	129.9	111.7	121.8	135.6	**140.3** ±1.8
Single-Large	130.0±1.7	167.9	**171.6**	128.6	155.8	159.6 ±3.0
Average	127.5	144.3	131.5	123.5	139.9	**148.4**
Multi-Umaze	129.2 ±1.3	_	121.3	131.3	144.1	**154.8** ±1.8
Multi-Medium	128.7 ±1.4	_	122.3	131.6	140.2	**149.4** ±2.3
Multi-Large	142.3 ±1.6	_	126.7	135.4	165.5	**176.1** ±3.4
Average	133.4	_	123.4	132.8	149.9	**160.1**

Table 7. **Locomotion.** The performance of ReDiffuser, Diffuser and several diffusion-based offline reinforcement learning algorithms in MuJoCo environment on the D4RL Locomotion tasks. We report the mean and bias of scores over 1∼10 random seeds and each seed corresponds to 10 planning episodes. We have bolded the maximum score per task.

Dataset	**Environment**	**Diffuser**	**AdaptDiffuser**	**DD**	**DS**	**Diffusion-QL**	**HDMI**	**HD**	**ReDiffuser**
Medium-Expert	HalfCheetah	86.9 ±5.2	89.6	90.6	95.7	**96.8**	92.1	92.5	88.3 ±3.1
Medium-Expert	Hopper	102.1 ±12.0	111.6	111.8	107.0	111.1	113.5	**115.3**	104.8 ±9.4
Medium-Expert	Walker2d	104.9 ±4.8	108.2	108.8	108.0	**110.1**	107.9	107.1	107.4 ±1.4
Medium	HalfCheetah	45.2 ±0.5	44.2	49.1	47.8	**51.1**	48.0	46.7	45.2 ±0.7
Medium	Hopper	92.9 ±12.9	96.6	79.3	76.6	90.5	76.4	99.3	**100.2** ±6.0
Medium	Walker2d	80.0 ±1.7	84.4	82.5	83.6	**87.0**	79.9	84.0	82.5 ±2.3
Medium-Replay	HalfCheetah	36.6 ±2.8	38.3	39.3	41.1	**47.8**	44.9	38.1	38.6 ±1.7
Medium-Replay	Hopper	57.2 ±12.7	92.2	100.0	89.5	**101.3**	99.6	94.7	62.4 ±6.9
Medium-Replay	Walker2d	80.6 ±13.2	84.7	75.0	80.7	**95.5**	80.7	84.1	82.2 ±10.7
Average		76.3	83.4	81.8	81.1	**88.0**	82.6	84.6	79.1

B. Implementation Details

Architecture. The architecture of the target network consists of 3 one-dimensional convolutional layers and 1 fully connected layer in all tasks (KUKA, Maze2D and Locomotion) except for HalfCheetah whose horizon of the decision is 4, so we simply model its target network with 4 fully connected layers. The architecture of the predictor network is the addition of two fully connected layers to the corresponding target network. The output dimension of RND varies with the complexity of the decision space: 240 for Umaze, 336 for Medium, 384 for Large, 510 for KUKA, 30 for HalfCheetah, 150 for Hopper and 250 for Walker2d. The settings of the training horizon and number of denoising steps are following Diffuser.

Training. We collect the decision training set following the offline setting, which is generated by the trained Diffuser. We randomly reset the initial state and preserve the planned trajectories. At the training stage, the model is trained with a learning rate of 1e-04 and batch size of 256. In adaptive-horizon planning, we set the gap between adjacent horizons as 32; In value-embedded planning, we preserve the decision with the maximum value among 64 candidate sampled decisions.

Exploration for sequential execution in Locomotion. We have explored the possibility of continuously executing planned actions in Locomotion tasks based on the evaluation of confidence. However, this attempt fails due to the Diffuser's inability to capture the complex dynamics in these tasks. We draw some figures to show the bias between the planned state and the actual state at each time step during an episode. While the bias is trending upward to cause cumulative errors when executing a sequence of actions, we believe the bias will diminish with enhanced model fitting capabilities.

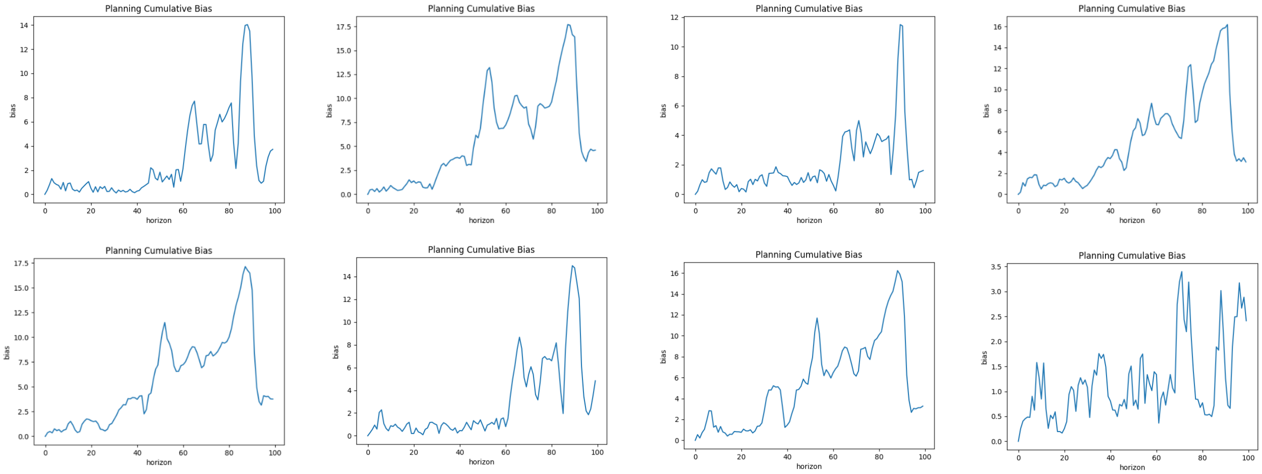


Figure 5. Visualization of the bias between the planned states and the actual states.